HSCNN+: Advanced CNN-Based Hyperspectral Recovery from RGB Images

Zhan Shi* Chang Chen* Zhiwei Xiong† Dong Liu Feng Wu
University of Science and Technology of China

Abstract

*Hyperspectral recovery from a single RGB image has seen a great improvement with the development of deep convolutional neural networks (CNNs). In this paper, we propose two advanced CNNs for the hyperspectral reconstruction task, collectively called **HSCNN+**. We first develop a deep residual network named **HSCNN-R**, which comprises a number of residual blocks. The superior performance of this model comes from the modern architecture and optimization by removing the hand-crafted upsampling in HSCNN. Based on the promising results of HSCNN-R, we propose another distinct architecture that replaces the residual block by the dense block with a novel fusion scheme, leading to a new network named **HSCNN-D**. This model substantially deepens the network structure for a more accurate solution. Experimental results demonstrate that our proposed models significantly advance the state-of-the-art. In the NTIRE 2018 Spectral Reconstruction Challenge, our entries rank the 1st (HSCNN-D) and 2nd (HSCNN-R) places on both the "Clean" and "Real World" tracks. (Codes are available at [clean-r], [realworld-r], [clean-d], and [realworld-d].)*

1. Introduction

Hyperspectral imaging aims to obtain the spectrum reflected or emitted from a scene or an object. The spectral characteristic has been proven useful in many fields, ranging from remote sensing to medical diagnosis and agriculture [16, 17, 18]. In recent years, the hyperspectral image begins to be applied to various computer vision tasks, such as image segmentation, face recognition, and object tracking [31, 28, 33]. Thus hyperspectral imaging has received an increasing amount of research attention and efforts [34, 35, 15, 38, 4, 39].

However, since conventional acquisition of high quality hyperspectral images need to capture three-dimensional signals with a two-dimensional sensor, trade-offs between spectral and spatial/temporal resolutions are inevitable [6, 14], which severely limits the application scope of hyperspectral images. To overcome these difficulties and enable hyperspectral image acquisition in dynamic conditions, a number of solutions based on compressed sensing are proposed by encoding the spectral information, which transfer the cost from capture to computational reconstruction [26, 34, 35, 36]. Still, the hardware systems and reconstruction algorithms are of high complexity. As an alternative solution, it would be great if we can obtain the hyperspectral image through a ubiquitous RGB camera. This is not only convenient to implement but also affordable.

Hyperspectral recovery from RGB images is a severely ill-posed problem, since much information is lost after integrating the hyperspectral radiance into RGB values. Existing methods can be roughly divided into two categories. The first one is to design a specific system based on the ordinary RGB cameras. In order to reduce the lost information and better recover the hyperspectral image, the approaches of exploiting time-multiplexed illumination source, multiple color cameras, and a tube of faced reflectors are present to complete the reconstruction [15, 38, 30]. Nevertheless, such kind of methods rely on rigorous environment conditions and/or extra equipments.

As there is a high correlation between RGB values and their corresponding hyperspectral radiance [9], the second category of methods manage to exploit this correlation from a large number of training data and directly model the mapping between RGB and hyperspectral images. Since this mapping is highly non-linear, learning-based methods are generally used to model it [4, 1, 2]. Recently, with the success of deep learning in many computer vision tasks, CNN-based methods are also introduced to this task [13, 39, 3].

Among these methods, Xiong *et al.* [39] proposed a unified deep learning framework, *i.e.*, HSCNN, for hyperspectral recovery from both RGB and compressive measurements, which achieved state-of-the-art results on the ICVL dataset [4]. However, the upsampling operation in HSCNN requires the knowledge of an explicit spectral response function that corresponds to the integration of hyperspectral radiance to RGB values. It thus restricts the applicability of HSCNN when the spectral response function is unknown or difficult to obtain in practice. What is

*These authors contribute equally to this work.
†Correspondence should be addressed to zwxiong@ustc.edu.cn.

more, HSCNN failed to achieve an improved performance with the growth of the network depth, which is not satisfying when more accurate solutions are pursued.

To solve these problems, we first investigate the effect of the hand-crafted upsampling in HSCNN and replace it by a simple convolution layer, which thus relieves the requirement of the spectral response function in advance. Moreover, we replace the plain convolution layers in HSCNN by modern residual blocks to further boost its performance, deriving the HSCNN-R model. Based on the promising results of HSCNN-R, we propose another distinct architecture that replaces the residual block by the dense block with a novel fusion scheme, deriving the HSCNN-D model. Compared with HSCNN-R, this new model substantially deepens the network structure for a more accurate solution, at the cost of moderately increased computational complexity.

Our contributions can be summarized as follows:

- We develop the HSCNN-R network, which optimizes the architecture of HSCNN by removing the hand-crafted upsampling and utilizing residual blocks.
- We further propose the HSCNN-D network based on a densely-connected structure with a novel fusion scheme, which deepens the network structure and provides a more accurate solution.
- State-of-the-art results on hyperspectral recovery from RGB images, which rank the 1st and 2nd places on both the "Clean" and "Real World" tracks in the NTIRE 2018 Spectral Reconstruction Challenge [5].

2. Related Work

Hyperspectral image acquisition. Conventional imaging spectrometers usually operate in the scanning manner, which simply trades the temporal resolution for the spectral resolution [7]. To overcome this trade-off and enable hyperspectral acquisition for dynamic scenes, three representative techniques relying on computational reconstruction were proposed in the past decades. They are computed tomographic imaging spectrometry (CTIS) [11], prism-mask spectral video imaging system (PMVIS) [8], and coded aperture snapshot spectral imagers (CASSI) [34, 37]. Still, these systems are of high complexity in terms of both hardware implementation and reconstruction algorithm.

RGB to hyperspectral conversion is a relatively low cost approach to acquire hyperspectral images. However, much information is lost during the process of spectral integration, which makes the problem severely ill-posed. In this context, Goel *et al.* [15] used a conventional RGB camera to capture a scene illuminated by the time-multiplexed light source, and the scene reflectance is estimated across a number of wavelengths using a computational approach. Instead of using active illumination, Oh *et al.* [38] proposed a framework for reconstructing hyperspectral images by using multiple consumer-level digital cameras, where the hyperspectral image is reconstructed from different RGB measurements generated by the cameras with different spectral sensitivities. Takatani *et al.* [30] proposed a low cost method by augmenting a conventional camera with a tube of reflectors, yet it remains a trade-off between the spatial and spectral resolution.

Compared with the above methods that require rigorous environment conditions and/or extra equipments, hyperspectral recovery directly from a single RGB image is more favorable. To this end, Arad *et al.* [4] leveraged the hyperspectral prior from a large number of training images to create a sparse dictionary, which facilitates the subsequent hyperspectral reconstruction. Later, Aeschbacher *et al.* [1] introduced A+ [32] from super-resolution to further improve Arad's method. Alternatively, Akhtar and Mian [2] replaced the dictionary learning with Gaussian processes and utilized clustering in data processing. Leveraging the powerful capability of deep neural networks, Galliani *et al.* [13] and Xiong *et al.* [39] introduced CNNs from the semantic segmentation and super-resolution to this task and achieved promising results. On the other hand, Alvarez-Gila *et al.* [3] exploited GAN to learn a generative model of the joint spectral-spatial distribution of the natural hyperspectral image. Compared with the above methods, our proposed models offer more efficient and accurate solutions for improving the fidelity of hyperspectral reconstruction.

Convolutional neural network. CNNs play an important role in the current computer vision society, where an efficient and powerful framework can be generalized in many fields. He *et al.* [20] presented a deep residual network for image classification. The architecture of residual representation eases the training of deep networks, which has also made a great success in single image super-resolution [25]. To further utilize the information flow between layers in the network, Huang *et al.* [21] proposed a densely-connected convolutional network which connects all layers directly with each other. Our proposed HSCNN-R introduces the residual block into hyperspectral reconstruction, which shows an excellent performance on this task. To better exploit the correlation between different spectral bands, we further propose HSCNN-D by adopting the dense structure with a novel fusion scheme, which produces even more accurate results.

3. Our Proposed Method

3.1. Overview of HSCNN

HSCNN is one of the first CNN-based methods for hyperspectral recovery from a single RGB image. It introduces the VDSR network [22] originally used for single image super-resolution into hyperspectral reconstruction. Similar to the bicubic interpolation generally adopted in

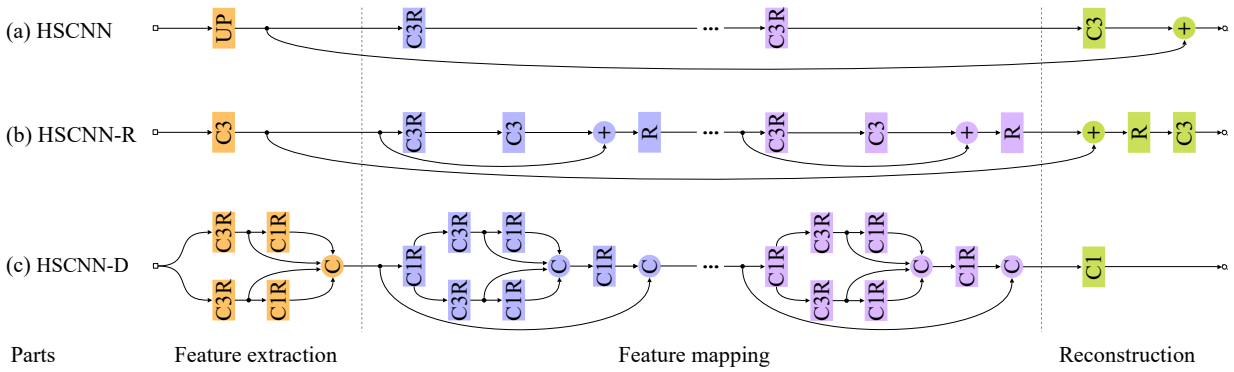

Figure 1. Network structures of our proposed models and the baseline HSCNN. The "C" with a rectangular block denotes the convolution, and the following "1" and "3" denote the kernel size (*i.e.*, 1×1 and 3×3 respectively). The "R" represents the ReLU activation function. And the "C" with a circular block denotes the concatenation.

super-resolution, HSCNN first upsamples the RGB input to a hyperspectral image using a spectral interpolation algorithm [29]. This spectrally upsampled image has the same number of bands as the expected hyperspectral output. Then, the network takes the spectrally upsampled image as input and predicts the missing details (residuals) by learning a stack of convolutional layers. The structure of HSCNN is depicted in Fig. 1(a).

However, since the spectral upsampling requires the knowledge of the spectral response function, HSCNN cannot be applied in the situation when the spectral response function is unknown. Actually, the explicit upsampling is suboptimal and can be replaced by a simple convolution layer. Also, according to the experimental results of HSCNN, it fails to achieve an improved performance with the growth of the network depth, which is not satisfying when more accurate solutions are pursued. To solve these problems, we propose several new models that greatly improve the capability of HSCNN.

3.2. HSCNN-u

To investigate the effect of the upsampling operation in HSCNN, we design an intermediate network, called HSCNN-u, which learns an end-to-end mapping from the pairs of RGB/hyperspectral images. HSCNN-u replaces the spectral upsampling operation by a simple convolutional layer with filter size of 1×1 and without bias. In this way, the convolution layer can be regraded as a learned spectral upsampling, since the upsampling method used in HSCNN is also linear. The following structure of HSCNN-u is the same as HSCNN. As demonstrated by the experimental results in Sec. 4.1, the spectral interpolation algorithm used in HSCNN is suboptimal. Embedding the interpolation into the network in the form of convolution not only achieves a slightly better performance but also relieves the requirement of the spectral response function in advance. The latter greatly promotes the applicability of HSCNN, since the spectral response function varies for different camera models (even under different configurations for the same camera model) and is not always known in practice. Moreover, for realworld RGB images that are often in the compressed formats such as JPEG, the actual spectral response function may be even more complicated and difficult to derive. In this case, HSCNN-u still works while HSCNN cannot.

3.3. HSCNN-R

While HSCNN-u gets rid of the explicit upsampling operation, it still inherits the network structure of HSCNN. To boost the performance of hyperspectral reconstruction, we propose the HSCNN-R model that replaces the plain convolutional architecture in HSCNN by the modern residual block [20] while remaining the global residual leanring. The residual block is defined as

$$y = F(x, W_l) + x, \tag{1}$$

where x and y are the input and output of the block, W_l represents the weight matrix of the l-th block, and $F(\cdot)$ denotes the residual mapping to be learned which comprises two convolutional and one ReLU layers [27]. The addition is element-wise and performed by a shortcut connection.

The structure of HSCNN-R is depicted in Fig. 1(b), where the filter size of convolution is uniformly set to 3×3. Similarly to HSCNN-u, it also replaces the upsampling operation by a convolutional layer. As can be seen in Sec. 4.1, HSCNN-R achieves a significantly improved performance under the same network depth as HSCNN, which proves the superiority of the new architecture. On the other hand, while simply increasing the filter number can slightly improve the performance, the result will become worse when the network depth exceed 40. This should be caused by the vanishing of gradient during the back-propagation.

Loss function. CNN-based methods for image restoration usually adopt the mean square error (MSE) as the loss function [12, 22, 10] during training, which has also been

applied to the hyperspectral reconstruction task [4, 39]. However, for hyperspectral images, the luminance level usually varies significantly among different bands, and the same deviation in the pixel value may have different influence to the bands with different luminance levels. It thus makes the MSE loss generate a bias towards the bands with high luminance levels, which is not desired because each band matters equally. To address this issue, we adopt the mean relative absolute error (MRAE) as the new loss function for training the HSCNN-R model, which brings notable performance gain as demonstrated in Sec. 4.1.

Training and testing. We train the HSCNN-R model using RGB patches of size 50×50 and their corresponding hyperspectral cubes. Batch size is set to 64 and the optimizer is Adam [23] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial learning rate is 2×10^{-4} with the polynomial function as the decay policy. We set $power = 1.5$ empirically for training efficiently, and we stop training at 1000 epochs. The algorithm proposed in [19] is adopted for initializing the weights in each convolutional layer. We implement the proposed network with the PyTorch framework and train it on a 1080Ti GPU. It takes about 60 hours to train a typical HSCNN-R model with 16 residual blocks (equivalent to a depth of 34) and each with a filter number of 64. During testing, we feed a full-size RGB image (with a typical resolution of 1300×1392) into the model to obtain a hyperspectral image with 31 bands at the same spatial resolution. Hence, at least 12G GPU memory is required for inference.

3.4. HSCNN-D

Besides the residual connection described above, we also investigate the densely-connected structure as an alternative solution, *i.e.*, the HSCNN-D model, which is shown in Fig. 1(c). To the best of our knowledge, it is the first time to apply the dense structure to the hyperspectral reconstruction task. The dense connection enables the l-th layer to receive the features from all preceding layers (*i.e.*, f_0, ..., f_{l-1}) as input, which can be represented as

$$f_l = g_l([f_0, f_1, ..., f_{l-1}]), \tag{2}$$

where $g_l(\cdot)$ denotes the l-th convolutional layer and $[f_0, f_1, ..., f_{l-1}]$ stands for the concatenation of the features output from preceding layers. The dense structure can substantially alleviate the vanishing of gradient during training. It is demonstrated in experiments that the propagation issue brought by the increasing depth of the network can thus be addressed. In other words, the HSCNN-D model overcomes the limitation of HSCNN and HSCNN-R. With a much deeper network, HSCNN-D can offer higher reconstruction fidelity provided sufficient computing resources.

In addition to the inherent advantage of convergence, the dense structure has another characteristic suitable for hyperspectral reconstruction. Specifically, given a three-channel (*i.e.*, RGB) image as input, we aim to reconstruct a hyperspectral image with multi-channels (typically tens of bands in practice) as output. The main gap between the input and output lies in the decrease of channels. The concatenation operator adopted in each dense block explicitly increases the channel number which has the potential to learn a more effective model for this inverse problem.

Feature extraction	Feature mapping	Reconstruction
$\begin{bmatrix} C(3\times3\times16/16) \\ C(1\times1\times16/16) \end{bmatrix}$	$\left\{ \begin{bmatrix} C(1\times1\times64) \\ C(3\times3\times16/16) \\ C(1\times1\times8/8) \\ C(1\times1\times16) \end{bmatrix} \right\} \times 38$	$C(1\times1\times1)$

Table 1. A typical setting of hyper-parameters for the HSCNN-D model. $C(\cdot)$ stands for the convolution with (kernel size $\times$ kernel size $\times$ filter number). $[\cdot]\times$ and $\{\cdot\}\times$ stand for concatenation operators with certain blocks ($\times 1$ is omitted). And the symbol "/" denotes the parallel operation for path-widening fusion.

Path-widening fusion. Inspired by [40], we further propose a path-widening fusion based on the dense structure, as shown in Fig. 1(c). Intuitively, the operation of fusion is similar to the group convolution [24]. The difference is that we expand the number of forward paths to enlarge the ensembling size for higher capacity. With the concatenation, we aggregate both the fused and the preceding features using a 1×1 convolution in the dense block. Experimental results demonstrate that our proposed fusion scheme can further promote the performance of the HSCNN-D model (as detailed in Sec. 4.1).

Training and testing. We use TensorFlow and the Adam [23] solver for optimization with the momentum factor set as 0.9 and the coefficient of weight decay (L_2-Norm) as 0.0001. The learning rate is decayed exponentially from 0.001 to 0.0001. We stop training when no notable decay of training loss is observed. The algorithm proposed in [19] is adopted for initializing the weights except the last layer. Specifically, the last layer for reconstruction is initialized by the random weights drawn from Gaussian distributions with $\sigma = 0.001$. The biases in each convolutional layer are initialized to zero. The loss function, patch size, and batch size are set the same as in HSCNN-R. It takes about 38 hours on 8 Tesla M40 GPUs to train a typical HSCNN-D model with 38 dense blocks (equivalent to a depth of 160) and each with the filter numbers listed in Table 1. During testing, a general CPU along with at least 32G memory is required for inference.

4. Experimental Results

The experiments are conducted strictly following the instructions of the NTIRE 2018 Spectral Reconstruction Challenge [5]. The given training dataset in the challenge

Data Set	Sparse Coding [4] MRAE / RMSE	HSCNN [39] MRAE / RMSE	HSCNN-u MRAE / RMSE	HSCNN-R MRAE / RMSE	HSCNN-D MRAE / RMSE
Clean	0.0816 / 49.217	0.0190 / 17.006	0.0186 / 16.653	0.0145 / 13.911	0.0135 / 13.128
Real World	- / -	- / -	0.0342 / 24.987	0.0306 / 23.457	0.0293 / 22.935
Running Time (s)	112.93 (CPU)	0.51 (GPU)	0.43 (GPU)	0.96 (GPU)	77.28 (CPU)

Table 2. Comparison between our proposed methods and previous methods. The official validation set of NTIRE 2018 Spectral Reconstruction Challenge is adopted for evaluation. The filter number of each layer in HSCNN-R is 64 and the depth is 34. For HSCNN-D, the filter numbers of each layer are listed in Table 1 and the depth is 160. Red color indicates the best performance and blue color indicates the second best one. We calculate the average running time using an i7-6850K CPU or a 1080Ti GPU.

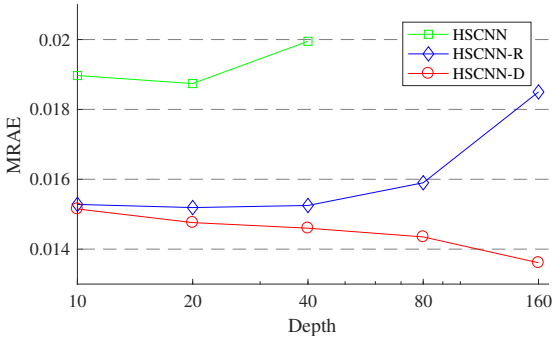

Figure 2. The influence of depth on the network performance.

contains 256 pairs of images (203 pairs in "Train1" and 53 pairs in "Train2"). In the development phase, we reserve 10 pairs of images from "Train2" as our own validation set for parameter tuning and adopt the others as the training set. In the test phase, we use all 256 pairs of images for training and the official validation set (5 pairs, hyperspectral images available in the test phase). No additional preprocessing or postprocessing is needed for both the "Clean" track and the "Real World" track, which demonstrates the robustness and generalizability of our proposed models.

Let $I_R^{(i)}$ and $I_G^{(i)}$ denote the i^{th} pixel of the reconstructed and groundtruth hyperspectral images respectively, and n is the total number of pixels. Following the scoring script provide by the challenge, we adopt the MRAE and root mean square error (RMSE) as the evaluation metrics

$$MRAE = \frac{1}{n}\sum_{i=1}^{n}(|I_R^{(i)} - I_G^{(i)}|/I_G^{(i)}), \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(I_R^{(i)} - I_G^{(i)})^2}. \tag{4}$$

4.1. Ablation Experiments

Compared with the prior art HSCNN [39], our proposed models integrate the concepts of residual block and dense block respectively to derive two distinct solutions. Extensive ablation experiments are conducted on the official validation set to evaluate them respectively.

Upsampling and loss. As described in Sec. 3.2, we conduct an experiment to verify that replacing the hand-crafted upsampling by a simple convolutional layer contributes to a even better performance. As listed in Table 2, HSCNN-u achieves 2.1% decrease in MRAE compared with HSCNN for the "Clean" track. More importantly, it makes HSCNN-u capable of dealing with the images from the "Real World" track, where the spectral response function is unknown. In terms of the loss function discussed in Sec. 3.3, the HSCNN-u with MRAE loss achieves 12.9% decrease compared to the one with MSE loss, which demonstrates the superiority of using MRAE loss.

Fusion scheme for HSCNN-D. As described in Sec. 3.4, we conduct another experiment to investigate the contribution of our proposed fusion scheme in the HSCNN-D model. Compared with a dense network without fusion, our HSCNN-D brings 8.7% decrease in MRAE for the "Clean" track. Note that, we keep the parameter number of the two networks the same for a fair comparison. Experimental results demonstrate that a superior performance can be achieved with our proposed fusion scheme.

Network depth and width. The depth and width (*i.e.*, filter number) are two important hyper-parameters in the neural network, which jointly determine its basic capacity. However, a deeper or wider network does not always yield a better performance. To this end, we conduct a comprehensive experiment to investigate the relationship between the network performance and these two hyper-parameters. Other hyper-parameters such as the decay policy of learning rate and the initialization method are kept the same to eliminate their influence.

Fig. 2 shows the influence of depth on the network performance. Unsurprisingly, the MRAE of HSCNN [39] increases quickly with the growth of depth due to the convergence problem brought by the plainly-connected structure. HSCNN-R behaves better under the same depth as HSCNN but its performance also degrades once the depth exceeds 40. In contrast, HSCNN-D benefits from the increasing depth. Specifically, the network with a depth of 160 has 6.7% decrease in MRAE than the one with a depth of 40. In terms of the filter number, more filters contributes to a bet-

Model	HSCNN-R Clean / RealW.	HSCNN-D Clean / RealW.
Single model	0.0145 / 0.0306	0.0135 / 0.0293
+Self-ensemble	0.0142 / 0.0303	0.0134 / 0.0291
+Model-ensemble	0.0134 / 0.0297	0.0131 / 0.0288

Table 3. Quantitative results of ensemble methods. The baseline of ensemble is the single model specified in Table 2. The MRAE metric is used on the official validation set of the challenge.

Method	Track1: Clean		Track2: Real World	
	MRAE	RMSE	MRAE	RMSE
HSCNN-D	0.0138	14.452	0.0310	24.067
HSCNN-R	0.0139	13.987	0.0321	25.014
3rd method	0.0152	16.192	0.0333	27.104
4th method	0.0153	16.076	0.0335	26.450
5th method	0.0165	16.923	0.0345	26.977

Table 4. Comparison between our proposed models and three competitors. The final test set of NTIRE 2018 Spectral Reconstruction Challenge is adopted for evaluation. Red color indicates the best performance and blue color indicates the second best one.

ter performance. For instance, the HSCNN-R model with 256 filters in each layer has about 4% decrease in MRAE than the one with 64 filters.

Ensemble methods. We adopt two types of ensemble to further enhance the performance. The first one is called self-ensemble. We firstly flip the input image left/right and feed it into the network to obtain a mirrored output. Then we average the mirrored output and the original output into the target result. The second one is called model-ensemble, whose result is the linear combination of three models with different depths, filter numbers, or initializations. Specifically, two additional models are trained for model-ensemble of HSCNN-R (256 filters in each layer and a depth of 20/30) besides the one specified in Table 2. For HSCNN-D, two additional models with a depth of 160/240 are trained for model-ensemble. Experimental results listed in Table 3 demonstrate the performance improvement achieved using these two ensemble methods.

4.2. Comparison with State-of-the-Art Methods

Comparison with existing models. We adopt two representative models for comparison, *i.e.*, sparse coding [4] and HSCNN [39], in the "Clean" track. For the "Real World" track, HSCNN does not work so we use HSCNN-u for comparison instead. All these models are trained using the whole training set of the challenge, and the official validation set is adopted for evaluation. The quantitative results are listed in Table 2. As can be seen, our proposed models significantly outperform previous models. Specifically, HSCNN-R achieves 22.0% decrease in MRAE compared with HSCNN. HSCNN-D achieves the best performance, which further brings 6.9% decrease in MRAE than HSCNN-R.

Comparison with competitors in the challenge. Our proposed models rank the 1st and 2nd places in the NTIRE 2018 Spectral Reconstruction Challenge for both the "Clean" and "Real World" tracks. We list the results from the top 5 methods on the final test set in Table 4. Compared with the 3rd method, HSCNN-D achieves 9.2% decrease in MRAE and HSCNN-R is 8.6% in the "Clean" track. In the "Real World" track, the improvement over the 3rd method in MRAE is 6.9% for HSCNN-D and 3.6% for HSCNN-R respectively. In terms of the RMSE metric, our proposed models also give top performance, and HSCNN-R even outperforms HSCNN-D in the "Clean" track.

Visual and spectral signature comparison. To evaluate the perceptual quality of hyperspectral reconstruction, we show two examples of error maps in Fig. 3 and Fig. 4 for the "Clean" and "Real World" tracks. As can be seen, the errors of HSCNN-R and HSCNN-D are notably smaller than the baseline sparse coding and HSCNN in the "Clean" track and also smaller than HSCNN-u in the "Real World" track. For a more intuitive evaluation, the spectral signatures of four selected spatial points from the above examples are shown in Fig. 5. Compared with sparse coding, the reconstructed results from CNNs are much closer to the groundtruth. Moreover, HSCNN-R and HSCNN-D give higher reconstruction fidelity compared with HSCNN and HSCNN-u.

Running time. In addition to performance, we also evaluate the computational complexity of each model. Note that, we conduct the inference of HSCNN-D using a general CPU, due to the limitation of GPU memory. For the sparse coding method, we use the provided MATLAB scripts for evaluation. As listed in Table 2, HSCNN-D costs about 77 seconds per image (CPU time) to achieve the best performance, while HSCNN-R only takes 0.96 seconds (GPU time) to achieve a competitive performance. It is thus a trade-off of performance and speed between the two models. According to the requirement in practical applications, HSCNN-D and HSCNN-R provide alternative solutions either in pursuit of performance or speed.

5. Conclusion

In this paper, we present two advanced CNN-based models for hyperspectral recovery from a single RGB image. Based on the pioneer work of HSCNN, we first derive the HSCNN-R model by introducing the modern residual block. To enable hyperspectral reconstruction when the spectral response function is unknown, we replace the hand-crafted upsampling operation by a convolutional layer. Based on the promising results of HSCNN-R, we further propose the HSCNN-D model which replaces the resid-

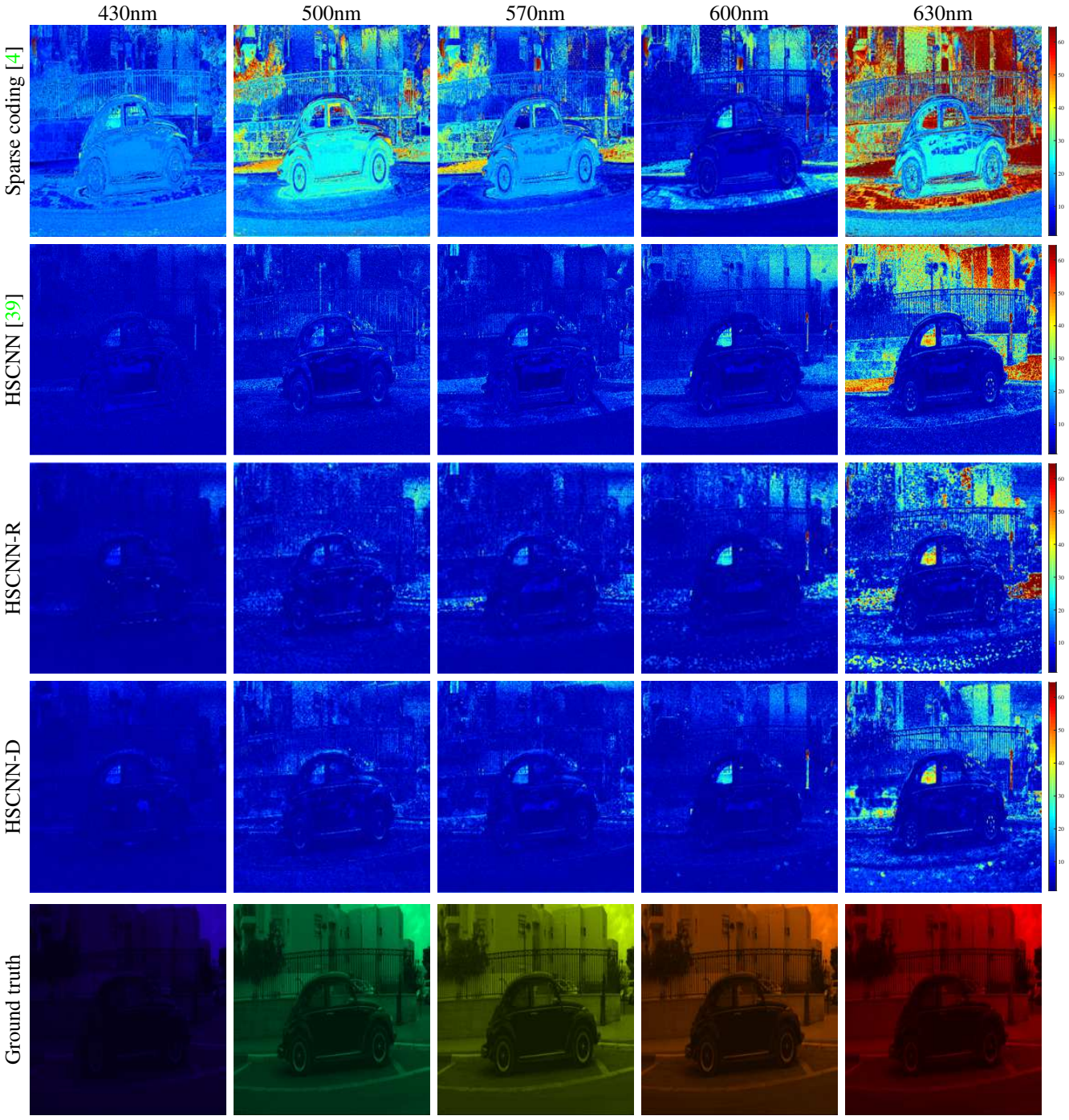

Figure 3. Visual comparison of five selected bands for hyperspectral recovery from "Clean" RGB images.

ual block by the dense block with a novel fusion scheme. It substantially deepens the network structure to achieve a more accurate solution.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China under grants 61671419 and 61425026.

References

[1] J. Aeschbacher, J. Wu, R. Timofte, D. CVL, and E. ITET. In defense of shallow learned spectral reconstruction from rgb images. In *ICCVW*, 2017. 1, 2

[2] N. Akhtar and A. Mian. Hyperspectral recovery from rgb images using gaussian processes. In *arXiv:1801.04654*, 2018. 1, 2

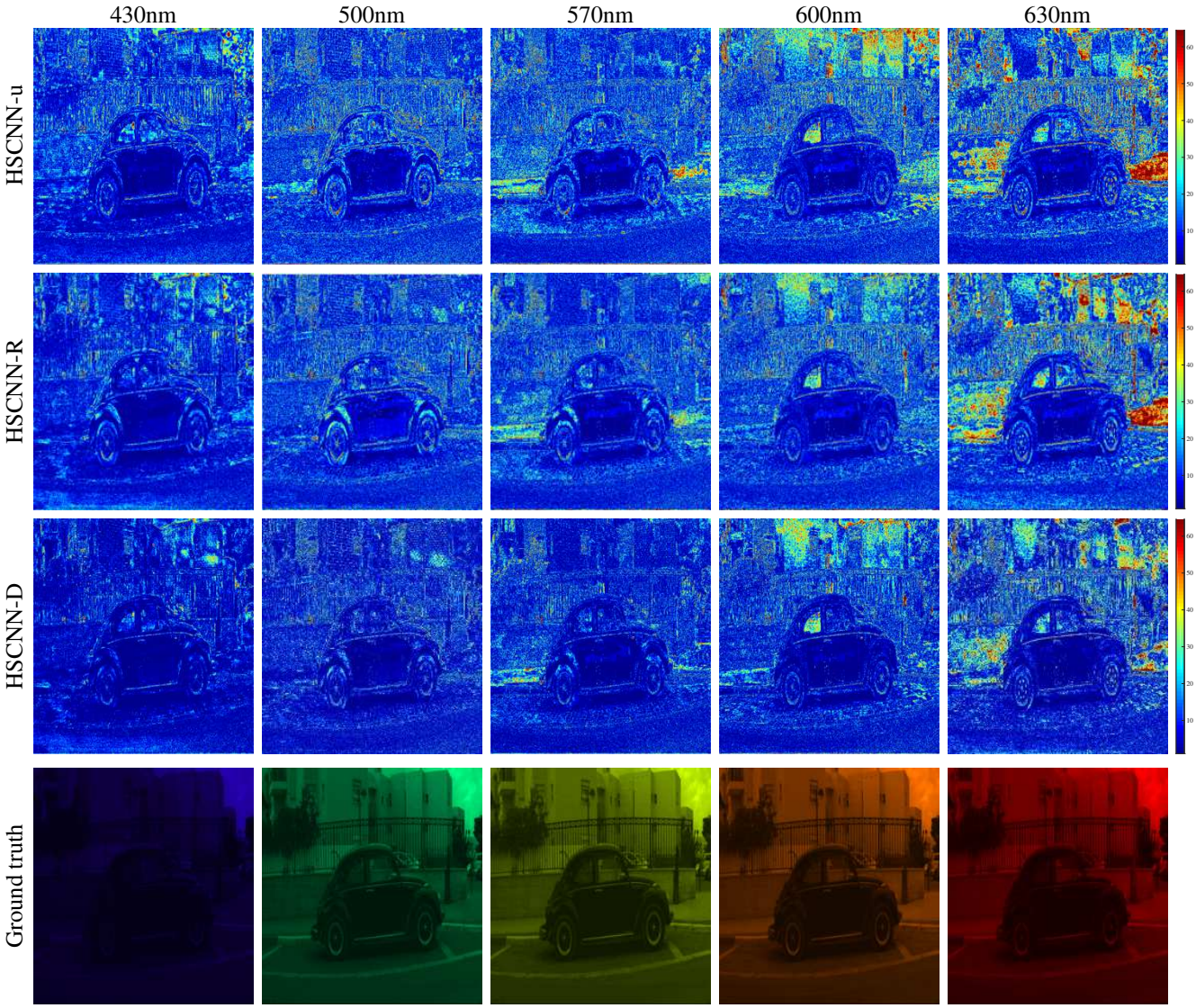

Figure 4. Visual comparison of five selected bands for hyperspectral recovery from "Real World" RGB images.

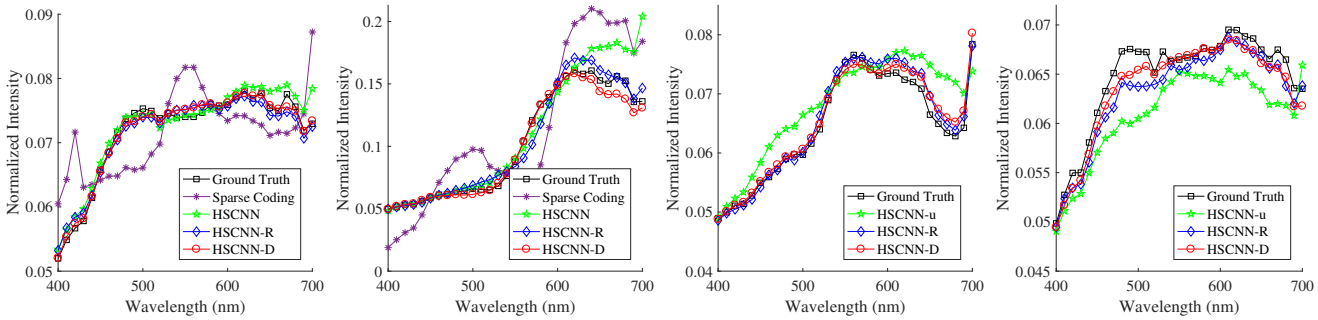

Figure 5. Spectral signatures of selected spatial points from reconstructed hyperspectral images. In (a) and (b), sparse coding and HSCNN are adopted for comparison in the "Clean" track. In (c) and (d), HSCNN-u are adopted for comparison in the "Real World" track.

[3] A. Alvarez-Gila, J. van de Weijer, and E. Garrote. Adversarial networks for spatial context-aware spectral image reconstruction from rgb. In *ICCVW*, 2017. 1, 2

[4] B. Arad and O. Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *ECCV*, 2016. 1, 2, 4, 5, 6, 7

[5] B. Arad, O. Ben-Shahar, R. Timofte, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on spectral reconstruction from rgb images. In *CVPRW*, 2018. 2, 4

[6] R. W. Basedow, D. C. Carmer, and M. E. Anderson. Hydice system: Implementation and performance. In *Proc. SPIE*, 1995. 1

[7] D. J. Brady. *Optical imaging and spectroscopy*. John Wiley and Sons, 2009. 2

[8] X. Cao, H. Du, X. Tong, Q. Dai, and S. Lin. A prism-mask system for multispectral video acquisition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2423–2435, 2011. 2

[9] A. Chakrabarti and T. Zickler. Statistics of real-world hyperspectral images. In *CVPR*, 2011. 1

[10] C. Chen, X. Tian, Z. Xiong, and F. Wu. Udnet: Up-down network for compact and efficient feature representation in image super-resolution. In *ICCVW*, 2017. 3

[11] M. Descour and E. Dereniak. Computed-tomography imaging spectrometer: experimental calibration and reconstruction results. *Appl. Opt.*, 34(22):4817–4826, 1995. 2

[12] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, 2016. 3

[13] S. Galliani, C. Lanaras, D. Marmanis, E. Baltsavias, and K. Schindler. Learned spectral super-resolution. In *arXiv:1703.09470*, 2017. 1, 2

[14] N. Gat. Imaging spectroscopy using tunable filters: a review. In *Proc. SPIE*, 2000. 1

[15] M. Goel, E. Whitmire, A. Mariakakis, T. S. Saponas, N. Joshi, D. Morris, B. Guenter, M. Gavriliu, G. Borriello, and S. N. Patel. Hypercam: hyperspectral imaging for ubiquitous computing applications. In *UbiComp*, 2015. 1, 2

[16] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock. Imaging spectrometry for earth remote sensing. *Science*, 228(4704):1147–1153, 1985. 1

[17] A. Gowen, C. O'Donnell, P. Cullen, G. Downey, and J. Frias. Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. *Trends in Food Science & Technology*, 18(12):590–598, 2007. 1

[18] D. Haboudane, J. R. Miller, E. Pattey, P. J. Zarco-Tejada, and I. B. Strachan. Hyperspectral vegetation indices and novel algorithms for predicting green lai of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90(3):337 – 352, 2004. 1

[19] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 4

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[21] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[22] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2, 3

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2014. 4

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 2

[26] X. Lin, Y. Liu, J. Wu, and Q. Dai. Spatial-spectral encoded compressive hyperspectral imaging. *ACM Trans. Graph*, 33(6):233, 2014. 1

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 3

[28] Z. Pan, G. Healey, M. Prasad, and B. Tromberg. Face recognition in hyperspectral images. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 25(12):1552–1560, 2003. 1

[29] B. Smits. An rgb-to-spectrum conversion for reflectances. *J. Graphics Tools*, 4(4):11–22, 1999. 3

[30] T. Takatani, T. Aoto, and Y. Mukaigawa. One-shot hyperspectral imaging using faced reflectors. In *CVPR*, 2017. 1, 2

[31] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7):2367–2379, 2010. 1

[32] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 2

[33] H. Van Nguyen, A. Banerjee, and R. Chellappa. Tracking via object reflectance using a hyperspectral video camera. In *CVPRW*, 2010. 1

[34] A. Wagadarikar, R. John, R. Willett, and D. Brady. Single disperser design for coded aperture snapshot spectral imaging. *Appl. Opt.*, 47(10):B44–B51, 2008. 1, 2

[35] L. Wang, Z. Xiong, D. Gao, G. Shi, and F. Wu. Dual-camera design for coded aperture snapshot spectral imaging. *Appl. Opt.*, 54(4):848–858, 2015. 1

[36] L. Wang, Z. Xiong, D. Gao, G. Shi, W. Zeng, and F. Wu. High-speed hyperspectral video acquisition with a dual-camera architecture. In *CVPR*, 2015. 1

[37] L. Wang, Z. Xiong, G. Shi, F. Wu, and W. Zeng. Adaptive nonlocal sparse representation for dual-camera compressive hyperspectral imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(10):2104–2011, 2017. 2

[38] S. Wug Oh, M. S. Brown, M. Pollefeys, and S. Joo Kim. Do it yourself hyperspectral imaging with everyday digital cameras. In *CVPR*, 2016. 1, 2

[39] Z. Xiong, Z. Shi, H. Li, L. Wang, D. Liu, and F. Wu. Hscnn: Cnn-based hyperspectral image recovery from spectrally undersampled projections. In *ICCVW*, 2017. 1, 2, 4, 5, 6, 7

[40] L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng. On the connection of deep fusion to ensembling. In *arXiv:1611.07718*, 2016. 4