Deeply Learned Filter Response Functions for Hyperspectral Reconstruction

Shijie Nie[1], Lin Gu[1], Yinqiang Zheng[1], Antony Lam[2], Nobutaka Ono[3], Imari Sato[1]
[1]National Institute of Informatics, [2]Saitama University, [3]Tokyo Metropolitan University
nsj@nii.ac.jp, ling@nii.ac.jp, yqzheng@nii.ac.jp
antonylam@cv.ics.saitama-u.ac.jp, onono@tmu.ac.jp, imarik@nii.ac.jp

Abstract

Hyperspectral reconstruction from RGB imaging has recently achieved significant progress via sparse coding and deep learning. However, a largely ignored fact is that existing RGB cameras are tuned to mimic human trichromatic perception, thus their spectral responses are not necessarily optimal for hyperspectral reconstruction. In this paper, rather than use RGB spectral responses, we simultaneously learn optimized camera spectral response functions (to be implemented in hardware) and a mapping for spectral reconstruction by using an end-to-end network. Our core idea is that since camera spectral filters act in effect like the convolution layer, their response functions could be optimized by training standard neural networks. We propose two types of designed filters: a three-chip setup without spatial mosaicing and a single-chip setup with a Bayer-style 2x2 filter array. Numerical simulations verify the advantages of deeply learned spectral responses compared to existing RGB cameras. More interestingly, by considering physical restrictions in the design process, we are able to realize the deeply learned spectral response functions by using modern film filter production technologies, and thus construct data-inspired multispectral cameras for snapshot hyperspectral imaging.

1. Introduction

Hyperspectral imaging captures detailed light distribution along the wavelength axis. It is shown to be beneficial for remote sensing, medical diagnosis, industrial detection, and so on [12, 24]. For example, the tumor margin, invisible to surgeon's eyes, could be better visualized in hyperspectral images. Cases of leaked invisible gas may also be obvious using spectral signals.

Most existing devices to capture hyperspectral images are scanning based, that is, either to drive a line slit along one spatial dimension (pushbroom scan) or to continuously change narrow bandpass filters in front of a grayscale camera (filter scan). The key drawback is that scanning is slow, which prevents their application to dynamic scenes. Thus scanning-free, snapshot hyperspectral devices have been developed, by using for example, fiber bundles [25] and random/regular aperture masks [32, 13, 7]. Unfortunately, these devices are extremely limited in spatial resolution.

A computational hyperspectral reconstruction method from a single RGB image is promising in overcoming the drawbacks of the aforementioned devices, as evidenced in recent research on RGB-to-Spectrum reconstruction [26, 28, 4, 17, 30, 3, 33]. However, existing RGB cameras, either using the three-chip beam splitting prism technique or single-chip Bayer filter array, are designed to mimic human color perception [19], thus their spectral response functions are not necessarily optimal for computer vision tasks, *i.e.* hyperspectral reconstruction. Very recently, Arad and Ben-Shahar [5] identified the dependence of hyperspectral reconstruction accuracy on the camera's spectral response. In [5], they find the best filter combination among a finite set of candidate filters via brute force search and hit-and-run evolutionary optimization.

The latest film filter production technologies have allowed us to implement image sensors with any non-negative and smooth spectral response functions. Therefore, rather than selecting filters from existing ones, in this paper, we aim to directly learn optimized spectral response functions in the infinite space of non-negative and smooth functions. We then manufacture our learned filter, based on this data-driven approach to construct a multispectral camera for snapshot hyperspectral imaging (Sec. 6).

Based on our observation that camera spectral filters act in effect, like the convolution layer in neural networks (details in Sec. 4.2), we are able to optimize them by using deep learning techniques. We simultaneously learn the optimized filter response functions and the mapping for spectral reconstruction via a high resolution end-to-end network. Inspired by existing RGB cameras, we consider a three-chip setup without spatial mosaicing and a single-chip setup with a Bayer-style 2x2 filter array. Numerical simulations on publicly available datasets verify the advantages of deeply learned camera spectral responses over existing RGB cam-

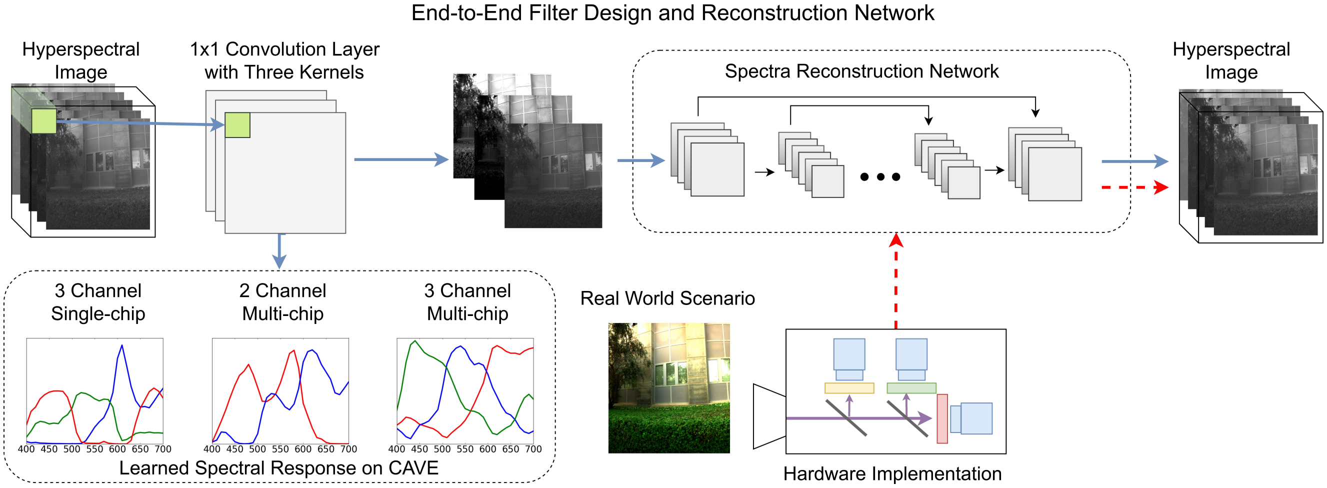


Figure 1. Our proposed design-realization-application framework for data-inspired spectral imaging hardware. The design stage (marked in blue arrow) is data-driven. It includes an end-to-end network to simultaneously learn the filter response and the spectral reconstruction mapping. The learned spectral response function on CAVE dataset is also shown. In the realization stage (marked in red arrow), the learned response functions are realized by using film filter production technologies, and a data-inspired multispectral camera is constructed. In the online application stage, the captured multispectral image is imported into the already trained spectral reconstruction network to generate hyperspectral images. This framework is illustrated using the multi-chip setup with three channels.

eras.

With the deeply learned filters, we propose our data-inspired multispectral camera for snapshot hyperspectral imaging.

Our contributions can be summarized as follows:

1. The connection between camera spectral response function and the convolution layer of neural networks. We find that the camera spectral response can be regarded as a hardware implementation of the convolution layer.
2. By simulating the camera response as a convolution layer and appending onto the spectral reconstruction network, we can simultaneously learn the optimized response functions and hyperspectral reconstruction mapping.
3. We propose two setups for optimized filter design: a three-chip setup without mosaicing and a single-chip setup with a Bayer-style 2x2 filter array. We demonstrate that the deeply learned response functions are better than standard RGB responses in a specific computer vision task, *spectral reconstruction*.
4. We realize the deeply learned filters by using interference film production technologies, and construct a snapshot hyperspectral imaging system.

The remaining parts of this paper are organized as follows. In Sec. 2, we briefly discuss the related work. Sec. 3 highlights a novel framework for data-inspired hardware construction, and Sec. 4 includes our end-to-end network for simultaneous filter design and hyperspectral reconstruction. Numerical simulation results are shown in Sec. 5, and the practical construction of the filters and imaging system are given in Sec. 6. Finally, we conclude this research in Sec. 7.

2. Related Work

To resolve the speed bottleneck of scanning based hyperspectral cameras, scanning-free devices have been proposed by using, for example, fiber bundles [25] and aperture masks with randomly [32, 13] or regularly [7] distributed light windows. The major drawback of such snapshot devices lies in their limited spatial resolution. There are also a number of fusion based super-resolution algorithms to boost the spatial resolution by using a high-resolution grayscale [18, 35] or RGB [8, 20, 1, 22, 23, 2] image.

Rather than making a hyperspectral imager directly, approaches for increasing the spectral resolution of a single RGB image have recently attracted much attention. The key in hyperspectral reconstruction is to find a mapping between the RGB value and the high-dimensional spectral signal, which is obviously an ill-posed problem, and requires proper priors for reconstruction. In [26], Nguyen et al. tried to eliminate the illumination effect via a white balancing algorithm, and learn the mapping from illumination-free RGB values to reflectance spectra on the basis of a radial basis function (RBF) network. Robles-Kelly [28] aimed at the same problem and proposed to learn a representative dictionary using a constrained sparse coding method. Arad and Ben-Shahar [4] focused on hyperspectral images of natural scenes and developed an RGB-to-spectrum mapping method using sparse coding. Very recently, Jia *et al.*

[17] examined the intrinsic dimensionality of natural hyperspectral scenes and proposed a three-dimensional manifold based mapping method for spectral reconstruction.

In contrast to sparse coding and shallow neural networks, deep learning has recently been applied to RGB based hyperspectral reconstruction. Galliani *et al*. [30] first introduced a convolutional neural network for spectral reconstruction from a single RGB image. They adapted the Tiramisu network and reported favorable results over the dictionary based method [4] and the shallow network [26]. Alvarez-Gila *et al*. [3] applied a conditional generative adversarial framework to better capture spatial semantics. Xiong *et al*. [33] proposed a unified convolutional neural network for hyperspectral reconstruction from RGB images and compressively sensed measurements. Compared with pixelwise operations [26, 28, 4], the imagewise operations in deep learning based methods [30, 3, 33] are more likely to incorporate spatial consistency in the reconstruction.

All the research above simulated RGB images using typical response functions from commercial RGB cameras. Very recently, Arad and Ben-Shahar [5] recognized the accuracy of hyperspectral reconstruction is dependent on the filter response, and tried to find the best filter combination among a finite set of candidate filters via brute force search and hit-and-run evolutionary optimization. In this paper, we further expand the search domain to the infinite space of nonnegative and smooth curves. Leveraging powerful deep learning techniques, we simultaneously learn an optimized filter response and the spectral reconstruction mapping. Interestingly, our hardware implementation of optimized filter responses has parallels with ASP vision [10], which uses custom CMOS diffractive image sensors to directly compute a fixed first layer of the CNN to save energy, data bandwidth, and CNN FLOPS. However, in the case of ASP vision, their aim is to hard code a pre-defined edge filtering layer that is common to CNNs and the v1 layer of the human visual cortex. Then [10] uses it in solving various tasks such as recognition efficiently. Our aim is to leverage the CNN and deep learning framework to optimize camera filter design. To our knowledge, we are the first to achieve this and demonstrate accurate hyperspectral reconstruction from our designed filters.

3. Design-Realization-Application Framework

In this paper, we advocate a novel design-realization-application framework for data-inspired and task-oriented spectral imaging hardware development, which is illustrated in Fig. 1. As for the data-driven design stage, we construct an end-to-end network by appending a tailored convolution layer onto the spectral reconstruction network. Since we properly incorporate the nonnegativity and smoothness constraints, the convolution layer acts in effect as the filter spectral response functions that we aim to design. It encodes an input hyperspectral image into the most appropriate hidden feature map (multispectral image), such that the succeeding reconstruction network can recover the original input hyperspectral image as faithfully as possible. In this sense, our end-to-end network is similar to the autoencoder-decoder.

In the realization stage, we try to physically realize the deeply learned response functions by using film filter production technologies. In the multi-chip setup, we can easily construct a multispectral camera such that the output of this camera is sufficiently close to the learned hidden feature map. We admit that it is much more involved to realize the learned filter array in the single-chip setup. Yet, we believe this can be achieved in the near future, thanks to the latest progress in micro filter array production technologies.

In the online application stage, we use the customized multispectral camera to capture images, and directly import them into the already trained reconstruction network to generate hyperspectral images. Therefore, our reconstruction will in principle share all the benefits arising from deep learning, as mentioned in [30, 3, 33].

It is worth mentioning that, compared with the study on filter selection [5], our work not only expands the search domain for better filters, but also saves on reconstruction time, since we do not need to calculate the sparse code online. Also, in contrast to reconstruction, our designed filters actually offer a principled lossy compression method to save on storage space for existing hyperspectral images.

Although our framework is presented for spectral reconstruction, we believe it can also be generalized to many other data-driven hardware design tasks.

4. Filter Design and Spectral Reconstruction

In this section, the details on the end-to-end network for simultaneous filter response design and spectral reconstruction will be given. We will start with the spectral reconstruction network, and later append a special convolution layer onto it to learn the filter response functions as well.

4.1. Spectral Reconstruction Network

Noted that arbitrary end-to-end network could be used for our spectral reconstruction. Here, for the sake of generality, we construct a spectral reconstruction network by adapting the well-known U-net [29] architecture, which has been widely used for image-to-image translation applications, such as pix2pix [16], CycGAN, Semantic Segmentation [31] and hyperspectral reconstruction [3]. Many previous encoder-decoder networks [27] pass the input through a series of down-sampling operations, such as maxpooling, until a bottleneck layer before reversing the process. Passing the information through these layers would inevitably sacrifice much of low-level details in the high resolution input grid. Therefore, in the image-to-image application,

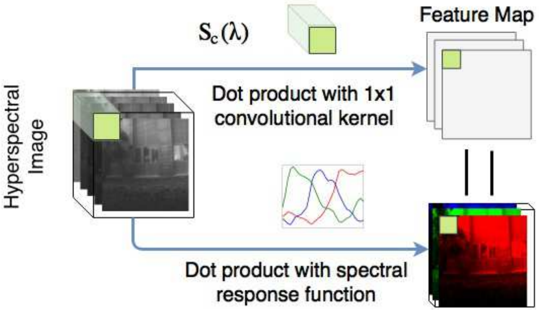

$S_c(\lambda)$
Feature Map
Hyperspectral Image
Dot product with 1x1 convolutional kernel
Dot product with spectral response function

Figure 2. Similarity between the 1×1 convolution and the filter spectral response.

the skip connection structure would allow low-level information to be directly shared across layers. Basically, the skip connection allows information to reach deeper layers as applied in [15, 14, 11]. This structure can mitigate the issue with vanishing/exploding gradients when the model is "very deep" [14]. What is more, U-net also works well on small sized training datasets [31]. This suits our application particularly well as existing hyperspectral datasets are still limited in scale.

We use modules formed as follows: 2D convolution-BatchNorm-Relu. The network takes images of size 256×256×3 as input and finally produces the corresponding spectral images of size 256×256×31. Let Ck denote a convolutional block including one convolutional layer with k filters, one leakyReLU activation layer, one BatchNormalization layer. The convolutional layer in each Ck has 3×3 sized kernels with stride 2. The downsampling factor is 2, with proper zero padding to edges. The α parameter in the leakyReLU layer is set to 0.2. CDk denotes the same block as Ck, except that the convolution layer is replaced by the deconvolution layer. It upsamples the input by a factor of 2 as well. A dropout layer with 50% dropout rate is added after each block. The whole architecture is composed as: C64-C128-C256-C512-C512-C512-C512-CD512-CD512-CD512-CD256-CD128-CD64-CD31.

Compared to a standard U-net, we modify the last layer of the U-net from 3 channels to 31 channels, and change the loss function from cross-entropy to Mean Squared Error (MSE).

4.2. Filter Spectral Response Design

As shown in Fig. 1, one key novelty of this paper is in drawing the connection between camera color imaging formulation and a convolutional layer. This allows us to optimize the spectral imaging parameters by using existing network training algorithms and tools. For simplicity, we will assume that the CCD/CMOS sensor has an ideal flat response temporarily, and will address this factor when constructing a real system.

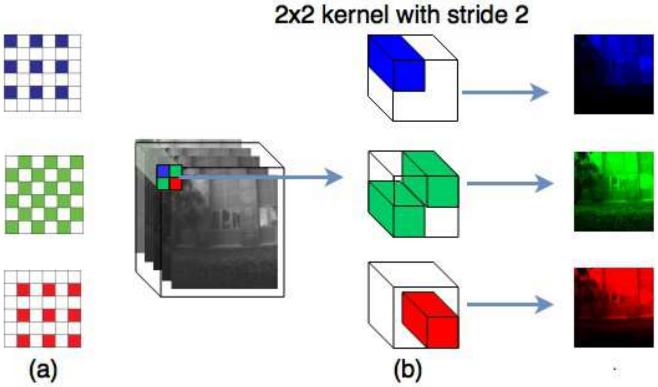


Figure 3. The typical Bayer filter array setup and our special convolution kernel for the Bayer-style 2×2 filter array design.

Given the spectral radiance $L(x, y, \lambda)$ at position (x, y), the recorded intensity by a linear sensor coupled with a color filter is given by

$$I_c(x, y) = \int_\lambda S_c(\lambda)L(x, y, \lambda)d\lambda, \quad (1)$$

where λ is the wavelength and $S_c(\lambda)$ is the spectral response function of the color filter. In most commercial cameras, there are red-green-blue trichromatic filters, *i.e.* $c \in \{R, G, B\}$, so as to mimic the human color perception.

In practice, the above equation could be discretely approximated as

$$I_c(x, y) = \sum_{n=1}^{N} S_c(\lambda_n)L(x, y, \lambda_n), \quad (2)$$

where the filter response is in the form of a vector $\mathbf{S_c} = [S_c(\lambda_1), S_c(\lambda_2), \cdots, S_c(\lambda_N)]$ at sampled wavelengths, and N is number of spectral channels.

An interesting observation is that Eq. 2 is identical to the convolution operation of a 1x1 convolution kernel in forward propagation. By regarding the filter spectral response function $\mathbf{S_c}$ as the weight of 1x1 convolution kernel, as shown in Fig. 2, the intensity $I_c(x, y)$ could be interpreted as the output activation map of a convolution, which is actually the dot product between entries of the convolution kernel (color filter) and input (incident light) $\mathbf{L}(x, y)$.

With this observation, as shown in Fig. 1, we now add a 1x1 convolution layer with three convolution kernels, which act like the three color filters in a three-channel camera. With the appended layer, we train this end-to-end network with the N-channel hyperspectral images as input. With this strategy, we can obtain the optimized spectral responses from the learned weight of the 1x1 convolution kernel.

4.2.1 Multi-chip Setup without Mosaicing

Some commercial RGB cameras adopt the multi-chip setup, that is, to have a separate color filter for each CCD/CMOS

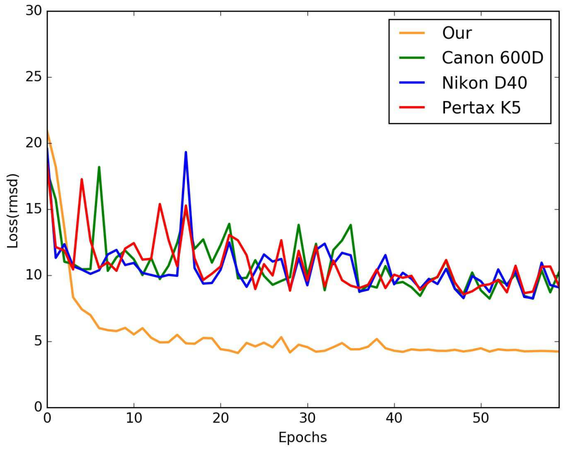


Figure 4. RMSE of each epoch of our designed and existing spectral response function on the CAVE dataset [34].

sensor. They use the specialized trichroic prism assembly. Without spatial mosaicing, they are usually superior in color accuracy and image noise than the Bayer filter array assembly in a single-chip setup. One alternative is to combine beam splitters and color filters together, as illustrated in Fig. 1, which is suitable in constructing multi-channel camera prototypes.

In this multi-chip setup, it is obvious that we can directly obtain the filter spectral response functions as described above.

4.2.2 Single-chip Setup with a 2x2 Filter Array

The majority of commercial RGB cameras have a single CCD/CMOS sensor inside, and use the 2×2 Bayer color filter array to capture RGB images with spatial mosaicing. A demosaicing method is needed to obtain full-resolution RGB images.

Our strategy could also be extended to this single-chip scenario. Inspired by the spatial configuration of the Bayer filter array, we consider a 2×2 filter array with three independent channels, and design their spectral response functions through our end-to-end network.

As illustrated in Fig. 3(a), in the Bayer filter array pattern, in each 2×2 cell, there are only one blue pixel, one red pixel, and two green pixels. We could directly simulate them with a 2×2 convolution kernel of stride 2, which is shown in Fig. 3(b). This would transform the 2x2 convolution kernel to a 1x1 convolution at a specific position. In our implementation, for the 'red' and 'blue' channels, we manually freeze 75% of the weights of the convolution filter to zero. For the 'green' channel, we only freeze half the weights to zero. Since the Bayer pattern requires two 'green' filters to share the same spectral response function, we approximate the shared spectral response function with the average anti-diagonal weight of the convolution kernel.

4.3. Nonnegative and Smooth Response

Physical restrictions require that the filter response function should be nonnegative. Also, existing film filter production technologies can only realize smooth response curves with high accuracy. Therefore, we have to consider these constraints in the numerical design process.

There are various regularizers in convolutional neural network, which were originally designed to penalize the layer parameters during training. Interestingly, our nonnegativity and smoothness constraints on the spectral response functions could be easily enforced by borrowing those regularizers.

To achieve nonnegative responses, we enforce a nonnegative regularizer on the kernel in our filter design convolution layer, such that $S_c(\lambda) \geq 0$. As for the smoothness constraint, we use the L2 norm regularizer, which is commonly used to avoid over-fitting in deep network training. Specifically, we introduce a regularization term $\eta\sqrt{\sum_{n=1}^{N}(S_c(\lambda_n))^2}$, where η controls the smoothness. Throughout the experiment, η is set to 0.02.

5. Experiment Results Using Synthetic Data

Here, we conduct experiments on synthetic data to demonstrate the effectiveness of our method. We evaluate our method on the dataset comprising of both natural and indoor scenes [34, 9].

5.1. Training Data and Parameter Setting

The CAVE [34] dataset is a popular indoor hyperspectral dataset with 31 channels from 400nm to 700nm at 10nm steps. Each band is a 16-bit grayscale image with size 512*512. The Harvard dataset [9] is a real world hyperspectral dataset including both outdoor and indoor scenarios. The image data is captured from 420nm to 720nm at 10nm steps. The image data is captured from 420nm to 720nm at 10nm steps. For the sake of clarity, we label 50 images under natural illumination the "Harvard Natural Dataset" and call the rest of the 27 images under mixed or artificial illumination the "Harvard Mixed Dataset".

In the training stage, we apply random jitter by randomly cropping 256×256 input patches from the training images. We trained our algorithm with batch size 2 and 50 iterations for each epoch. We trained the network with the Adam optimizer [21] with an initial learning rate of 0.002 and $\beta_1 = 0.5, \beta_2 = 0.999$. All of the weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02.

We run our proposed algorithms on an NVIDIA GTX 1080 GPU. Our server is equipped with an Intel(R)

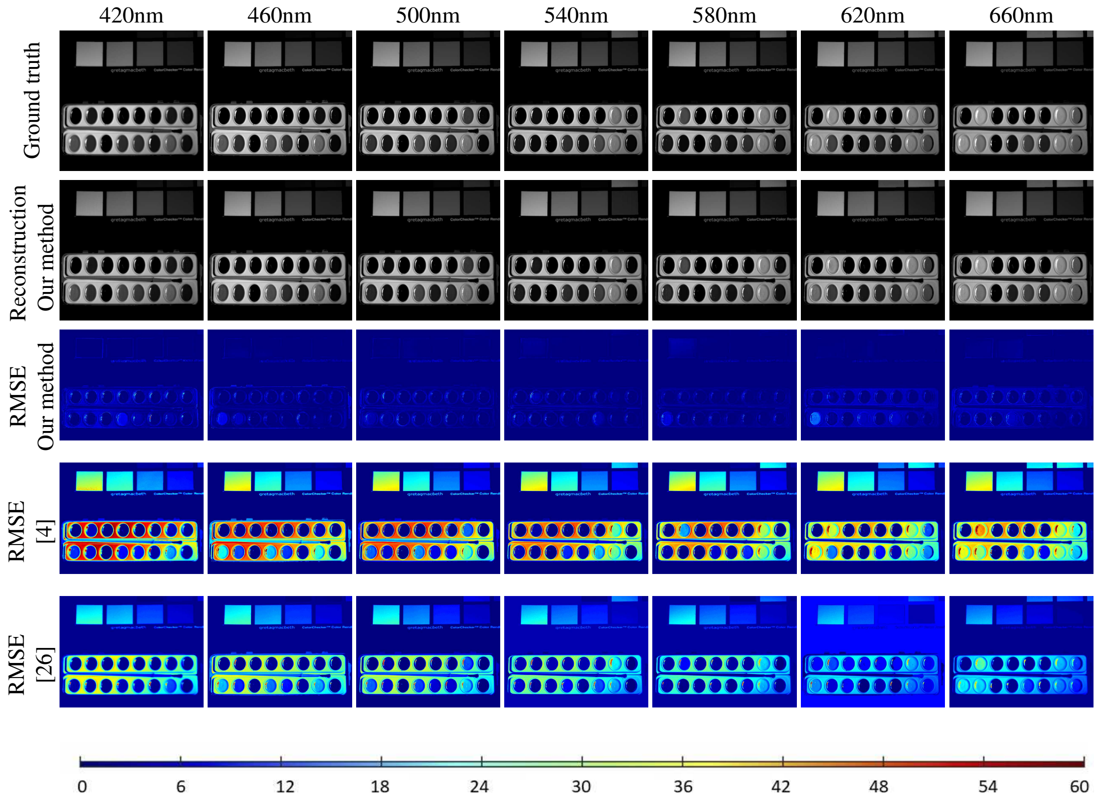


Figure 5. Sample Results from the CAVE Database [34]

Core(TM) i7-6800K CPU @ 3.40GHz and 128GB of memory. The training time for the CAVE [34], Harvard Natural and Mixed [9] datasets take $1.84, 8.88$ and 8.52 hours, respectively. The average time to reconstruct spectra from an individual image takes about 5.83 seconds.

Throughout the experiment, we choose the root mean square error (RMSE) as our evaluation metric. For each dataset, we reconstruct the hyperspectral image for all of the testing data and then calculate the average and variance of the RMSE between the reconstructed hyperspectral image and the ground truth. For the sake of consistency, we re-scale all of the spectra into range of $[0, 255]$.

5.2. Results on 3 Channel Multiple-Chip Setting

At first, we evaluate the multi-chip setup described in Sec. 4.2.1. In this section, we evaluate the performance of multi-chip setup with 3 sensors. The optimal spectral response function for the CAVE dataset [34] is given in Fig. 1.

The average and variance of the RMSE are shown in Table 1, which was compared with three baseline methods: [4], [26] and [17]. The RGB inputs of three baseline methods are generated from the spectral response function of Cannon 600D. This table shows that the RMSE of our method outperforms the alternative methods in spectral reconstruction in all three datasets.

Table 1. Average and Variance of RMSE of reconstruction on the hyperspectral databases [34, 9, 17].

	CAVE[34]	Harvard Natural[9]	Mixed[9]
Our	4.48 ± 2.97	7.57 ± 4.59	8.88 ± 4.25
[4]	8.84 ± 7.23	14.89 ± 13.23	9.74 ± 7.45
[26]	14.91 ± 11.09	9.06 ± 9.69	15.61 ± 8.76
[17]	7.92 ± 3.33	8.72 ± 7.40	9.50 ± 6.32

We also demonstrate the spatial consistency of the recovered hyperspectral images from CAVE datasets in Fig. 5 which shows images at 7 different wavelengths. The similar result on Harvard Nature and Mixed dataset could be found in supplementary material.

We also represent the recovered spectra for random points from three datasets in Fig. 6, which shows our method is consistently better than the alternatives.

To demonstrate the efficacy of our spectral response function, we also train and test our spectral reconstruction network on the RGB images generated by existing types of cameras. Here we compare the average RMSE on the testing set for each training epoch in Fig. 4.

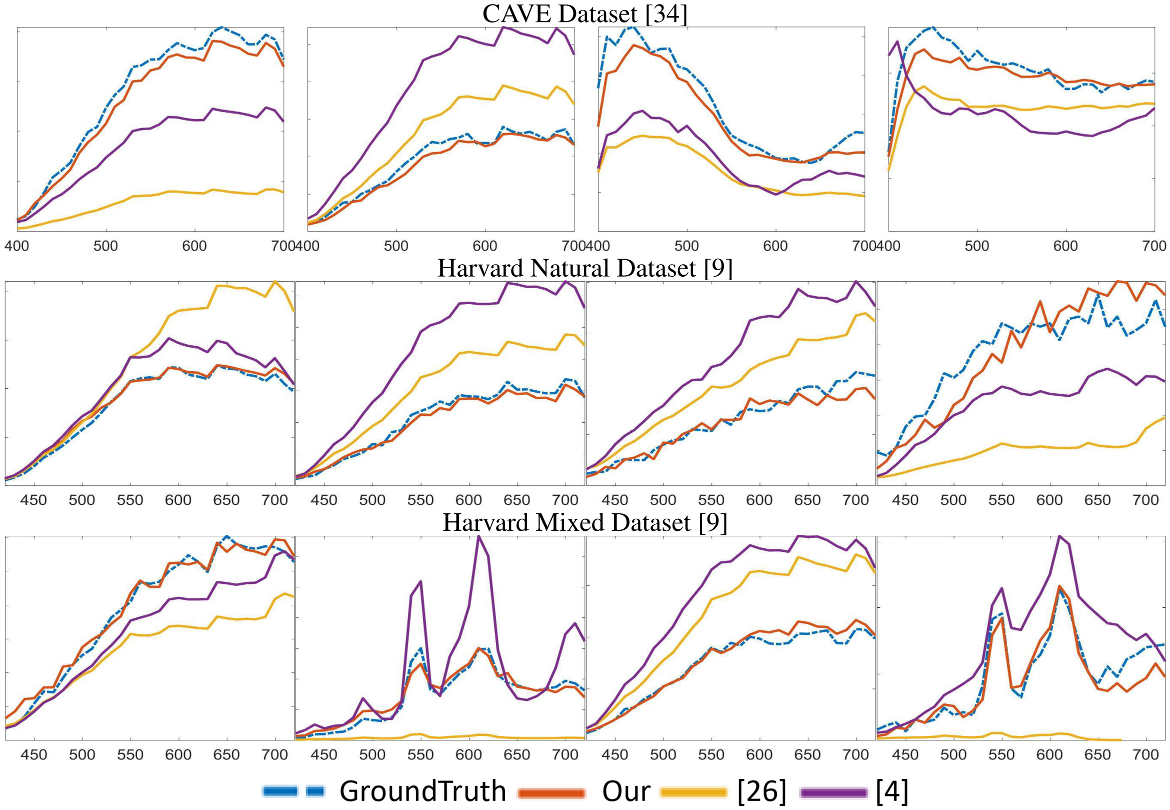


Figure 6. The reconstructed spectra samples for randomly selected pixels in the CAVE and Harvard Natural and Mixed datasets [34, 9]. Each row corresponds to its respective dataset.

As shown in Fig. 4, the reconstruction error of our method rapidly converges as the epoch increases compared to other spectral reconstruction networks based on existing camera types. Our method also shows superior performance at epoch 60.

We also discuss the robustness of our method in supplementary material. Specifically, we first report performance in the case of added Gaussian noise. Then we report the response function trained without physical constraints.

5.3. Filter Array Design for Single Chip Setting

We also demonstrate our performance in designing the filter array (Sec. 4.2.2). When comparing with the alternatives, we simulate the single-chip digital camera by encoding the image in a Bayer pattern. We then perform gradient-corrected linear interpolation, a standard demosaic method, to convert the Bayer-encoded image into the color image before conducting the comparison.

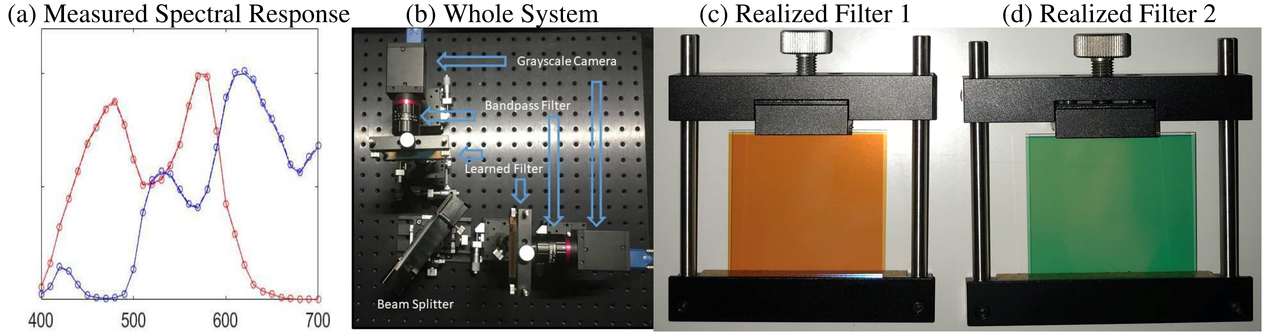
Table 2. Average and Variance of RMSE of reconstruction with filter array on CAVE dataset [34].

Our	[4]	[26]
4.73 ± 3.12	13.25 ± 13.88	18.13 ± 9.33

We present our quantitative analysis of 3 channel single-chip settings on the CAVE dataset in Table 2. The optimal spectral response function is given in 1 where the corresponding position of each spectral response function is illustrated in Fig. 3. Note that, similar to the Bayer Pattern, the spectral response colored in green covers 50% of chip. Our method maintains sufficient accuracy under the array setting where the performance of existing methods deteriorate under the demosaicing process in the single chip setting.

6. Data-Inspired Multipectral Camera

Here, we aim to construct a multispectral camera for image capture and hyperspectral reconstruction. We use the FLIR GS3-U3-15S5M camera to capture images, which collects light in the spectral range from 300nm to 1100nm. To block out UV and NIR sensitivity, we add a visible band-pass filter onto the camera lens. Since the multi-sensor setup is easier to implement than a filter array, we conduct the design operation as in Sec. 5.2. When evaluated on the CAVE dataset [34], the average RMSE of two-channel optimized filter is 5.76, slightly higher than the three-channel setup 4.48. We note both our results are still far better than the alternative algorithms based on three-channel in-

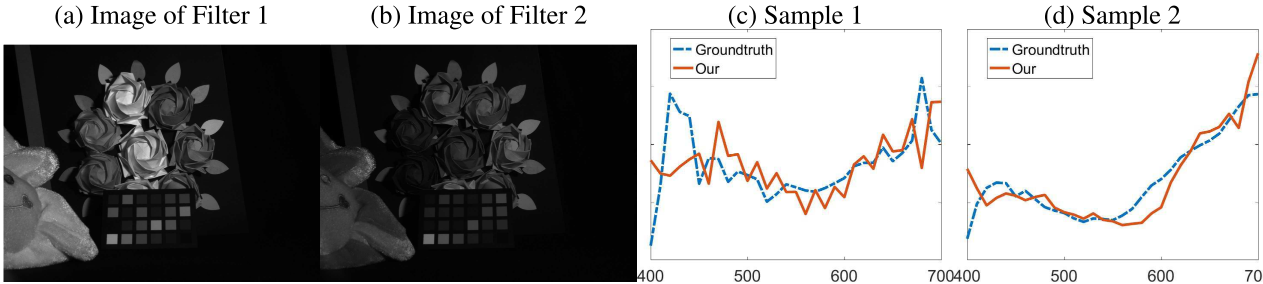

(a) Measured Spectral Response

(b) Whole System

(c) Realized Filter 1

(d) Realized Filter 2

Figure 7. The realization of our multispectral camera. (a) The measured spectral response of our designed filter trained on CAVE [34]. Circles indicate the actual response while the solid lines are the designed spectral response function. (b) Our multispectral imaging system setup. (c) Filter of (a)'s red curve. (d) Filter of (a)'s blue curve.


(a) Image of Filter 1

(b) Image of Filter 2

(c) Sample 1

(d) Sample 2

Figure 8. Results from our multispectral camera. (a,b) The captured images of filter 1 and 2, respectively. (c,d) The reconstructed spectra of randomly selected pixels.

put. Due to the expensive cost in customizing filters, here we choose to realize the designed filters in the case of two channels, whose response functions are shown in Fig. 7(a). We turned to a leading optics company to implement the designed response functions. The realized film filters are of size 50mm×50mm×1mm (see 7(c,d)), and the measured spectral response functions are shown in Fig. 7(a) (Solid line indicates designed response and circles indicate actually measured response). The film filter is an interference filter consisting of multiple thin SiO2 and Nb2O5 layers. With the interference effect between the incident and reflected lights at thin layer boundaries, the designed film filter endows us spectral response functions that are very close to our design. We use a 50-50 beamsplitter to construct a coaxial bispectral camera, and align two FLIR GS3-U3-15S5M cameras properly as illustrated in Fig. 7(b). Sample images captured through two filters are shown in Fig. 8(a,b). We also report the reconstructed spectra via our system compared to the ground truth. Consistent with the previous simulations, our reconstructions are reasonably accurate, as shown in Fig. 8(c,d).

7. Conclusion

In this paper, we have shown how to learn the filter response functions in the infinite space of nonnegative and smooth curves by using deep learning techniques. We appended a specialized convolution layer onto the U-net based reconstruction network, and successfully found better response functions than standard RGB responses, in the form of three separate filters and a Bayer-style 2x2 filter array. For building a real multispectral camera, we have also incorporated the camera CCD included responses into the design process. We successfully designed/implemented two filters, and constructed a data-inspired bispectral camera for snapshot hyperspectral imaging.

At the very beginning of this research, we were speculating that, given a proper dataset, the deeply learned responses should finally converge to the color matching function of human eyes, since the latter has been "optimized" in the long history of evolution. However, we observed in our current experiments that the learned response functions might vary significantly from one training dataset to another. We will leave the collection of a comprehensive database as our future work. Meanwhile, we will also extend this work to optimize the camera for a wider range of vision tasks such as classification [6].

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Number JP15H05918.

References

[1] N. Akhtar, F. Shafait, and A. Mian. Sparse Spatio-spectral Representation for Hyperspectral Image Super-resolution. In *Proc. of European Confer-*

ence on Computer Vision (ECCV), pages 63–78, Sept. 2014.

[2] N. Akhtar, F. Shafait, and A. Mian. Hierarchical beta process with gaussian process prior for hyperspectral image super resolution. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 103–120, Oct. 2016.

[3] A. Alvarez-Gila, J. van de Weijer, and E. Garrote. Adversarial networks for spatial context-aware spectral image reconstruction from rgb. *IEEE International Conference on Computer Vision Workshop (ICCVW 2017)*, 2017.

[4] B. Arad and O. Ben-Shahar. Sparse Recovery of Hyperspectral Signal from Natural RGB Images. *ECCV*, pages 19–34, 2016.

[5] B. Arad and O. Ben-Shahar. Filter selection for hyperspectral estimation. In *ICCV*, pages 3172–3180, 2017.

[6] H. Blasinski, J. Farrell, and B. Wandell. Designing illuminant spectral power distributions for surface classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2164–2173, 2017.

[7] X. Cao, H. Du, X. Tong, Q. Dai, and S. Lin. A Prism-Mask System for Multispectral Video Acquisition. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 33(12):2423–2435, 2011.

[8] X. Cao, X. Tong, Q. Dai, and S. Lin. High resolution multispectral video capture with a hybrid camera system. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 297–304, June 2011.

[9] A. Chakrabarti and T. Zickler. Statistics of Real-World Hyperspectral Images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 193–200, 2011.

[10] H. G. Chen, S. Jayasuriya, J. Yang, J. Stephen, S. Sivaramakrishnan, A. Veeraraghavan, and A. Molnar. Asp vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels. In *CVPR*, June 2016.

[11] K. S. e. a. D Silver, J Schrittwieser. Mastering the game of go with deep neural networks and tree search. *Nature*, 2017.

[12] M. T. Eismann. *Hyperspectral Remote Sensing*. 2012.

[13] L. Gao, R. Kester, N. Hagen, and T. Tomasz. Snapshot image mapping spectrometer (IMS) with high sampling density for hyperspectral microscopy. *Optical Express*, 18(14):14330–14344, 2010.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] Y. Jia, Y. Zheng, L. Gu, A. Subpa-Asa, A. Lam, Y. Sato, and I. Sato. From rgb to spectrum for natural scenes via manifold-based mapping. In *ICCV*, pages 4715–4723, october 2017.

[18] C. Jiang, H. Zhang, H. Shen, and L. Zhang. A Practical Compressed Sensing-Based Pan-Sharpening Method. *IEEE Geoscience and Remote Sensing Letters*, 9(4):629–633, July 2012.

[19] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk. What is the space of spectral sensitivity functions for digital color cameras? In *WACV*, 2013.

[20] R. Kawakami, J. Wright, Y.-W. Tai, Y. Matsushita, M. Ben-Ezra, and K. Ikeuchi. High-resolution hyperspectral imaging via matrix factorization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2329–2336, June 2011.

[21] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015*, pages 1–15, 2015.

[22] H. Kwon and Y.-W. Tai. RGB-Guided Hyperspectral Image Upsampling. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 307–315, 2015.

[23] C. Lanaras, E. Baltsavias, and K. Schindler. Hyperspectral Super-Resolution by Coupled Spectral Unmixing. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 3586–3594, 2015.

[24] G. Lu and B. Fei. Medical hyperspectral imaging: A review. *Journal of Biomedical Optics*, 2016.

[25] H. Matsuoka, Y. Kosai, M. Saito, N. Takeyama, and H. Suto. Single-cell viability assessment with a novel spectro-imaging system. *Journal of Biotechnology*, 94(3):299–308, 2002.

[26] R. M. H. Nguyen, D. K. Prasad, and M. S. Brown. Training-based spectral reconstruction from a single RGB image. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8695 LNCS(PART 7):186–201, 2014.

[27] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] A. Robles-Kelly. Single image spectral reconstruction for multimedia applications. In *the 23rd ACM International Conference on Multimedia*, pages 251–260, 2015.

[29] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

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

[31] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017.

[32] A. Wagadarikar, N. Pitsianis, X. Sun, and B. David. Video rate spectral imaging using a coded aperture snapshot spectral imager. *Optical Express*, 17(8):6368–6388, 2009.

[33] Z. Xiong, Z. Shi, H. Li, L. Wang, D. Liu, and F. Wu. Hscnn: Cnn-based hyperspectral image recovery from spectrally undersampled projections. *The IEEE International Conference on Computer Vision (ICCV)*, pages 518–525, 2017.

[34] 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, Sept 2010.

[35] X. Zhu and R. Bamler. A Sparse Image Fusion Algorithm With Application to Pan-Sharpening. *IEEE Trans. Geoscience and Remote Sensing*, 51(5):2827–2836, May 2013.