Deep Residual Convolutional Neural Network for Hyperspectral Image Super-resolution

Chen Wang[1], Yun Liu[2], Xiao Bai[1], Wenzhong Tang[1], Peng Lei[3], and Jun Zhou[4]

[1] School of Computer Science and Engineering, Beihang University, Beijing, China
[2] School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
[3] School of Electronic and Information Engineering, Beihang University, Beijing, China
[4] School of Information and Communication Technology, Griffith University, Nathan, Australia

Abstract. Hyperspectral image is very useful for many computer vision tasks, however it is often difficult to obtain high-resolution hyperspectral images using existing hyperspectral imaging techniques. In this paper, we propose a deep residual convolutional neural network to increase the spatial resolution of hyperspectral image. Our network consists of 18 convolution layers and requires only one low-resolution hyperspectral image as input. The super-resolution is achieved by minimizing the difference between the estimated image and the ground truth high resolution image. Besides the mean square error between these two images, we introduce a loss function which calculates the angle between the estimated spectrum vector and the ground truth one to maintain the correctness of spectral reconstruction. In experiments on two public datasets we show that the proposed network delivers improved hyperspectral super-resolution result than several state-of-the-art methods.

Keywords: hyperspectral image super-resolution, deep residual convolutional neural network

1 Introduction

Hyperspectral imaging acquires spectral representation of a scene through capturing a large number of continuous and narrow spectral bands. The spectral characteristics of the hyperspectral image have been proven useful for many visual tasks, including tracking [15], segmentation [19], face recognition [16] and document analysis [10]. However, in each narrow band only a small fraction of the overall radiant energy reaches the sensor. To maintain a good signal-to-noise ratio, the imaging system increases the pixel size on the chip and uses long exposures, which however, results in low spatial resolution of hyperspectral images.

Recently, several matrix factorization based approaches [9, 13, 14, 1, 7] have been proposed for hyperspectral image super-resolution. All these methods need

auxiliary data of RGB image of the same scene, and these methods cannot be directly applied to hyperspectral images which are normally obtained beyond the visible range.

Convolutional neural networks have recently been intensively explored due to their powerful learning capability. Motivated by this property, we propose a deep convolutional neural network method which learns an end-to-end mapping between low- and high-resolution images. This network does not need any high resolution RGB image to provide additional information. Furthermore, Hyperspectral image is a data-cube and has obvious physical meaning in spectral dimension. Traditional deep networks were developed for super-resolution of grayscale images, therefore, can not be directly applied to hyperspectral image. To address this problem, we introduce a loss function which calculates the angle between the estimated spectrum vector and the ground truth one to maintain the correctness of spectral reconstruction. When the network comes deeper, the vanishing gradients problem are significantly critical, so we use residual-learning and additional supervised output to solve this problem.

2 Related Work

In this section, we review relevant hyperspectral image super-resolution methods and deep learning methods for grayscale image super-resolution.

2.1 Hyperspectral Image Super-resolution

In early years, Pan-sharpening techniques [2, 17] were introduced to merge a high resolution panchromatic (single band) image and a low resolution hyperspectral image to reconstruct a high resolution hyperspectral image. In addition, filtering techniques [8, 12] were proposed which used high resolution edges from other images of the same scene to guide filtering process. These methods indeed improve the spatial resolution of hyperspectral images, but the reconstructed high resolution images sometimes contain spectral distortions.

More recently, matrix factorization based techniques for hyperspectral image super-resolution have been proposed. Kawakami et al. [9] used matrix factorization to firstly learn a series of spectral bases. Then the sparse coefficients of these spectral bases were calculated, which best reconstructed the corresponding high resolution RGB signals. At last, they used these coefficients and the spectral bases to reconstruct the high resolution hyperspectral image. This work was extended by Akhtar et al. [1] who imposed a non-negativity constraint over the solution space. Kwon et al. [13] upsampled the image guided by high resolution RGB images of the same scene and then utilized sparse coding to locally refine the upsampled hyperspectral image through dictionary substitution. Lanaras et al. [14] proposed a method to perform hyperspectral super-resolution by jointly unmixing the RGB and hyperspectral images into the pure reflectance spectra of the observed materials and the associated mixing coefficients. To improve the accuracy of non-negative sparse coding, a clustering-based structured sparse

coding method [7] was introduced to exploit the spatial correlation among the learned sparse codes. All these matrix factorization based methods need high resolution RGB images to provide extra information.

2.2 Deep Learning Methods on Grayscale Images

During the past three years, deep learning methods [4, 20, 5, 6, 18, 11] have been used for single-band grayscale image super-resolution and demonstrated great success. To the best of our knowledge, however, deep learning has not been introduced for hyperspectral image super-resolution. A hyperspectral image is not simply a concatenation of several single-band images because of its obvious physical meaning in the spectral dimension. So it is necessary to develop suitable networks for hyperspectral image.

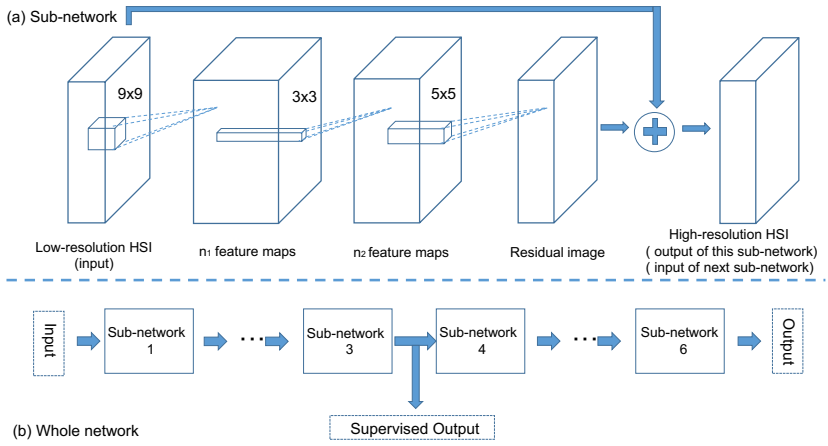

Fig. 1. The structure of the proposed deep residual convolutional neural network.

3 Deep Residual Convolutional Neural Network

In this section, we describe the proposed deep residual convolutional neural network and the new loss function to cope with spectral dimension of the image data.

3.1 Deep Residual Convolutional Neural Network

We propose a deep residual convolutional neural network to increase the resolution of hyperspectral image. The structure of this network is outlined in Fig. 1. We cascade 6 sub-networks which have the same structure. Each sub-network

is a residual convolutional neural network. The input of the sub-network is low-resolution hyperspectral image and the output is high-resolution hyperspectral image. Meanwhile, the output of each sub-network is then regarded as the input of the next sub-network. Each sub-network has three types of filter: $c \times f_1 \times f_1 \times n_1$ for the first layer, $n_1 \times f_2 \times f_2 \times n_2$ for the second layer, and $n_2 \times f_3 \times f_3 \times c$ for the last layer. In this paper, we set $f_1 = 9, f_2 = 3, f_3 = 5, n_1 = 96, n_2 = 64$, where f_* means the size of the convolutional kernels and n_* means the number of feature maps, c means the number of original image channels. The first layer takes the input image and represents it as a set of feature maps (n_1 channels). The second layer is used to dig deeper features. At last, the third layer is used to transform the features (n_2 channels) back into the original image space (c channels).

When the network comes deeper, the vanishing gradients problem can be critical. Inspired by [11], we use residual-learning to solve this problem. As shown in Fig. 1, we get the residual image after the third layer. The final output of the sub-network is the sum of residual image and the input of this sub-network. It is worth mentioning that our network is a progressive structure which gradually learns the residual components. Back propagation goes through a small number of layers if there is only one supervising signal at the end of the network. So we add supervised output at the end of the third sub-networks to make sure the weights of the first three sub-networks can be updated efficiently. The optimal weights are learned by automatically minimizing the loss function of both supervised output and the final output. We will define the loss function in the next subsection.

We have not used any pooling layers or deconvolution layers. The whole network takes interpolated low-resolution image (to the size of high resolution image) as input and predicts the missed high frequency parts. Although there is no pooling layer, the size of the feature map gets reduced every time the convolution operations are applied. So we use zero-padding before each convolution operation to make sure that all feature maps have the same size.

3.2 Loss Function

We now describe the loss function of our network. Given a training dataset $\{x^{(i)}, y^{(i)}\}_i^N$, where $x^{(i)}$ or $y^{(i)}$ means it is the i_{th} image in the dataset and N is the total number of images, our goal is to learn a model f that predicts values $\hat{y} = f(x)$, where x is a low-resolution hyperspectral image (after interpolation), y is the target high-resolution hyperspectral image,

Mean squared error $\frac{1}{2}\|y - \hat{y}\|^2$ is widely used in least-squares regression setting. This favors high Peak Signal-to-Noise Ratio (PSNR) by minimizing the first loss function which is defined as

$$l_1 = \frac{1}{2N} \sum_i^N \left\| y^{(i)} - \hat{y}^{(i)} \right\|^2 \tag{1}$$

Considered that reflectance spectrum is the most important information in a hyperspectral image, we add a new loss function which calculates the angle between the estimated spectrum vector and the ground truth one. Let N_m be the number of spectrum vector of hyperspectral image, y_j be the jth spectral vector. We have the second loss function

$$l_2 = \frac{1}{NN_m} \sum_i^N \sum_j^{N_m} \frac{\langle y_j^{(i)}, \hat{y_j}^{(i)} \rangle}{\left\| y_j^{(i)} \right\| \times \left\| \hat{y}_j^{(i)} \right\|} \tag{2}$$

The final loss function is the linear combination of these two loss functions

$$L = \alpha l_1 + (1 - \alpha) l_2 \tag{3}$$

where $\alpha = 0.5$ in our work.

We implement our model using the TensorFlow framework. Training is carried out by minimizing the loss function using mini-batch gradient descent based on back-propagation. After training, low-resolution hyperspectral image is used as the input to test out network, and we record the super-resolution results for comparison purpose.

4 Experiment

We used two publicly available hyperspectral datasets: CAVE [21] and Harvard [3] in the experiments. The first dataset includes 32 indoor images. The spatial resolution of the images is 512×512. Each image has 31 spectral bands with 10nm spectral resolution, ranging from 400nm to 700nm. The second dataset has 50 indoor and outdoor images recorded under daylight illumination and 27 images under artificial or mixed illumination. The spatial dimension of the images is 1392×1040 pixels, with 31 spectral bands covering the visible spectrum from 420nm to 720nm at 10nm spectral resolution. For convenience, we used only the top left 1024×1024 pixels of each image to make the spatial dimension of the ground truth a multiple of 32. Fig. 2 shows some representative images from these CAVE datasets. Fig. 3 shows some representatives image from Harvard datasets.

4.1 Implementation Detail

In our experiments, the original images served as the ground truth. To obtain low-resolution hyperspectral images, we blurred the original images using a Gaussian kernel (standard deviation=3), downsampled it by a scale factor (=2,3,4) and then upscaled it to the desired size using bicubic interpolation with a scale factor (=2,3,4). These images have the same size with ground truth but lose the high frequency components, so we still call them low-resolution images.

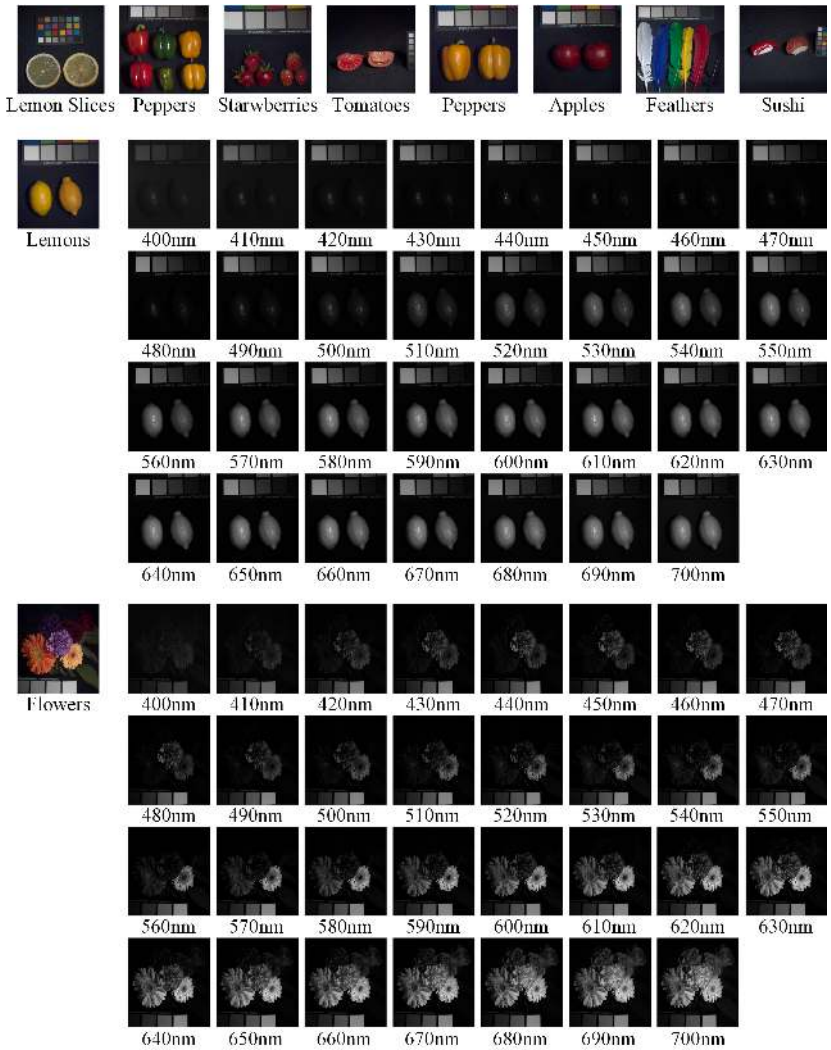

Fig. 2. Selected examples of hyperspectral image from CAVE database.

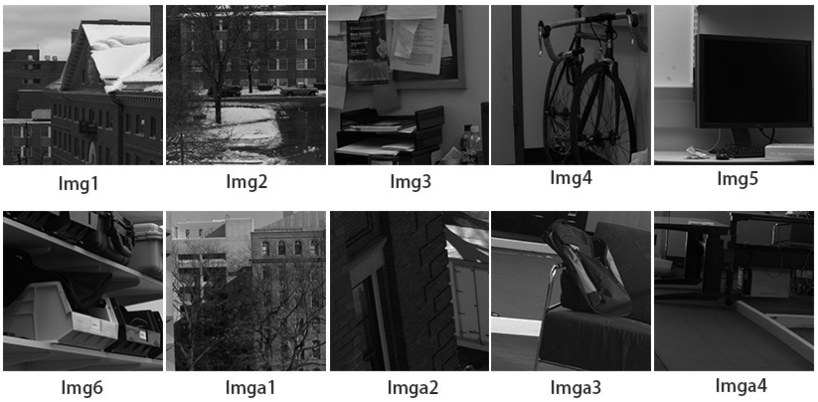

Fig. 3. Selected examples of hyperspectral image from Harvard database.

The testing set included 7 images from the CAVE dataset (Balloons, Chart and stuffed toy, Faces, Flowers, Jelly beans, Real and fake apples, and Oil painting) and 10 images from the Harvard dataset (Img1, Img2, Img3, Img4, Img5, Img6, Imga1, Imga2, Imga3 and Imga4). The rest 92 images were used for training. For testing, we used the whole interpolated low-resolution images as the input and compared the output with the corresponding ground truth. For training, we used $32 \times 32 \times 31$ cubic-patches randomly cropped from the training images. In total, 30,000 cubic-patches were generated from the training set.

We cascaded 6 sub-networks and added supervised output to the end of the third one. In our experiments, adding more layers does not bring obvious improvement to the result. So we finally used 18 convolution layers. After each convolution layer, we used ReLU as the nonlinear mapping function. For training, we used mini-batch gradient descent based on back propagation and the batch size was set to 128. We set the momentum parameter to 0.9 and the weight decay to 0.0001. Learning rate was initially set to 0.01 and then decreased by a factor of 10 if the validation error did not decrease for 3 epochs. If the learning rate was less than 10^{-6}, the procedure was terminated. We trained three independent networks for factor$= 2, 3, 4$. It took 2 days to train one network on a GTX 980 Ti, but it took only 4.2 second to process a testing hyperspectral image on average.

4.2 Experimental Results

All hyperspectral image super-resolution methods reviewed in section 2 used extra high resolution image to help the estimation process, so it is unfair to directly compare our method with these methods. We used another comparing strategy. Firstly, we set bicubic interpolation method as the baseline to evaluate the learning ability of our network. Then we compared our method with three single-band image super-resolution neural networks [4, 18, 11] to show that our network is more suitable for hyperspectral images. We considered hyperspectral

images as a series of independent gray-level images and used these images to train and test the neural networks. To be fair, all methods used the same training set and we followed the network settings in their original paper for each method being compared.

In this work, we used Peak Signal-to-Noise Ratio (PSNR) and spectral angle mapper(SAM) [22] as the evaluation measurements. Peak Signal-to-Noise Ratio (PSNR) is used as the primary evaluation measure for the estimated high-resolution hyperspectral image $\hat{Z}$ and the ground truth image Z in an n-bit intensity range. Let B be the number of bands of hyperspectral image and N_m be the number of spectral vector of hyperspectral image, PSNR is calculated by

$$PSNR = 10 \times \log_{10} \left(\frac{2^n - 1}{\frac{1}{BN_m} \left\| \hat{Z} - Z \right\|_F^2} \right). \tag{4}$$

To evaluate the correctness of spectral responses, we used the spectral angle mapper (SAM) [22], which is defined as the angle between the estimated spectrum vector $\hat{z}_i$ and the ground truth spectrum vector z_j, averaged over the whole image. The SAM is given in degrees

$$SAM = \frac{1}{N_m} \sum \arccos \frac{\hat{z}_j^T z_j}{\|\hat{z}_j\|_2 \|z_j\|_2}. \tag{5}$$

Table 1 and Table 2 show the average PSNR and SAM values on the two datasets. Our approach achieves higher PSNR than all four methods. Notably, in terms of SAM, our method is clearly the best among all the methods. Considering hyperspectral image as a series of gray-level images instead of a whole data-cube always leads to spectral distortion. The loss function proposed in this paper solves this problem well. To complement the tabulated results, we also visualize the experimental results in Fig. 4 and Fig. 5. Due to space limitation, we only show one spectral image at 550nm. The fourth column of Fig. 4 and Fig. 5 shows the absolute difference between the estimated high resolution hyperspectral image and the ground truth. It shows that with our method, a significantly larger number of pixels have very small reconstruction errors below 1 in grayscale value.

5 Conclusion

We have introduced a deep residual convolutional neural network for hyperspectral image super-resolution. The input to this network is a single low-resolution hyperspectral image and no extra RGB or other high-resolution images are needed. A new loss function is used to make the framework more suitable for hyperspectral image. Residual-learning and additional supervised output are used to solve the vanishing gradients problem. Experimental results show that the proposed method performs well under both PSNR and SAM measurements.

Table 1. Results on the CAVE dataset

Method	CAVE Database					
	PSNR			SAM		
Scale	x2	x3	x4	x2	x3	x4
Bicubic	31.73	31.58	30.14	6.00	5.99	6.36
Dong[4]	36.28	35.10	34.67	4.23	4.20	4.52
Shi[18]	36.65	35.39	34.91	4.12	4.16	4.43
Kim[11]	36.98	35.87	35.02	4.35	4.21	4.39
Ours	38.24	37.86	37.14	1.56	1.73	1.85

Table 2. Results on the Harvard dataset

Method	Harvard Database					
	PSNR			SAM		
Scale	x2	x3	x4	x2	x3	x4
Bicubic	36.67	36.52	36.39	3.09	3.10	3.17
Dong[4]	38.98	38.41	38.10	2.57	2.61	2.74
Shi[18]	39.35	38.63	38.29	2.55	2.57	2.61
Kim[11]	39.54	39.02	38.59	2.46	2.49	2.53
Ours	40.13	39.68	39.14	1.14	1.21	1.35

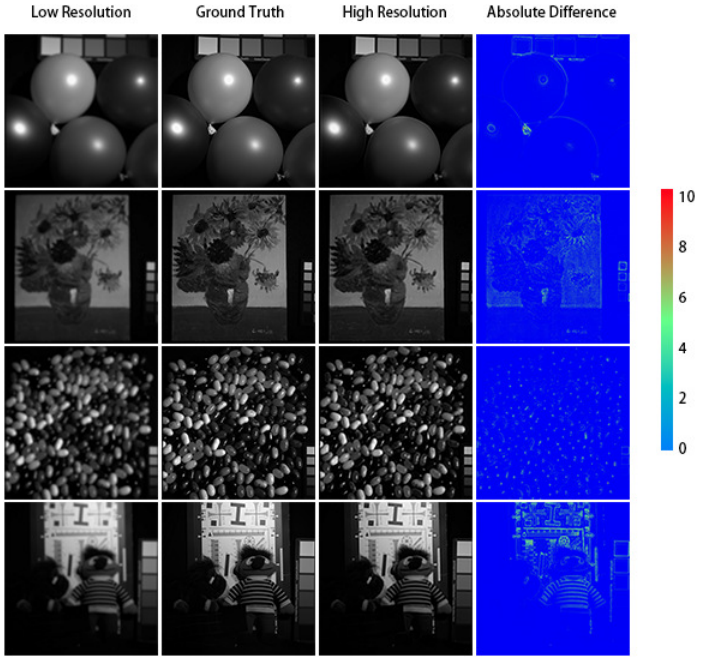

Fig. 4. Spectral images at 550nm for sample CAVE [21] data. Estimated high resolution images are shown along with their absolute difference with the ground truth. The scale factor is 3.

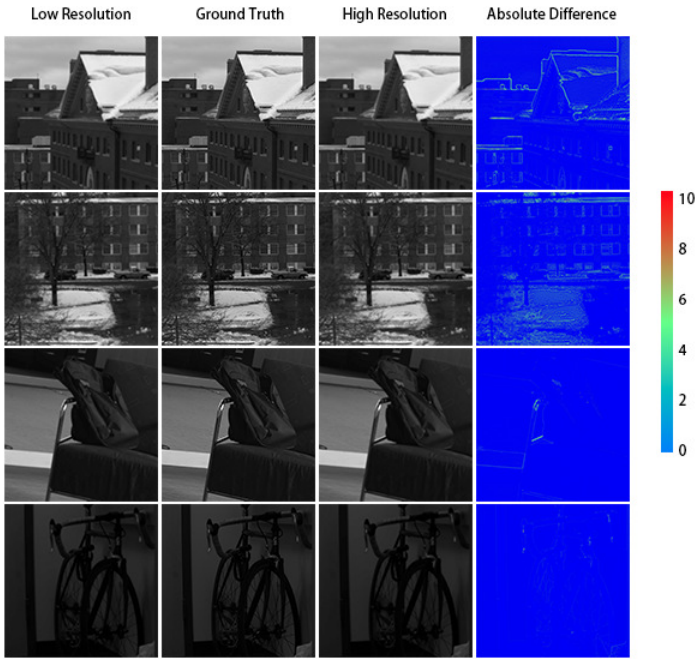

Fig. 5. Spectral images at 550nm for sample Harvard [3] data. Estimated high resolution images are shown along with their absolute difference with the ground truth. The scale factor is 3.

Acknowledgements

This work is supported by NSFC project No.61370123 and BNSF project No.4162037. It is also supported by funding from State Key Lab of Software Development Environment in Beihang University.

References

1. Akhtar, N., Shafait, F., Mian, A.: Sparse spatio-spectral representation for hyperspectral image super-resolution. In: European Conference on Computer Vision. pp. 63–78 (2014)
2. Brower, C.A.L.B.V.: Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening. Websterny Uspenfieldny Us (2000)
3. Chakrabarti, A., Zickler, T.: Statistics of real-world hyperspectral images. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 193–200 (2011)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution. Springer International Publishing (2014)

5. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(2), 295–307 (2016)
6. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. European Conference on Computer Vision(ECCV) (2016)
7. Dong, W., Fu, F., Shi, G., Cao, X., Wu, J., Li, G., Li, X.: Hyperspectral image super-resolution via non-negative structured sparse representation. IEEE Transactions on Image Processing 25(5), 2337–2352 (2016)
8. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(6), 1397–1409 (2013)
9. Kawakami, R., Matsushita, Y., Wright, J., Ben-Ezra, M., Tai, Y.W., Ikeuchi, K.: High-resolution hyperspectral imaging via matrix factorization. In: CVPR 2011. pp. 2329–2336 (2011)
10. Khan, Z., Shafait, F., Mian, A.: Hyperspectral imaging for ink mismatch detection. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 877–881 (2013)
11. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1637–1645 (2016)
12. Kopfand, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics 26(3), 96 (2007)
13. Kwon, H., Tai, Y.W.: RGB-guided hyperspectral image upsampling. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 307–315 (2015)
14. Lanaras, C., Baltsavias, E., Schindler, K.: Hyperspectral super-resolution by coupled spectral unmixing. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3586–3594 (2015)
15. Nguyen, H.V., Banerjee, A., Chellappa, R.: Tracking via object reflectance using a hyperspectral video camera. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. pp. 44–51 (2010)
16. Pan, Z.H., Healey, G., Prasad, M., Tromberg, B.: Face recognition in hyperspectral images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1552–1560 (2003)
17. Shah, V.P., Younan, N.H., King, R.L.: An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets. IEEE Transactions on Geoscience and Remote Sensing 46(5), 1323–1335 (2008)
18. Shi, W., Caballero, J., Huszr, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1874–1883 (2016)
19. Tarabalka, Y., Chanussot, J., Benediktsson, J.A.: Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 40(5), 1267–1279 (2010)
20. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 370–378 (2015)
21. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S.K.: Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum. IEEE Transactions on Image Processing 19(9), 2241–2253 (2010)

22. Yuhas, R.H., Boardman, J.W., Goetz, A.F.H.: Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques (1993)