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 exacerbates the undesirable behaviors of neural networks such as memorization and sensitivity to out-of-distribution samples. This work addresses these issues with three contributions. First, we propose a simple, yet effective data augmentation routine, termed Spectral Mixup, to construct effective virtual training samples. Second, 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. Finally, we extend the network to a semi-supervised setting so that it can learn from datasets containing low-resolution HSIs only. With these contributions, our method is able to learn from heterogeneous datasets and virtual examples. We find that while it is difficult to collect HR HSIs, it is relatively easy to collect only LR HSIs and it is very easy to collect HR RGB images. It is thus very appealing to have a HSI SR method which can learn from these heterogeneous sources. Our method is designed for this aim.

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 [22], remote sensing [21] 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), the topic of HSI SR has received little attention. One of the main reasons is the lack of large-scale HSI datasets for 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 from heterogeneous datasets and virtual examples.

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 and neighboring spectral bands 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 and to reduce the computational complexity, we propose a universal group convolutional neural network that can accommodate different spectral groups. We also expand the number of bands of RGBIs via a linear spectral interpolation. This way, the size of the group for the network can be chosen freely.

Semi-supervised learning (SSL) exploits unlabeled data to reduce over-fitting to the limited amount of labeled data [16, 31, 45, 48, 24]. While good progress has been achieved, the strategies are mainly designed for image classification and object recognition. 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 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 RGBI 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.

We further propose a data augmentation routine, termed Spectral Mixup, to create effective ‘virtual’ training examples. Data augmentation is a strategy to create virtual samples by modifying the original samples. Data augmentation is known to increase the generalizability of learning methods. Common methods for classification tasks include reflections, rotations, cropping, and color jittering. They assume that examples obtained by those operations share the same class with the original example and that can hardly be applied to our regression task. The recent work mixup [57] creates virtual examples by using convex combinations of pairs of examples and their labels to favor functions which preserve simple linear behavior in-between training examples. Motivated by these, we propose Spectral Mixup for HSI SR to create virtual samples using convex combinations of spectral bands of the same image. Spectral Mixup favors functions that preserve simple linear behavior in-between spectral bands. We show in experiments that it outperforms mixup for our task.

To summarize, this work makes three contributions: 1) a multi-tasking HSI SR method to learn together with an auxiliary RGBI SR task; 2) A SSL method to learn also from ‘unlabeled’ LR HSIs; and 3) a simple, yet effective data augmentation method Spectral Mixup.

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; 2) Single HSI SR from LR HSIs; and 3) HSI SR from both HR RGBIs and LR HSIs of the same scene. Our method belongs to the second group.

HSR 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 [7] to deep neural networks [20, 44, 8]. This trend highlights the need for bigger training datasets.

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 [27, 59], using feature pyramids [30], employing GAN losses [33], and modeling real-world degradation effects [23]. As to single HSI SR, there has been great early work [3, 60] 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, 34] 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 [35, 26]. While the backbone network of our method is also based on GCs, our three proposed contributions are all new.

Fusion-based methods use HR RGBIs of the same scene as references to improve the spatial resolution of the LR HSIs [12, 51]. This stream of methods have received more research attention than the former two. Many learning techniques have been applied to this data fusion task including Bayesian inference [5, 6, 58], matrix factorization [32, 17], sparse representation [4, 19], and deep neural networks [41, 49]. The common goal of these methods is to learn to propagate the detailed information in the HR RGBIs 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 are very well co-registered [26]. This data registration is a challenge on its own and registration errors will lead to degraded SR results [14, 61].

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 [29, 11, 42]. For instance, generative approaches have been explored in [29] and predicting the orientation of image patches is used in [11]. Another related setting is multi-task learning (MTL) [46]. 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 [10], task consistency [56], and loss balance [15] are useful for learning with auxiliary tasks.
3. Approach

HSIs provide tens of narrow bands, so processing all the bands together is time-consuming and requires very large datasets in order to avoid over-fitting. In this work, we follow [26] and use a grouping strategy to divide input HSIs into overlapping groups of bands. This way, the spectral correlation among neighboring bands can be effectively exploited without increasing the parameters of the model. Another major advantage of using a grouping strategy is that it offers the possibility to train our auxiliary task RGBI SR along with our main task HSI SR within the same network. Without using the grouping strategy, the difference in the number of bands is very large between the two tasks. 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.

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., $D_{HS} = \{x_{HS}^i, x_{HS}^{iN}_{HS}\}_{i=1}^N$ and $D_{RGB} = \{x_{RGB}^i, x_{RGB}^{iN}_{RGB}\}_{i=1}^N$, where $x_{HS}^i \in \mathbb{R}^{H_1 \times W_1 \times C}$, $x_{HS}^{iN}_{HS} \in \mathbb{R}^{H_1 \times W_1 \times C}$, $x_{RGB}^i \in \mathbb{R}^{h_2 \times w_2 \times Z}$, and $x_{RGB}^{iN}_{RGB} \in \mathbb{R}^{h_2 \times w_2 \times Z}$. 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 $h$, $w$, and $H$. Given a scaling factor $\tau$, we have $H_i = \tau H_1$ and $W_i = \tau W_1$ 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, $\Phi_{HS} = (\Phi_{En}, \Phi_{De}^{HS})$ and $\Phi_{RGB} = (\Phi_{En}, \Phi_{De}^{RGB})$. The general architecture is shown in Fig. 1. In order to share the same encoder between the two SR tasks, we divide the $C$ input channels of $x_{HS}$ into groups of $M$ bands. For HSI SR, the encoder network $\Phi_{En}$ takes $M$ channels as input and generates $M$ channels as output. The outputs of all the groups of $x_{HS}$ are then concatenated according to their original spectral band position to assemble a new HSI $X_{HS} \in \mathbb{R}^{H_1 \times W_1 \times C}$. The neighboring groups of $x_{HS}$ can have overlaps and we average the results of the overlapping areas when assembling $X_{HS}$. 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 [30] and HSI SR [26]. The reconstructed $X_{HS}$ is then fed into the decoder network $\Phi_{De}^{HS}$ as a whole to generate the final output $\hat{X}_{HS}$, which is then compared to the ground truth $X_{HS}$ to compute the loss. $\Phi_{De}^{HS}$ takes all the bands directly to learn long-range spectral correlations beyond individual groups to refine the results.

For RGBI SR, we first increase the number of bands of $x_{RGB}$ from $Z$ to $M$ via a simple spectral interpolation which will be explained in Sec. 3.1.1. The interpolated $M$-band image is then passed through the encoder $\Phi_{En}$ to obtain a new $M$-band RGB image $\hat{X}_{RGB} \in \mathbb{R}^{H_2 \times W_2 \times M}$ of the desired resolution. Because the decoder is shared by two tasks, $\hat{X}_{RGB}$ is also needed to be fed to its own task-specific decoder network $\Phi_{De}^{RGB}$ for further refinement. The final output $\hat{X}_{RGB}$ from $\Phi_{De}^{RGB}$ is then compared to the ground-truth image $X_{RGB}$.

In order to have a modular design, the three subnetworks have the same basic architecture. They are all composed of a sequence of Spatial-Spectral Block (SSB) modules. The SBB module was proposed in [26] as a basic building block for their HSI SR network. Each SBB has a Spatial Residual Module and a Spectral Attention Residual Module. Two Convolutional layers (the first one followed by a ReLu layer) with 3x3 filters are used in the Spatial Residual Module to capture spatial correlations. Two Convolutional layers (the first one again followed by a ReLu layer) with 1x1 filters are used in the Spectral Attention Residual Module to capture spectral correlations. Please refer to the Fig. 2 in [26] for more details of the SBB module. We construct the whole network with standard Convolutional Layers, SSBs, Upsampling Layers and Concatenation Operations. There are also skip connections at multiple scales to facilitate the information flow. The input LR images are also scaled to the desired size via Bicubic Interpolation and fused with the network output for residual learning. The complete network is shown in Fig. 1. We employ the PixelShuffle [43] operator for the upsampling layer. Given a scaling factor $\tau$, the first upsampling layer upcales the features $\tau/2$ times and the second one handles the remaining $\times 2$ factor. The internal features of all SBB modules are limited to 256 in this work. The filter size of all Convolutional Layers, except for those in the Spectral Attention Residual Module of SSBs, are set to $3 \times 3$.

3.1.1 Spectral Interpolation of RGB Images

The task is to increase the number of band from $Z$ to $M$ for RGB images. $Z = 3$ in this case and $M = 8$ in this work. Because the generated $M$-band images will be used to train the SR network for supervision transfer to HSI SR, we posit
Figure 1: The architecture of our network consisting of a shared encoder and two specific decoders for the two SR tasks.

Figure 2: The pipeline of our semi-supervised learning.

that these new images need to have certain properties. First, they should not contain artifacts. Second, the correlation between the bands of the new images should follow a distance rule in that the correlation between neighboring HSI bands should be higher than that between distant bands. For this, we propose a simple interpolation method. Given \(Z\) bands, we interpolate \(K = (M - Z)/(Z - 1)\) new bands to each of the \(Z - 1\) intervals between consecutive bands. For the \(i^{th}\) band \(\hat{x}(i)\) between the original bands \(z\) and \(z + 1\), we have:

\[
\hat{x}(i) = (1 - \frac{i}{K + 1})x(z) + \frac{i}{K + 1}x(z + 1).
\]

Note that if \(K\) is not an integer, we use \(\lceil K \rceil\) for the first interval and \(\lfloor K \rfloor\) for the second one.

3.2. Spectral Mixup

Data augmentation is a strategy to create virtual samples by alternating the original samples. The recent mixup method [57] creates virtual examples by using convex combinations of pairs of examples and their labels. While it is very effective for high-level classification tasks, it does not offer help for low-level SR tasks [53] because the detailed image structures are broken up by their mixing up of two images. These detailed structures are important for SR tasks. Taking account of this observation, we propose a data augmentation routine Spectral Mixup specifically for HSI SR. It creates virtual samples and their ground truths by using convex combinations of spectral bands within self-image and within its ground-truth image, respectively.

More specifically, given \(x_{\text{HS}}\) and it ground truth \(X_{\text{HS}}\), both with \(C\) channels, we generate a mixing matrix \(B \in \mathbb{R}^{C \times C}\) filled with random numbers from a uniform distribution on the interval \([0, 1]\). \(B\) is then row-wise normalized to make sure that the values in the projected image have the same magnitude as that of the original image. The new example and its ground truth are then created as:

\[
\hat{x}_{\text{HS}}^{(i,j)} = \alpha x_{\text{HS}}^{(i,j)} + (1 - \alpha)Bx_{\text{HS}}^{(i,j)},
\]

\[
\hat{X}_{\text{HS}}^{(i,j)} = \alpha X_{\text{HS}}^{(i,j)} + (1 - \alpha)BX_{\text{HS}}^{(i,j)},
\]

where \((i,j)\) index over all positions to get the values of pixels. The randomly projected images are fused with the original images to strike a balance between increasing variations and preserving the fidelity of real HSIs. For instance, the relationships between the bands of real HSIs should be largely kept. In this work, \(\alpha\) is set to 0.5 and we study the influence of this parameter in Sec. 4. The implementation of Spectral Mixup training is very straightforward and can be done with a few lines of code. Spectral Mixup also introduces very little computation overhead. By applying it, more examples from the vicinity of the original example can be sampled. Learning with those new examples encourages the network to have simple linear behavior in-between spectral bands which is found very useful for HSI SR.
3.3. Semi-Supervised HSI SR

While training with auxiliary RGB SR task and Spectral Mixup can greatly improve the performance, there is still a need to learn from unlabeled HSIs, i.e. LR HSIs without HR HSIs as ground truth. There are a diverse sets of methods developed for semi-supervised learning (SSL) such as entropy minimization and using pseudo-labels. However, they are mostly designed for high-level classification 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. It works as follows: Given an image $x_{\text{HS}}$, we convert it to an RGB image $x_{\text{RGB}}$ with the response function of a standard RGB camera $F$: $x_{\text{RGB}} = F(x_{\text{HS}})$. The original HSI $x_{\text{HS}}$ and the converted RGB image $x_{\text{RGB}}$ 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: $\hat{x}_{\text{HS}} = \Phi_{\text{HS}}(x_{\text{HS}})$ and $\hat{x}_{\text{RGB}} = \Phi_{\text{RGB}}(x_{\text{RGB}})$. $\hat{x}_{\text{HS}}$ is then converted to an RGB image by using the same camera response function: $\hat{x}_{\text{RGB}} = F(\hat{x}_{\text{HS}})$. Finally, a consistency loss $L_{\text{SSL}}(\hat{x}_{\text{RGB}}, \hat{x}_{\text{RGB}})$ is computed between the two HR RGB results. This consistency check makes a good use of ‘unlabeled’ HSIs and ‘labeled’ RGB images. It transfers supervision from the RGB side to the HSI side. The diagram of this SSL method is shown in Fig. 2.

3.4. Loss Function

In order to capture both the spatial information and spectral correlation of the SR results, we follow [26] 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:

$$L_{\text{SSTV}} = \frac{1}{N} \sum_{n=1}^{N} (||\nabla_h \hat{X}^n||_1 + ||\nabla_w \hat{X}^n||_1 + ||\nabla_c \hat{X}^n||_1),$$

(4)

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

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

(5)

The overall loss for our SSL tasks is:

$$L_{\text{Total}} = L_{\text{HS}}(X_{\text{HS}}, \hat{X}_{\text{HS}}) + L_{\text{SMixup}}(\hat{X}_{\text{HS}}, \hat{X}_{\text{HS}}) + L_{\text{RGB}}(X_{\text{RGB}}, \hat{X}_{\text{RGB}}) + L_{\text{SSL}}(\hat{X}_{\text{RGB}}, \hat{X}_{\text{RGB}}).$$

(6)

The main loss is augmented by the three auxiliary losses which are optional but highly beneficial. Since the final loss combines multiple terms for leveraging heterogeneous datasets and virtual examples, it may introduce a few hyper-parameters to balance the contributions of all terms. HSI SR methods with many hyperparameters can be problematic as the validation set is quite small normally to properly search for good values. However, we find in practice that the weights of those losses can be fixed to 1 and do not need to be tuned on a per-experiment or per-dataset basis. If desired, their contribution can be controlled by using different amounts of training data for each loss and by altering the frequency of feeding training samples for each loss.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on four public datasets. The datasets considered are three nature HSI datasets: CAVE dataset [50], Harvard dataect [13], and NTIRE 2020 dataset [8], and one remote sensing HSI dataset Chikusei [52]. Images in CAVE 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 Chikusei dataset has 128 bands spanning from 363 nm to 1018 nm.

The CAVE dataset contains 32 images of 512 × 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, 1/4 of the training data is considered as the labeled set and the remaining 3/4 considered as unlabeled. 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 images are taken as the labeled images (with HR HSIs) while the remaining 34 are taken as unlabeled images. 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 as labeled images and 300 as unlabeled images. For Chikusei, there is only one big image of 2517 × 2335 pixels. We cropped 4 image crops of 256 × 256 pixels for test and use the rest for training. 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 ×2 and take these downsampled images as our HR RGB images.

Methods. We compare the proposed method to four state-of-the-art HSI SR methods: GDRRN [35], 3DFCNN [38], SSPSR [26], and MCNet [34]. We use exactly 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.

Training Details. The method is implemented with PyTorch. We use the ADAM optimizer [28] and train all variants of our method for 10 epoches. This is a small number...
for image restoration and RGBI SR. Due to space limit, for in HSI fusion task, while the other three are standard metrics of the reconstructed HSIs, their mean values of all spectral bands are reported. CC, SAM, and ERGAS are widely used in HSI fusion task, while the other three are standard metrics for image restoration and RGBI SR. Due to space limit, for these choices will be studied in Sec. 4.3.

We first present the results in the semi-supervised setting for the ×4 case. **Bold** indicates the best results. RGBSR means learning with auxiliary RGBI SR. SMixup is our Spectral Mixup.

| Methods | Components | CAVE | Harvard | NTIRE |
|---------|------------|------|---------|--------|
| Ours    |            |      |         |        |
| Ours    | ✓          | 0.01282 | 41.38106 | 3.99731 | 0.01431 | 40.42362 | 3.20297 | 0.01675 | 37.76302 | 2.36922 |
| Ours    | ✓          | 0.01191 | 42.01763 | 3.63288 | 0.01366 | 40.69577 | 3.11316 | 0.01583 | 38.21667 | 2.27241 |
| Ours    | ✓          | 0.01246 | 42.08879 | 3.54516 | 0.01375 | 40.63699 | 3.12395 | 0.01542 | 38.56435 | 2.16249 |
| Ours    | ✓          | 0.01173 | 42.57401 | 3.40209 | 0.01348 | 40.80286 | 3.07281 | 0.01515 | 38.75436 | 2.11092 |
| Ours    | ✓          | 0.01187 | 42.21247 | 3.53264 | 0.01351 | 40.81661 | 3.06886 | 0.01547 | 38.55183 | 2.16476 |
| Ours (final) | ✓  ✓  ✓ | 0.01138 | 42.74453 | 3.31609 | 0.01335 | 40.89392 | 3.03947 | 0.01507 | 38.83877 | 2.09341 |

| Components | Metrics |
|------------|---------|
| RGBSR | SSL | RMSE ↓ | CC ↑ | MPSNR ↑ | MSSIM ↑ | ERGAS ↓ | SAM ↓ |
| Ours    |            | 0.01230 | 0.94992 | 39.71319 | 0.93529 | 5.38314 | 2.58381 |
| Ours    | ✓          | 0.01211 | 0.95161 | 39.80637 | 0.93691 | 5.48464 | 2.55369 |
| Ours    | ✓          | 0.01216 | 0.95097 | 39.82008 | 0.93649 | 5.46244 | 2.56245 |
| Ours    | ✓          | 0.01215 | 0.95096 | 39.83383 | 0.93675 | 5.24407 | 2.59671 |
| Ours    | ✓          | 0.01219 | 0.95109 | 39.81107 | 0.93617 | 5.25279 | 2.56749 |
| Ours (final) | ✓  ✓  ✓ | 0.01181 | 0.95375 | 40.09431 | 0.94035 | 5.08513 | 2.49154 |

Table 1: Results of all methods on the CAVE, Harvard, and NTIRE datasets in the semi-supervised setting for the ×4 case. **Bold** indicates the best results. RGBSR means learning with auxiliary RGBI SR. SMixup is our Spectral Mixup.

| Ours | 0.01282 | 0.01375 | 0.01542 | 0.01211 | 0.01215 | 0.01219 | 0.01181 |
| Ours (final) | 0.01138 | 0.01335 | 0.01507 | 0.01211 | 0.01215 | 0.01219 | 0.01181 |

Table 2: Results of all methods on the Chikusei dataset in the semi-supervised setting for the ×4 case. **Bold** indicates the best results. RGBSR means our auxiliary RGBI SR task. SMixup is our Spectral Mixup.

compared to the ones used by comparison methods. For instance, GDRRN [35] trains for 30 epochs, 3DFCNN [38] trains for 200 epochs, SSPSR [26] for 40 epoches, and MCNet [34] for 200 epoches. We choose a small number in order to thoroughly evaluate all the variants of our method. We find that 10 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 $10^{-3}$ and is reduced by a factor of 0.3 after every 3 epoches. As to the batch size, 16 is used for all the methods. For our method, when the SSL loss is used, we use 8 due to the GPU memory limit.

**Evaluation Metrics.** We follow the literature [26] and evaluate the performance of all methods under six widely used metrics. They are cross correlation (CC) [36], spectral angle mapper (SAM) [55], root mean squared error (RMSE), erreur relative globale adimensionelle de synthese (ERGAS) [47], peak signal-to-noise ratio (PSNR), and structure similarity (SSIM) [62]. For PSNR and SSIM of the reconstructed HSIs, their mean values of all spectral bands are reported. CC, SAM, and ERGAS are widely used in HSI fusion task, while the other three are standard metrics for image restoration and RGBI SR. Due to space limit, for some experiments, we only report numbers of three metrics and put the rest into the supplementary material.

**Other Parameters.** In this work, we focus on scaling factor $\times 4$ and $\times 8$. We report the results for $\times 4$ in the main paper, and report the results of $\times 8$ in the supplementary material. 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. We use group size 8, i.e., $M = 2$, with an overlap of 2 by following [26]. The response function of Canon 1D Mark 3 [25] is used. In all the main experiments, the size of the RGB dataset is 3 times of the ‘labeled’ HSI dataset, so its size changes on a per-dataset basis. $\alpha$ is set to 0.5. The influence of some of these choices will be studied in Sec. 4.3.

### 4.2. 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 NTRIE dataset are shown in Table 1. The results in this table show that our method outperforms all other state-of-the-art methods.
 Methods | RMSE | MPNSR | ERGAS
---|---|---|---
Bicubic | 0.03568 | 29.89529 | 4.86576
3DFCNN | 0.03129 | 30.92347 | 4.26353
SSPSR | 0.02713 | 32.18152 | 3.69979
Ours | 0.01993 | 34.42891 | 2.71569

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

| Methods | Components | RMSE ↓ | MPNSR ↑ | ERGAS ↓ | RMSE ↓ | MPNSR ↑ | ERGAS ↓ |
|---|---|---|---|---|---|---|---|
| Ours | RGBSR | 0.01196 | 42.38359 | 3.45903 | 0.01344 | 40.91014 | 3.01309 |
| Ours | RGBSR | 0.01109 | 42.73668 | 3.35884 | 0.01325 | 41.03709 | 2.96643 |
| Ours | RGBSR | 0.01134 | 42.88402 | 3.28051 | **0.01317** | **41.08568** | 2.95718 |

| GDRRN [35] | - | - | 0.01629 | 39.74705 | 4.52683 | 0.01484 | 39.62759 | 3.67993 |
| 3DFCNN [38] | - | - | 0.01583 | 39.21786 | 5.41798 | 0.01519 | 39.66271 | 3.47738 |
| SSPSR [26] | - | - | 0.01245 | 42.13787 | 3.55146 | 0.01352 | 40.81499 | 3.05007 |
| MCNet [34] | - | - | 0.01245 | 42.25978 | 3.56246 | 0.01405 | 40.59229 | 3.10529 |

*Table 3*: Results of all methods on the CA VE and Harvard datasets in the fully-supervised setting for the ×4 case. **Bold** indicates the best results. RGBSR means our auxiliary RGBI SR task. SMixup is our Spectral Mixup.

significantly and consistently over all datasets and under all evaluation metrics. We would like to point out that our baseline model – our method without any of the three proposed contributions – is already a top-performing method and performs better than other comparing methods.

The good performance of our base model is mainly from its network design. Our deep group convolutional network is built on top of the recent work by Jiang et al. [26] with extensions. The major differences are: 1) we have used a deeper encoder (brunch) network, and 2) we have moved all the upsampling layers into the encoder and have removed a bottleneck layer (an intermediate image reconstruction layer with \( M \) channels) between the two upsampling layers. We find that these two changes improve the results. We also find that 3D convolution based methods are computationally heavy. That is probably the reason why the network of [38] is quite shallow. We believe that this is the reason why their method does not give top results. When compared to the very recent method MCNet [34], our base model also performs better in almost all cases. This is especially interesting because MCNet is trained for 200 epochs while our method is trained only for 10 epochs. Our results reinforce the findings made in [26] that group convolutional networks are good at extracting the correlation between spectral bands without increasing the model size.

We can also find that our proposed contributions, namely training with the auxiliary task RGBI SR, data augmentation via Spectral Mixup and the semi-supervised learning method based on cross-model consistency, all contribute positively and significantly to the final results. It is worth noticing that these improvements are made on top of our strong base model. This means the improvements are not
3. Conclusion

In this paper, we have proposed a new method for hyperspectral image (HSI) super-resolution (SR). We
build a deep group convolutional network which yields the state-of-the-art results. To further improve it, we have proposed three 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, a simple, yet effective data augmentation method Spectral Mixup is proposed to create virtual training samples for HSI SR to increase the robustness of the network to new examples. Finally, 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 three contributions are important and they help our method set a new state of the art on four public datasets.

References

[1] H. K. Aggarwal and A. Majumdar. Hyperspectral image denoising using spatio-spectral total variation. IEEE Geoscience and Remote Sensing Letters, 13(3):442–446, 2016.

[2] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[3] T. Akgun, Y. Altunbasak, and R. M. Mersereau. Spatio-spectral super-resolution of hyperspectral images. IEEE Transactions on Image Processing, 14(11):1860–1875, 2005.

[4] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Sparse spatio-spectral representation for hyperspectral image super-resolution. In ECCV, 2014.

[5] N. Akhtar, F. Shafait, and A. Mian. Bayesian sparse representation for hyperspectral image super-resolution. In CVPR, 2015.

[6] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Hierarchical beta process with gaussian process prior for hyperspectral image super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, ECCV, 2016.

[7] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In ECCV, 2016.

[8] Boaz Arad, Radu Timofte, Ohad Ben-Shahar, Yi-Tun Lin, and Graham D. Finlayson. Ntire 2020 challenge on spectral reconstruction from an rgb image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020.

[9] Seung-Hwan Baek, Incheon Kim, Diego Gutierrez, and Min H. Kim. Compact single-shot hyperspectral imaging using a prism. ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2017), 36(6):217:1–12, 2017.

[10] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Semantic object prediction and spatial sound prediction with binaural sounds. In European Conference on Computer Vision (ECCV), 2020.

[11] Lucas Beyer, Xiaohua Zhai, Avital Oliver, and Alexander Kolesnikov. S4L: self-supervised semi-supervised learning. In ICCV, 2019.

[12] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5(2):354–379, 2012.

[13] A. Chakrabarti and T. Zickler. Statistics of Real-World Hyperspectral Images. In CVPR, 2011.

[14] C. Chen, Y. Li, W. Liu, and J. Huang. Sirf: Simultaneous satellite image registration and fusion in a unified framework. IEEE Transactions on Image Processing, 24(11):4213–4224, 2015.

[15] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In CVPR, 2018.

[16] Dengxin Dai and Luc Van Gool. Ensemble projection for semi-supervised image classification. In ICCV, 2013.

[17] R. Dian, L. Fang, and S. Li. Hyperspectral image super-resolution via non-local sparse tensor factorization. In CVPR, pages 3862–3871, 2017.

[18] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2):295–307, 2016.

[19] W. Dong, F. Fu, G. Shi, X. Cao, J. Wu, G. Li, and X. Li. Hyperspectral image super-resolution via non-negative structured sparse representation. IEEE Transactions on Image Processing, 25(5):2337–2352, 2016.

[20] S. Galliani, Charis Laranas, D. Marmanis, E. Baltasavias, and K. Schindler. Learned spectral super-resolution. ArXiv, abs/1703.09470, 2017.

[21] Alexander F.H. Goetz. Three decades of hyperspectral remote sensing of the earth: A personal view. Remote Sensing of Environment, 113:S5 – S16, 2009.

[22] A.A. Gowen, C.P. O’Donnell, P.J. Cullen, G. Downey, and J.M. Frias. Hyperspectral imaging – an emerging process analytical tool for food quality and safety control. Trends in Food Science & Technology, 18(12):590 – 598, 2007.

[23] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In CVPR, 2020.

[24] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Köring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. arXiv preprint arXiv:2012.10782, 2020.

[25] J. Jiang, D. Liu, J. Gu, and S. S¨usstrunk. What is the space of spectral sensitivity functions for digital color cameras? In 2013 IEEE Workshop on Applications of Computer Vision (WACV), 2013.

[26] J. Jiang, H. Sun, X. Liu, and J. Ma. Learning spatial-spectral prior for super-resolution of hyperspectral imagery. IEEE Transactions on Computational Imaging, 6:1082–1096, 2020.
[27] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In CVPR, 2016. 2
[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5
[29] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In NeurIPS, 2014. 2
[30] W. Lai, J. Huang, N. Ahuja, and M. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In CVPR, 2017. 2, 3
[31] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In ICLR, 2017. 1
[32] C. Lanaras, E. Baltsavias, and K. Schindler. Hyperspectral super-resolution by coupled spectral unmixing. In 2015 IEEE International Conference on Computer Vision (ICCV), 2015. 2
[33] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017. 2
[34] Qiang Li, Qi Wang, and Xuelong Li. Mixed 2d/3d convolutional network for hyperspectral image super-resolution. Remote Sensing, 12(10), 2020. 2, 5, 6, 7, 12, 13, 14, 15, 16
[35] Y. Li, Lei Zhang, C. Ding, Wei Wei, and Y. Zhang. Single hyperspectral image super-resolution with grouped deep recursive residual network. 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), 2018. 2, 5, 6, 7, 12, 13, 14, 15, 16
[36] L. Loncan, L. B. de Almeida, J. M. Bioucas-Dias, X. Briot, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simões, J. Tourneret, M. A. Veganzones, G. Vivone, Q. Wei, and N. Yokoya. Hyperspectral pansharpening: A review. IEEE Geoscience and Remote Sensing Magazine, 3(3):27–46, 2015. 6
[37] Guolan Lua and Baowei Fei. Medical hyperspectral imaging: a review. Journal of Biomedical Optics, 2014. 1
[38] Shaohui Mei, Xin Yuan, Jingyu Ji, Yifan Zhang, Shuai Wan, and Qian Du. Hyperspectral image spatial super-resolution via 3d full convolutional neural network. Remote Sensing, 9(11), 2017. 2, 5, 6, 7, 12, 13, 14, 15, 16
[39] N. M. Nasrabadi. Hyperspectral target detection : An overview of current and future challenges. IEEE Signal Processing Magazine, 31(1):34–44, 2014. 1
[40] Rang M. H. Nguyen, Dilip K. Prasad, and Michael S. Brown. Training-based spectral reconstruction from a single rgb image. In ECCV, 2014. 2
[41] Ying Qu, Hairong Qi, and Chiman Kwan. Unsupervised sparse dirichlet-net for hyperspectral image super-resolution. In CVPR, 2018. 2
[42] Baifeng Shi, Judy Hoffman, Kate Saenko, Trevor Darrell, and Huijuan Xu. Auxiliary task reweighting for minimum-data learning. In NeurIPS, 2020. 2, 8
[43] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR, 2016. 3
[44] Z. Shi, C. Chen, Z. Xiong, D. Liu, and F. Wu. Hsnn+: Advanced cnn-based hyperspectral recovery from rgb images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. 2
[45] Annti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. 2017. 1
[46] Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, Dengxin Dai, , and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey, 2020. 2
[47] L. Wald. Data fusion: definitions and architectures: fusion of images of different spatial resolutions. In Presses des MINES, 2002. 6
[48] Quizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In CVPR, 2020. 1
[49] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by MS/HS fusion net. In CVPR, 2019. 2
[50] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum. IEEE Transactions on Image Processing, 19(9):2241–2253, 2010. 5
[51] N. Yokoya, C. Grohnfeldt, and J. Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. IEEE Geoscience and Remote Sensing Magazine, 5(2):29–56, 2017. 2
[52] N. Yokoya and A. Iwasaki. Airborne hyperspectral data over chikusei. Space Application Laboratory, University of Tokyo, Japan, Tech. Rep. SAL-2016-05-27, May 2016. 5
[53] Jaehun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In CVPR, 2020. 4, 8
[54] Y. Yuan, X. Zheng, and X. Lu. Hyperspectral image super-resolution by transfer learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(5):1963–1974, 2017. 2
[55] R. H. Yuhas, A. F. Goetz, and booktitle=JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop year = 1992 J. W. Boardman, title=Discrimination among semiarid landscape endmembers using the spectral angle mapper (sam) algorithm. 6
[56] Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In CVPR, 2020. 2
[57] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In International Conference on Learning Representations, 2018. 2, 4, 8
[58] Lei Zhang, Jiatao Nie, Wei Wei, Yanning Zhang, Shengcai Liao, and Ling Shao. Unsupervised adaptation learning for hyperspectral imagery super-resolution. In CVPR, 2020. 2
[59] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. 2
[60] Y. Zhao, Jinxiang Yang, Qingyong Zhang, L. Song, Y. Cheng, and Q. Pan. Hyperspectral imagery super-resolution by sparse representation and spectral regularization. *EURASIP Journal on Advances in Signal Processing*, 2011:1–10, 2011. 2

[61] Y. Zhou, A. Rangarajan, and P. D. Gader. An integrated approach to registration and fusion of hyperspectral and multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3020–3033, 2020. 2

[62] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

**Supplementary Material**

In this supplementary material, we provide

- results for scaling factor $\times 4$ under all the six metrics,
- results for scaling factor $\times 8$,
- more visual results.

**Further results for scaling factor $\times 4$**

Due to space limitation, only results under three metrics, i.e., RMSE, MPSNR, and ERGAS, are reported in the main paper. Here, we report the results under all six considered metrics, i.e., RMSE, CC, MPSNR, MSSIM, ERGAS, and SAM. For the case of scaling factor $\times 4$ and in the semi-supervised setting, the results on the CAVE dataset, the Harvard dataset, the NTIRE2020 dataset are shown in Table 6, Table 7 and Table 8, respectively. The results under the full-supervision setting on the CAVE dataset and on the Harvard dataset are reported in Table 9 and Table 10.

These tables show that our method outperforms other comparison methods by a large margin under all six metrics. All our three contributions are useful and their combination yields the best results. The conclusions we have in the main paper hold for all the six metrics.

**Results for scaling factor $\times 8$**

We also provide results for scaling factor $\times 8$. The results on the CAVE, the Harvard, and the NTIRE2020 datasets in the semi-supervised setting are shown in Table 11, Table 12, and Table 13, respectively. It is evident from these tables that our method also outperforms other methods significantly and consistently for scaling factor $\times 8$. The same trend is observed for both $\times 4$ and $\times 8$ that all our three contributions are useful and their combination yields the best results. This demonstrates the applicability of our method across different scaling factors.

---

1The results on the NTIRE2020 dataset have not completed before the deadline unfortunately, but the obtained results show that the variants of our method with only a subset of our contributions already outperform all the competing methods.

---

**More visual results**

We also provide more visual results on the NTIRE2020 datasets. For visualization, we use the same method as in the main paper. More specifically, we sample the 5th band, the 15th band and the 25th band of the hyperspectral image and assemble them together as an RGB image for visualization. The results of all methods on two different images for scaling factor $\times 8$ are provided in Fig. 4 and Fig. 5. To facilitate the comparison, we also show the error maps of all methods. The values in the error maps are the L2 distance between the predicted pixel values and the ground-truth pixel values, averaged over the three bands. It is clear from the visual results that our method generates better results than other methods. For instance, it produces sharper boundaries and less artefact.
| Components | Metrics | Methods |
|------------|---------|---------|
| RGB_SR     | Spec Mixup | SSL |
|            | RMSE ↓ | CC ↑ | MPSNR ↑ | MSSIM ↑ | ERGAS ↓ | SAM ↓ |
| Ours       | 0.01282 | 0.99255 | 41.38106 | 0.96382 | 3.99731 | 6.17381 |
| Ours       | 0.01191 | 0.99343 | 42.01763 | 0.96661 | 3.63288 | 4.87344 |
| Ours       | 0.01246 | 0.99321 | 42.08879 | 0.96508 | 3.54516 | 3.78728 |
| Ours       | 0.01173 | 0.99364 | 42.57401 | 0.96717 | 3.40209 | 3.59636 |
| Ours (final) | 0.01138 | 0.99389 | 42.74453 | 0.96806 | 3.31609 | 3.57381 |
| BicubicInt. | - | - | - | 0.01856 | 0.98682 | 38.73800 | 0.94197 | 5.27190 | 4.17591 |
| GDRRN [35] | - | - | - | 0.02048 | 0.98479 | 37.59559 | 0.92286 | 5.76242 | 7.77988 |
| 3DFCNN [38] | - | - | - | 0.01686 | 0.97777 | 38.22377 | 0.94694 | 8.81439 | 8.59455 |
| SSPSR [26] | - | - | - | 0.01349 | 0.99115 | 41.46318 | 0.96104 | 3.91284 | 4.07686 |
| MCNet [34] | - | - | - | 0.01362 | 0.98964 | 41.25849 | 0.96029 | 4.16633 | 4.65331 |

Table 6: Results of our method and other comparison methods on the CAVE dataset in the semi-supervised setting for the ×4 case.

| Components | Metrics | Methods |
|------------|---------|---------|
| RGB_SR     | Spec Mixup | SSL |
|            | RMSE ↓ | CC ↑ | MPSNR ↑ | MSSIM ↑ | ERGAS ↓ | SAM ↓ |
| Ours       | 0.01411 | 0.95826 | 40.42362 | 0.92569 | 2.30297 | 2.58523 |
| Ours       | 0.01366 | 0.95959 | 40.69577 | 0.92768 | 3.11316 | 2.56292 |
| Ours       | 0.01375 | 0.95951 | 40.83699 | 0.92786 | 3.12398 | 2.53555 |
| Ours       | 0.01348 | 0.96020 | 40.80286 | 0.92853 | 3.07281 | 2.54400 |
| Ours (final) | 0.01351 | 0.96013 | 40.81661 | 0.92864 | 3.06886 | 2.55412 |
| BicubicInt. | - | - | - | 0.01675 | 0.94994 | 38.89758 | 0.90925 | 3.80698 | 2.61754 |
| GDRRN [35] | - | - | - | 0.01655 | 0.93686 | 38.24997 | 0.90996 | 4.72408 | 3.26193 |
| 3DFCNN [38] | - | - | - | 0.01578 | 0.95095 | 39.30286 | 0.91648 | 3.66444 | 2.72254 |
| SSPSR [26] | - | - | - | 0.01422 | 0.95789 | 40.34669 | 0.92474 | 3.22915 | 2.59564 |
| MCNet [34] | - | - | - | 0.01463 | 0.95749 | 40.18194 | 0.92286 | 3.27906 | 2.64800 |

Table 7: Results of our method and other comparison methods on the Harvard dataset in the semi-supervised setting for the ×4 case.

| Components | Metrics | Methods |
|------------|---------|---------|
| RGB_SR     | Spec Mixup | SSL |
|            | RMSE ↓ | CC ↑ | MPSNR ↑ | MSSIM ↑ | ERGAS ↓ | SAM ↓ |
| Ours       | 0.01675 | 0.99049 | 37.76302 | 0.93697 | 2.36922 | 1.35066 |
| Ours       | 0.01583 | 0.99135 | 38.21667 | 0.94195 | 2.27241 | 1.48222 |
| Ours       | 0.01342 | 0.99182 | 38.56335 | 0.94279 | 2.16249 | 1.29018 |
| Ours       | 0.01515 | 0.99198 | 38.75436 | 0.94310 | 2.11092 | 1.23994 |
| Ours (final) | 0.01347 | 0.99186 | 38.55183 | 0.94279 | 2.16476 | 1.29852 |
| BicubicInt. | - | - | - | 0.02353 | 0.98297 | 34.74012 | 0.90050 | 3.19014 | 3.89655 |
| GDRRN [35] | - | - | - | 0.02484 | 0.97852 | 33.94214 | 0.89828 | 3.68688 | 3.54982 |
| 3DFCNN [38] | - | - | - | 0.02080 | 0.98629 | 35.83412 | 0.91539 | 2.82781 | 1.66409 |
| SSPSR [26] | - | - | - | 0.01721 | 0.98991 | 37.58708 | 0.93317 | 2.37159 | 1.53442 |
| MCNet [34] | - | - | - | 0.01613 | 0.99111 | 38.45578 | 0.93931 | 2.19636 | 1.45320 |

Table 8: Results of our method and other comparison methods on the NTIRE2020 dataset in the semi-supervised setting for the ×4 case.
| Components | Methods | RGB_SR | Spec Mixup | RMSE | CC | MPSNR | MSSIM | ERGAS | SAM |
|------------|---------|--------|------------|------|----|-------|-------|-------|-----|
|            | Ours    |        |            | 0.01196 | 0.99350 | 42.38359 | 0.96631 | 3.45903 | 4.00592 |
|            | Ours    |        |            | 0.01109 | 0.99422 | 42.73668 | 0.96869 | 3.35884 | 4.17068 |
|            | Ours    |        |            | 0.01134 | 0.99397 | 42.88402 | 0.96798 | 3.28051 | **3.47102** |
|            | Ours    |        |            | 0.01046 | 0.99447 | **43.32421** | **0.96980** | **3.11799** | **3.6885** |

Table 9: Results of our method and other comparison methods on the CAVE dataset in the fully supervised case for the $×4$ case.

| Components | Methods | RGB_SR | Spec Mixup | RMSE | CC | MPSNR | MSSIM | ERGAS | SAM |
|------------|---------|--------|------------|------|----|-------|-------|-------|-----|
|            | Ours    |        |            | 0.01344 | 0.96101 | 40.91014 | 0.92836 | 3.01039 | 2.50704 |
|            | Ours    |        |            | 0.01325 | 0.96165 | 41.03709 | 0.92949 | 2.96443 | 2.49562 |
|            | Ours    |        |            | 0.01317 | 0.96200 | **41.08568** | **0.93056** | **2.95718** | **2.49771** |
|            | Ours    |        |            | 0.01321 | 0.96178 | 41.05925 | 0.93016 | 2.96496 | 2.49897 |

Table 10: Results of our method and other comparison methods on the Harvard dataset in the fully supervised case for the $×4$ case.

| Components | Methods | RGB_SR | Spec Mixup | SSL | RMSE | CC | MPSNR | MSSIM | ERGAS | SAM |
|------------|---------|--------|------------|-----|------|----|-------|-------|-------|-----|
|            | Ours    |        |            |     | 0.02459 | 0.97233 | 35.89888 | 0.90591 | 7.11644 | 7.50539 |
|            | Ours    |        |            |     | 0.02300 | 0.97545 | 36.50494 | 0.91358 | 6.62181 | 6.89221 |
|            | Ours    |        |            |     | 0.02339 | 0.97705 | **36.64775** | **0.91078** | **6.45075** | **6.30468** |
|            | Ours    |        |            |     | 0.02209 | 0.97923 | 37.14066 | 0.91748 | 6.14668 | 6.24141 |
|            | Ours    |        |            |     | **0.02237** | **0.97732** | **36.87625** | **0.91685** | **6.34954** | **6.31945** |
|            | Ours (final) |        |            |     | 0.02154 | **0.99638** | **37.35599** | **0.92070** | **6.00466** | **5.62054** |

Table 11: Results of our method and other comparison methods on the CAVE dataset in the semi-supervised setting for the $×8$ case.
Figure 4: Exemplar results of $\times 8$ by our method and all comparison methods. The error is L2 distance to the ground-truth pixel values, averaged over the three bands.
Figure 5: Exemplar results of $\times 8$ by our method and all comparison methods. The error is L2 distance to the ground-truth pixel values, averaged over the three bands.
| Methods     | Components | Metrics       | Components | Metrics       |
|-------------|------------|---------------|------------|---------------|
|             | RGB_SR     | Spec Mixup    | SSL        | RMSE ↓ CC ↑   |
| Ours        |            |               |            | 0.02263       |
| Ours        |            |               |            | 0.02117       |
| Ours        |            |               |            | 0.02117       |
| Ours (final)|            |               |            | 0.02100       |
| BicubicInt. |            |               |            | 0.02495       |
| GDRRN [35]  |            |               |            | 0.02389       |
| 3DFCNN [38] |            |               |            | 0.02379       |
| SSPSR [26]  |            |               |            | 0.02282       |
| MCNet [34]  |            |               |            | 0.02348       |
|             |            |               |            | 0.02969       |
| Ours        |            |               |            | 0.02839       |
| Ours        |            |               |            | 0.02800       |
| Ours        |            |               |            | 0.02799       |
| BicubicInt. |            |               |            | 0.03961       |
| GDRRN [35]  |            |               |            | 0.03596       |
| 3DFCNN [38] |            |               |            | 0.28514       |
| SSPSR [26]  |            |               |            | 0.38575       |
| MCNet [34]  |            |               |            | 0.03268       |

Table 12: Results of our method and other comparison methods on the Harvard dataset in the semi-supervised setting for the ×8 case.

| Methods     | Components | Metrics       | Components | Metrics       |
|-------------|------------|---------------|------------|---------------|
|             | RGB_SR     | Spec Mixup    | SSL        | RMSE ↓ CC ↑   |
| Ours        |            |               |            | 0.02263       |
| Ours        |            |               |            | 0.02117       |
| Ours        |            |               |            | 0.02117       |
| Ours (final)|            |               |            | 0.02100       |
| BicubicInt. |            |               |            | 0.02495       |
| GDRRN [35]  |            |               |            | 0.02389       |
| 3DFCNN [38] |            |               |            | 0.02379       |
| SSPSR [26]  |            |               |            | 0.02282       |
| MCNet [34]  |            |               |            | 0.02348       |
|             |            |               |            | 0.02969       |
| Ours        |            |               |            | 0.02839       |
| Ours        |            |               |            | 0.02800       |
| Ours        |            |               |            | 0.02799       |
| BicubicInt. |            |               |            | 0.03961       |
| GDRRN [35]  |            |               |            | 0.03596       |
| 3DFCNN [38] |            |               |            | 0.28514       |
| SSPSR [26]  |            |               |            | 0.38575       |
| MCNet [34]  |            |               |            | 0.03268       |

Table 13: Results of our method and other comparison methods on the NTIRE2020 dataset in the semi-supervised setting for the ×8 case.