Physics-based Deep Learning for Optical Property Analysis

by

Nie Shijie

Dissertation

submitted to the Department of Informatics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI
March 2020

Abstract

One of the important goals in computational photography is to capture and represent how lights interact with real scenes. To overcome the limitations of ordinary cameras, techniques such as combining imaging and data-driven computation to simulate optical processes have attracted a great deal of attention. This thesis are going to explore two essential parts of optical properties: light spectral properties and light transport properties, and we aim to infer them with an image taken by ordinary cameras. The two representations of these optical properties is hyperspectral image and global/direct illumination.

Hyperspectral imaginary and global/direct illumination imaginary is costly and requires complex hardware setting. For direct and global illumination, capturing them requires modulated active light and multiple images. To capture hyperspectral image, Hyperspectral Imaging Systems (HISs) with complex hardware setting is required. For a normal RGB three channels image taken by ordinary camera, reconstruct these two optical properties by RGB is ill-posed. The three channels dimensions image is a many-to-three mapping: for direct/-global separation is six to three, for spectral reflectance is a number of spectral bands to three. This image is a subspace of the two high dimension space, and back-projection mapping is highly ill-posed. Several machine learning and deep learning methods were proposed, but their performance is not satisfying. The major limitation of deep learning is transparency. We proposed a framework, by encoding physical optical process into a deep learning network, the acquisition and analysis process is jointly learned. We carefully designed the loss and structure of deep learning architecture to get better results, and hardware implemented the first layer of network to capture three channels image for inference. By replacing complex hardware setting with our proposed deep learning network, the computation and acquire cost is reduced greatly.

This thesis propose the first method to analyze global and direct components from a single RGB image without any hardware restriction. My method is a novel generative adversarial network (GAN) based networks which impose the prior physics knowledge to force a physics plausible component separation. In

the experiments, my method has achieved satisfactory performance on images from our own testing set with global and direct component and hyperspectral analysis public datasets.

Furthermore, to analyze the hyperspectral image, as existing RGB cameras are tuned to mimic human trichromatic perception, we optimize a new spectral response that is necessary for hyperspectral reconstruction. We learn the 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 four types of designed filters: a three-chip setup without spatial mosaicing, a single-chip setup with a Bayer-style 2x2 filter array, a non-invasive filter learning approach combined with existing camera response functions, and a jointly learned camera sensor coded spatial pattern and response function. Numerical simulations verify the advantages of deeply learned spectral responses over 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. Finally, we simultaneously learn the camera spectral response (CSS) functions and a material classification network. We show that the proposed method has higher overall accuracy than existing cameras due to CSS optimization.

Acknowledgements

When I was a high school student, my dream was to get in one of the top universities in China and find a decent job. After entering the University of Science and Technology of China, my alumni who are top researchers around the world gave me a huge impression about their pure passion of pursing the truth and discovering unknown knowledge. I want to be one of them, then I went to Tokyo where I stayed three years for pursing my own truth.

I am especially grateful to Professor Imari Sato for her delicate supervision of my research. Before I joined the Programming Research Lab, I have no experience of doing research. Professor Imari Sato patiently guided me to survey the related works and finally find the key research problem. She always gives me wise suggestions and strong support, she taught me how to write a good research paper hand by hand, she cared about me even more than myself, she is a role model for me not only in research but also how to behave.

I must thank Dr. Lin Gu, a tallent project researcher in our lab. Dr. Lin Gu gives me a lot of useful advices and always asks important questions during my presentation to ensure I am really clear about the technical detail. Dr. Lin Gu carefully introduces basic knowledge such as image processing and machine learning for me to start my research life. He always check my drafts seriously to make sure I do not have any misunderstanding and gives me a lot of useful suggestions. We have so much fun time during coffee break and he is really good at playing Civilization V.

I must also thank our Prof. Yinqiang Zheng. He is really an extremely excellent researcher and very productive. Even he is not one of my supervisors,

but he really did a lot of work in helping me sort out my research, giving advices, and guiding me.

I would like to thank the members of my dissertation committee, Professor oyama, Professor Shinichi Satoh, Professor sugimoto, and Professor yqzheng for giving insight comments.

I would like to thank all other members in our lab, Mihoko Shimano, Lixiong CHEN, Art SUBPA-ASA, Yuta Asano, Shin Ishihara for having a great research life together. Some of them are co-author for my publications. And to all the intern students that helped me a lot through the journey.

Lastly, I would like to thank my parents, my girlfriend Chunmiao Li supporting me all the time, without your support I could not go so far.

Shijie Nie National Institute of Informatics & Tokyo, 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter contains three sections. First, a general overview of this thesis is proposed. Then, the detailed methods for connecting each optical component with physical-based deep learning are discussed. Finally, the thesis objective and main contribution are described.

1.1 Overview

Light interacts with the real scene in many ways, for example, absorption, reflection, diffusion, dispersion, etc. In this thesis, we discussed two major representations of optical processes: hyperspectral image and global/direct illumination. The conventional method to capture them requires complex hardware settings, for example, high-frequency patterns, spatial scanner, spectral scanner, and so on. After the capturing process, analyzing process was applied to extract discriminative features and infer the important information of real scene that cannot be easily inferred from RGB image. For instance, the direct component gives us the information of how the material properties of a scene point interact with the source and camera. The global components gives us the information of how photon interact between different objects and media in the

scene [95]. This information can be used for computer vision and graphics field such as image editing, rendering and shape from shading. The hyperspectral images also contains extra information of light wavelength axis. It is shown to be beneficial for remote sensing, medical diagnosis, industrial detection, and so on [25, 84]. For example, the tumor margin, invisible to the surgeon's eyes, could be better visualized in hyperspectral images. Cases of leaked invisible gas may also be obvious using spectral signals.

Since it is hard to capture these optical features and its attractive applications, recovering and analyze them through ordinary cameras has attracted much attention in the computer vision community. Given the same exposure time and sensor scale, an RGB image captured by latest imaging has much higher resolution and signal-to-noise ratio than a hyperspectral captured by hyperspectral sensors. If we get global and direct components from one image, the resolution is also reduced [95]. To overcome the limitations for acquisition of optical properties, traditional methods attempt to reconstruct optical properties using machine learning and deep learning methods, such as support vector machine (SVM), convolution neural network (CNN), generative adversarial network (GAN), etc. However, the majority of them did not consider the optical process as prior knowledge of machine learning methods, and their performance is limited.

Deep learning has been recently thriving in many research fields: from recognition objects in images [114, 44, 119], transformations between text and sound [43, 135], to accelerate medical research by predicting potential drug molecules, DNA mutation on gene [148], or even master the Go game [27] etc. Nearly in every field, deep learning-based methods achieve state-of-the-art performance than conventional methods. However, the No-Free-Lunch theorem [123, 47, 51, 142, 143] indicates that "Roughly speaking, we show that for both static and time dependent optimization problems, the average performance of any pair of algorithms across all possible problems is exactly identical.", quote as in [143]. In other words, each algorithm has a set of applicable problems, and efficiency of it should be slower than random search for other methods. According to this theorem, we believe that embedding domain-specific knowledge into models would largely compress the assumption space

and achieve more beneficial results for a specific problem. Optical properties, which define how objects interact with light, obeys several physical models. As shown in Fig. 1.1, the aim of this thesis is to improve performance for analyzing optical properties of real scenes shown in yellow arrows. The focus of this thesis is to combine optical process and its features with deep learning neural network in black arrows. More specifically, this thesis raised three principal problems: 1) How to recover global and direct components from a single RGB image. 2) How to analysis hyperspectral data from a single image taken from an optimized camera. 3) How to combine camera acquisition process with deep learning network structures. I will precise the objectives of this thesis work in the following sections.

1.2 Physical-based Deep Learning

The major limitation of deep learning models is it might be difficult to understand what is going on inside. Without relying on scientific knowledge, deep learning model is a ”black-box”, only from trial and test on data. However, scientific problems are always under physical constraints. For example, fluid dynamic obeys Navier-Stokes equations, and Lagrangian mechanics obeys Euler-Lagrange equations, computational acoustics obeys the Helmholtz equation. To solve these ordinary differential equations (ODE) and Partial differential equations (PDE), traditional off-the-shelf models are using numerical analysis and require a large number of computational resources. As a more efficient and derivable optimizing structure, deep learning modules intuitively proposed to describe these physical processes by encoding a differential equation. By concatenating these modules with existing generic deep learning models, researchers developed many end-to-end networks for various tasks [83, 90, 140, 107, 138, 156, 161, 125, 39, 49, 81, 126, 87, 76, 59, 162, 64, 65, 167, 147]. The proposed end-to-end network amplifies information from training examples and converge much faster, provide more accurate results, and simultaneously ensuring physical plausibility. Following previous practice on applications of encoding physics before deep learning, we impose optically properties con-

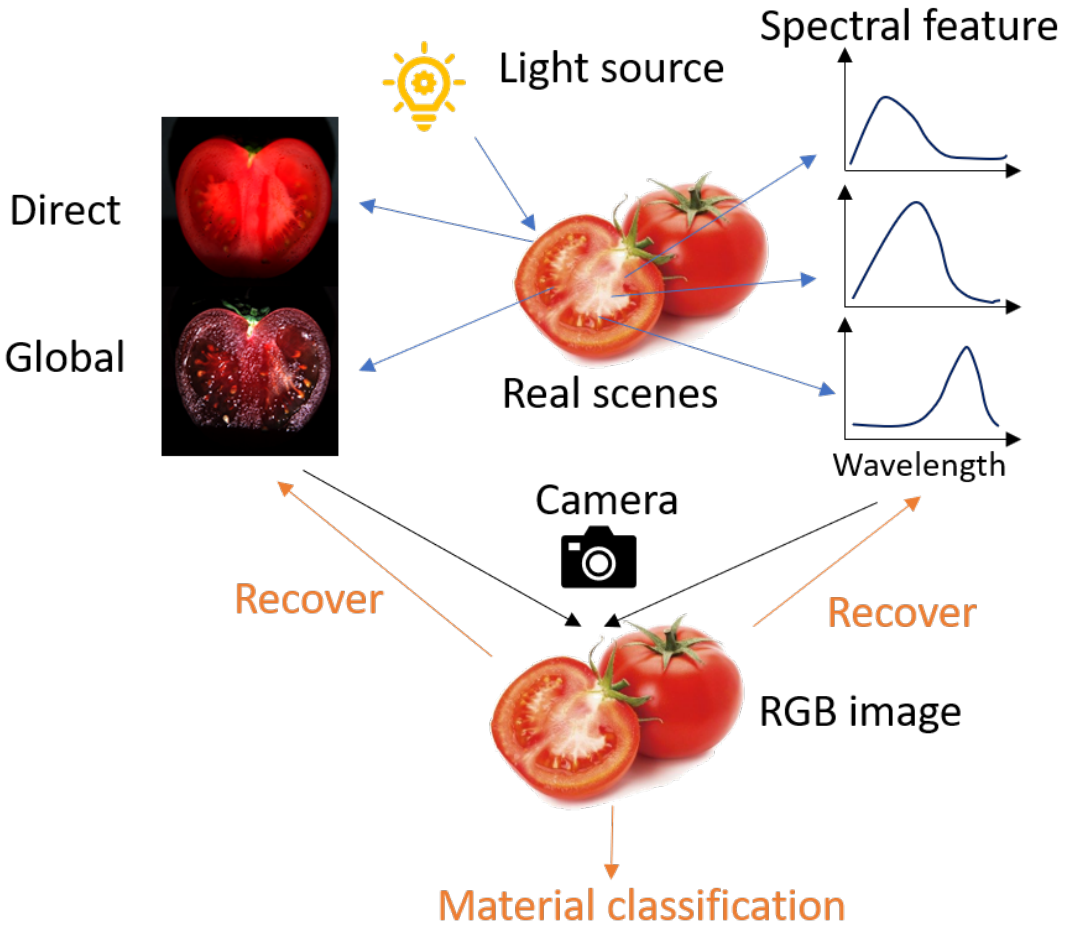


Figure 1.1 Schematic diagram of the thesis concept. Hyperspectral image and global/direct illuminations are optical properties shown in blue arrows. RGB images taken by ordinary camera are shown in black arrows. As optical properties contain rich information and RGB are easy to capture, recovering the optical properties from RGB is proposed. The focus for this thesis is to embedding domain-specific information of camera acquisition process shown in black arrows into deep learning neural network. The aim of this thesis is to enhance the analysis of optical properties shown in yellow arrows. Note the material classification is an essential application of hyperspectral analysis and can be improved using our method.

straint into three deep learning networks: a generative adversarial network, a convolutional neural network for material reconstruction and classification. Experiments show that our algorithm is highly effective and has outperformed state-of-the-art method generic deep learning and traditional machine learning algorithms without any optical properties constraints. These methods focus on solving three optical feature analysis problems: recovering indirect/direct illumination components proprieties from a single RGB image, recovering spectral reflectance, and spectral reflectance classification.

1.3 Impose Physical Inverse Loss for Indirect/direct Illumination Component Separation

When there is a light source, the radiance captured by a camera is the sum of both direct and global components. The direct component is the direct reflectance of the light from the source on the surface (Fig 1.2.(3)). The global component is the indirect lighting from complex phenomena such as inter-reflection, subsurface scattering, volumetric scattering, and diffuse (Fig 1.2.(4)). Measuring these two components has attracted wide attention by both computer vision and graphic community.

Traditionally, separating above two components requires multiple images taken under specific setting such as high-frequency light patterns [96, 37, 1, 124, 102, 103, 68]. In this thesis, I separate the two components directly from a single image without any hardware constraints and also present a dataset including 100 scenes with their direct and global components.

Our method is a novel generative adversarial network (GAN) based networks which imposes the physical inverse loss to force a physics plausible component separation. Since each separated component carries much information about both the scene and the environment, our method and the dataset would benefit both computer vision and graphic community. For example, the direct component conveys the information about the interaction between material properties,

geometry, and lights. Obtaining pure direction component would enhance the computer vision task, such as material recognition [80], depth recovery [42], shape reconstruction [91] and colour constancy [134]. For the global component, it plays an important role in rendering realistic scenarios [21]. This knowledge would also endow us a better image manipulation algorithm [10]. What is more, since global component reflects complex interaction amid the environment, separating it could also reveal the surrounding environment [34] by treating the foreground object as a complexly shaped and far-from-perfect mirror.

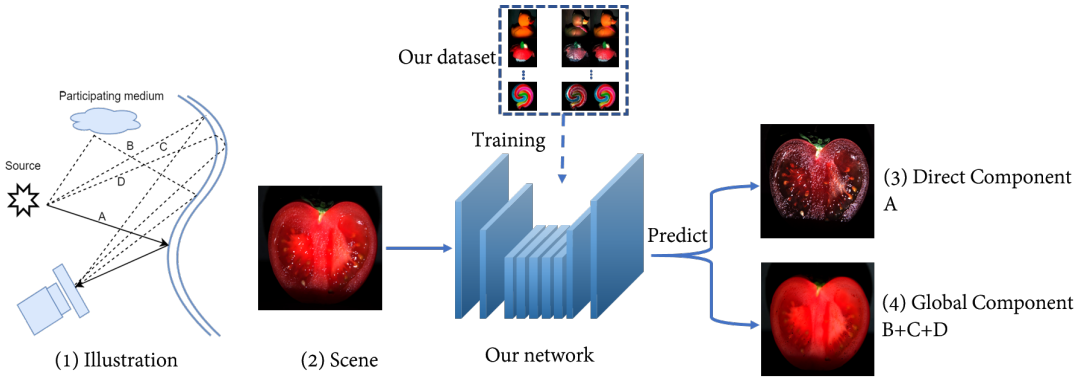


Figure 1.2 (1): The captured radiance of the scene is due to directional illumination (A) from source and global illumination (B + C + D). The global illumination may arise from volumetric scattering (B), subsurface scattering (C) and subsurface scattering (D). (2): A scene lit by a single light source. (3) Direct component directly from the light source (4) Global component arising from complex global illumination effects.

1.4 Impose Physical Based CNN structure for Spectral Reflectance Reconstruction and Hardware Design

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 [25, 84]. For example, the tumor margin, invisible to the 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 [88] and random/regular aperture masks [136, 33, 11]. Unfortunately, these devices are extremely limited in spatial resolution.

A computational hyperspectral reconstruction method for a single RGB image is promising in overcoming the drawbacks of the devices mentioned above, as evidenced in recent research on RGB-to-Spectrum reconstruction [97, 111, 5, 54, 32, 4, 151]. 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 [56], thus their spectral response functions are not necessarily optimal for computer vision tasks, *i.e.* hyperspectral reconstruction. Very recently, Arad and Ben-Shahar [6] identified the dependence of hyperspectral reconstruction accuracy on the camera's spectral response. In [6], they find the best filter combination among a finite set of candidate filters via brute force search and hit-and-run evolutionary optimization. We learn the optimized camera spectral response functions (to be implemented in hardware) and a mapping for spectral reconstruction by using an end-to-end network. This is achieved by modifying the behavior of convolution neural

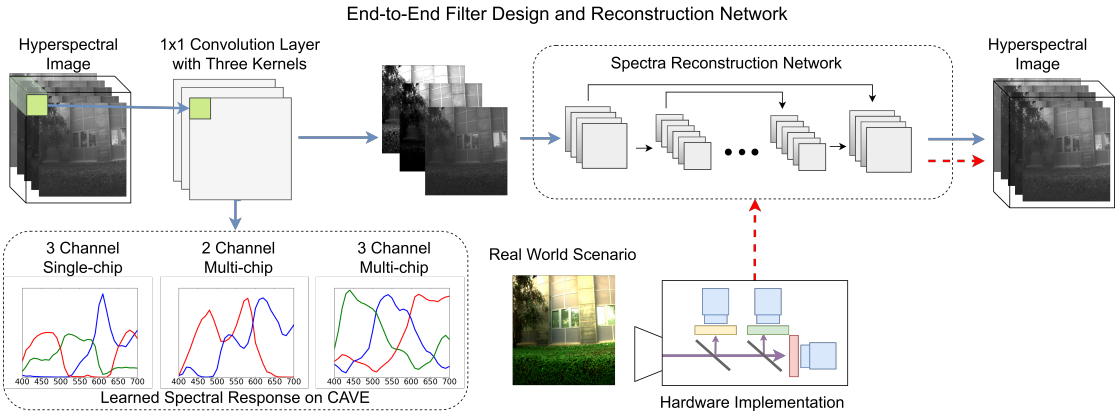


Figure 1.3 Our proposed design-realization-application framework for data-inspired spectral imaging hardware. The design stage (marked with blue arrow) is data-driven. It includes an end-to-end network to simultaneously learn the filter response and spectral reconstruction mapping. The learned spectral response function on CAVE dataset is also shown. On the realization stage (marked with a 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.

network layer.

1.5 Impose Physical Based CNN structure for Spectral Reflectance Classification and Filter Design

A typical application for hyperspectral image is material classification and remote sensing. Previous work [116, 163, 9] showed a different illumination would influence the accuracy, and it can be optimized for a specificated task. Lam *et al.* [70] proposed a method using basis lights derived from the camera response function witch uses Independent component analysis and nonnegative linear model analysis, their method does not require nonnegative constraints and shows that negative lights could be physically possible captured in real scenes. In recent years, machine learning-based methods [116, 131, 159, 160, 85, 60, 166, 35, 57, 141, 62] are applied in multispectral image segmentation. With these methods, the accuracy of public datasets is largely improved. However, capturing the hyperspectral image is slow and costly, and all the methods did not consider jointly learning hyperspectral acquiring and classification. With our method, inputs can be replaced with an RGB image using learned camera spectral response. This method leads to a faster inference and easier hyperspectral image acquisition for a specific task.

1.6 Smooth Constraint for Physical Possible Hardware Implementation

The latest film filter production technologies have allowed us to implement image sensors with any nonnegative and smooth spectral response functions. Therefore, rather than selecting filters from existing ones, in this thesis, we aim to directly learn optimized spectral response functions in the infinite space of

nonnegative 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. 4.5).

Based on our observation that camera spectral filters act in effect, like the convolution layer in neural networks (details in Sec. 4.1.3), we can optimize them by using deep learning techniques. We simultaneously learn the optimized filter response functions and the mapping for spectral reconstruction and classification 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 cameras.ASP Vision by Chen *et al.* [18] also provides evidence that the first layer of the CNN network can be implemented in hardware.With the deeply learned filters, we propose our data-inspired multispectral camera for snapshot hyperspectral imaging. The brute force search and random evolutionary optimization strategies in [6] are no longer feasible since the searching space is tremendously huge, and not implausible, especially for network-based methods.

Our contribution is regarding the camera spectral filter as the hardware implementation of the convolution layer in neural network, which will be detailed in Sec 4.1.3. With this could act in effect like the convolution operation. Thus their response functions can be automatically optimized by using the deep learning algorithms powered by a bundle of network architectures and computational hardware. More interestingly, by considering the CCD/CMOS sensor spectral response in the design process, we practically produced the filters bearing the deeply learned response functions using interference filter technology. With the designed filters, we propose a data-inspired multispectral camera for snapshot hyperspectral imaging.

1.7 Thesis Outline

The remaining parts of this thesis are organized as follows. In Chapter 2, we provide a literature review of previous work related on application of machine learning algorithm on analysis optical features and there limitations. In Chapter 3, we designed a physical-based framework to separate direct and global component based on a generative adversarial network. In Chapter 4 includes our end-to-end network for simultaneous learn filter design and hyperspectral reconstruction. Chapter 5 shows our proposed method jointly learned filter selection and material segmentation network. Finally, we conclude our research in Chapter 6 and discussed possible future research.

1.8 Contribution

In this dissertation, our contributions can be summarized as follows:

1. We explore ways to capture and analysis optical properties using images contain three channels or less. We propose a framework to optical property analysis from one single three dimensions image. We train our proposed network using hyperspectral and global/direct images and using only RGB image for inference.

2. We impose physical inverse loss into a novel generative adversarial network (GAN) based algorithms and achieves satisfactory results on our global/direct illumination dataset and public hyperspectral dataset.

3. We build 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. 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.

4. We propose four setups for optimized filter design: a three-chip setup without mosaicing, a single-chip setup with a Bayer-style 2x2 filter array, a non-destructive filter learning in conjunction with existing camera response, optimize sensor pattern spatial multiplexing and spectral response. We demonstrate that the deeply learned response functions are better than standard RGB responses in a specific computer vision task, *spectral reconstruction*.

5. We realize the deeply learned filters by using interference film production technologies and construct a snapshot hyperspectral imaging system.

6. We extend our method to the hyperspectral classification network and get superior performance.

Chapter 2

Related Work

In this chapter, I will provide an introduction of related work of traditional machine learning and deep learning based methods for analyzing optical property.

2.1 Conventional Machine Learning in Optical Analysis

In this section, I will provide an introduction of conventional machine learning methods such as SVM, PCA and NNMF in optical analysis.

Support Vector Machine (SVM)

Support Vector Machine (SVM) [26] a widely used machine learning method for image classification, segmentation and other tasks [17, 35, 58, 116, 131, 60, 29, 8, 62]. SVMs construct a margin separator which finds a hyperplane maximum the distance between features. Taking a hyperspectral image (HSI) for example,

we extract per-pixel HSI for training and testing. The objective function of input HSI $\vec{x}_i$ is:

$$\min_{\omega,b}[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\vec{\omega}\cdot\vec{x}_i + b))] + \lambda||\vec{\omega}^2|||], \tag{2.1}$$

where ω, b is the weight and bias, y_i is the label.

The supporting vector is the two hyperplanes obey $1 - y_i(\vec{\omega}\cdot\vec{x}_i + b) = \pm 1$, which is based on the assumption that feature space is linearly separable. However, existing experiment data are always no-linear with higher dimensional space. To find a hyperplane that can separate them, the kernel trick was proposed. The kernel trick stands for the inner product to transform data to linear separable space; most used kernel functions are polynomial, Gaussian kernel, and Radial basis function (RBF) kernel. We rely on scikit-learn, a python machine learning library built on LibSVM [17] to implement proposed models.

In the literature, different SVM based algorithms were applied for hyperspectral image classification. Kavitha *et al.* [60] proposed a method doing Remote Sensing dataset classification using SVM classifier, with Gabor wavelet as pre-processing method. Moreover, multiple SVM system was proposed for hyperspectral classifier image using Naive Bayes as a classifier fusion [8]. Markov random fields (MRFs) are a common practice for integrating spatial context into the classifier. In the MRF framework, the maximum a posteriori (MAP) decision rule is typically formulated as the minimization of a suitable energy function, and a combination method of SVM and MRF was proposed [131] to improve classification accuracy. Another widely used segmentation method is statistical region merging (SRM). By introducing a hierarchical version of statistical region merging (SRM) into the MRF and SVM based model, the segmentation maps can be built into different scales [35]. Saragadam *et al.* [116] proposed a method Using SVM to learn a coded illumination for spectrum feature extraction. Fauvel *et al.* [29] proposed a morphological transformations based method to do dimension reduction and boundary feature extraction.

Principal Component Analysis (PCA)

PCA is a standard method used in machine learning algorithms for data dimension reduction and produces a compact representation. Technically speaking, PCA uses an orthogonal transformation to convert a set of possibly related variables to a linearly unrelated variables called principal components. The first principal component has the largest possible variance, and each succeeding component has the highest variance and orthogonal to the preceding one. By preserving first n components, the data set size is reduced, and the most significant features that contribute to variance is preserved. PCA can be done by eigenvalue decomposition of a data covariance matrix or singular value decomposition (SVD) of the data matrix. Tipping *et al.* [132] implemented the PCA model based on a probability model and determine principal axes through maximum likelihood in latent variable space. As a common unsupervised method, PCA does not require similar training labels, but the classification accuracy deteriorates compared to similar method Linear discriminant analysis (LDA), theoretical and experimental evidence by [109]. However, PCA is still a common practice for data pre-processing in illumination and reflectance estimation [149], surface and material classification [9], hyperspectral image super-resolution [157], hyperspectral image classification [141], hyperspectral image band selection [16], and as baseline for several hyperspectral analysis methods [149, 164, 159, 45, 63, 58].

Non-Negative Matrix Factorization (NNMF)

Non-negative matrix factorization (NMF) has become a widely used instrument for analyzing high-dimensional data because it automatically extracts sparse and meaningful features from a set of non-negative data vectors. NMF is an algorithm to approximate a matrix $\mathbf{X}$ with a low-rank matrix approximation such as $\mathbf{X} \approx \mathbf{WH}$. where $\mathbf{W} \in \mathbb{R}^{p \times r}$ and $\mathbf{H} \in \mathbb{R}^{r \times n}$. r is much smaller than p and n. The interpretation of $\mathbf{W}$ is that each column is a basic element. By basis element, we mean some component that crops up again and again in all of the n original data points. These are the fundamental building blocks from which

we can reconstruct approximations to all of the original data points.

The interpretation of **H** is that each column gives the 'coordinates of a data point' in the basis **W**. In other words, it tells how to reconstruct an approximation to the original data point from a linear combination of the building blocks in **W**. A popular way of measuring how good the approximation **WH** is the Frobenius norm (denoted by the F subscript you may have noticed). The Frobenius norm is:

$$||\mathbf{X} - \mathbf{WH}||_F^2 = \sum_{i,j} (\mathbf{X} - \mathbf{WH})_{ij}^2. \tag{2.2}$$

The assumption of NMF is to extract sparse and easily nonnegative factors mechanically. It is considered that matrix elements such as user purchases or visits from different stores are all positive values, so non-negativity needs to be considered when reducing dimensions, and NMF non-negative matrix factorization just meets this kind of problem. In image processing, a face image can be squash into several basis images by NMF. The **W** can be treated as collection of images, and **H** tells the parameters to sum them up. This idea is similar to the PCA method. However, each basis images in PCA will produce a whole face, while NMF is a parts-based representation. Another difference is weights of NMF is nonnegative. When it comes to hyperspectral image, a hyperspectral image usually has 30 to 200 wavelength bands, which shows the corresponded incident light reflected by the pixel, which contains a large amount of data redundancy. To make data dimension reduction, NMF provides basis vectors and spectral signatures of vectors as where. Kawakami *et al.* [61] proposed a method for using NMF to provide an unmixing algorithm and estimate a basis representations. In the inference stage, these representations are used in conjunction with RGB input to produce the desired output of the hyperspectral image. Coupled nonnegative matrix factorization (CNMF) [155] is proposed to fusion hyperspectral data. This method takes Low-spatial-resolution hyperspectral and high-spatial-resolution multispectral data into account and unmixing them separately. By sharing the weight of two bases, this method can produce high-quality fused data. NMF is also applied for illumination and reflectance spectral separation of hyperspectral image. Zheng [164] proposed a Low-Rank Matrix Factorization with nonnegative constraint

due to the physical restrictions of illumination and reflectance spectra. The idea of coupled matrix factorization can be applied for fuse RGB and hyperspectral image, using high resolution RGB image and low resolution hyperspectral [71].

2.2 Deep Learning

In this section, I will provide a review of deep learning based methods and its applications for analyzing optical properties. As a new area of machine learning study, deep learning has been proposed to be closer to artificial intelligence. Deep learning mainly concerns learning multiple layers of representation and abstraction, which help to explain different data formats, such as images, text, and sound. Deep learning is a combination of many techniques, including hierarchical probabilistic models, neural networks, and various supervised and unsupervised feature learning algorithms.

The development of neural networks comes from the idea of constructing a system that can simulate the human brain. McCulloch and Pitts [89] proposed that neurons, a name for interconnected basic cells, can be used by human brain to generate highly complex patterns. They created a model called an MCP model, which has a significant influence on the research of artificial neural networks. Some milestones in the field, such as LeNet [73], which opens the door to Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) [48], has sped up the occurrence of "era of deep learning". In 2006, Hinton et al. [46] presented the Deep Belief Network, which trains only one layer one time in an unsupervised way. That work is seen as a significant breakthrough in the deep learning field.

Two essential factors to the boost of deep learning are the vast, high-quality labeled datasets, and the strong parallel GPU computational power, which move the CPU-based training to GPU-based training. Both push forward the significant acceleration of training on deep models.

Deep Learning in Vision

Deep learning has been applied in computer vision research field for many years and facilitates the rapid development of many problems, such as human pose estimation (e.g., [19, 133]), motion tracking (e.g., [24, 23]) and object detection (e.g., [22, 104]).

The mainly used deep learning techniques involved in computer vision include:

- Perceptron is a binary classifier for supervised learning [31], which decide the input vector belongs to a specific class. As a linear classifier, the representation ability is limited. A single layer perception is even impossible to solve an XOR function. However, as a simplified model of neural network, it is the basis of Multilayer perception (MLP), Convolutional neural network (CNN), and other modern models. Definition of perception is an algorithm for binary classification with a threshold function, which maps input $\mathbf{x}$ to an output value $f(\mathbf{x})$.

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{w} * \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

- Multilayer perceptron (MLP) An MLP can be seen as an N-layer network, also can be named as "Artificial Neural Networks" (ANN). MLP overcomes the limitation of the perceptron, which requires input vector linear separable. MLP could be modeled as a collection of neurons (or units) that are connected in an acyclic directed graph and using a back propagation algorithm to learn it. The non-linear activation function like sigmoid, tanh function gives MLP the nonlinear expression ability. The most common type of MLP is a fully-connected layer in which neurons between two adjacent layers are fully pairwise connected, but neurons within a single layer share no connections.
- Convolutional neural networks (CNNs)
  LeNet starts the era of Convolutional Neural Networks [73]. Although the structure of this net only includes three convolutional neural networks, one max-pooling layer, and one fully-connected layer, this works well on the MNIST handwriting digits dataset. The convolution layer can capture

the spatial and temporal dependencies, which perform a better fitting to the image dataset. Given the different weights for the convolution kernel, Convolutional neural networks (CNN) layers can extract high-level features like edges, colors, orientations. Padding and pooling layers are the plugins of CNN layers to control the spatial size. Max-pooling layers also suppress noise.

- Generative Adversarial Networks (GANs)
  Ian Goodfelllow *et al.* first proposed generative Adversarial Networks [36]. GAN is called the "the most interesting idea in the last ten years in ML" by Yann LeCun. Compared with the discriminative algorithm, which typically works on classification problem, GAN tries to learn the distribution of input data x and generate a continuous probabilities of P(y—x). This feature indicates that GAN is suitable for generative tasks like face and sound generation, image style transfer, pose predictions. The GAN has to components: generator and discriminator. Generator tried to generate fake images that are similar to ground truth image, measured by L1 or L2 loss. The discriminator tried to distinguish whether the output image is real or fake. Both networks are trying to optimize a different loss function and confusing each other.

2.3 Global and Direct Separation

The very first approach [96] separates the direct and global components by capturing multiple images under high-frequency light patterns and compare the pixel with or without the light. Assuming neighboring points share the same direct and global components, this approach could work on a single image at the cost of lower resolution. This is also sensitive to the violation assumption, such as sharp depth or color variance.

Gu*et al.* [37] reduce the number of required images to three by simultaneously projecting multiplex sinusoid light patterns. Achar *et al.* [1] allows images captured by the human held device by compensating the small motion. By synchronizing illumination and project defocus, Gupta *et al.* [42] separate the global component before recovering the depth. Similarly, with the coaxial

camera setup, O'Toole *et al.* [145] also proposes a system to modulate both the light and the camera to probe the light transport matrix selectively. Gupta *et al.* , Reddy *et al.* [40, 110] further separate the global component into near range and far range by projecting multiple binary [40] or sinusoid patterns [110]. Recently, Subpa-asa *et al.* [124] propose a method for separating global and direct components on a single image without any resolution loss but still rely on high-frequency lighting.

Noted that the above methods treat the subsurface scattering as a whole rather than decompose it into more details [144]. Some recent efforts focus on the light transport subsurface scattering. By analyzing the side slice of the surface illuminated with high-frequency light source, Mukaigawa *et al.* [93] manage to image n-bounce subsurface scattering. Tanaka [130] decomposes the appearance of a surface seen from above into a few layers at various depths. Apart from the subsurface scattering, there are more researches on other components such as volumetric fluid [41], translucent object [92, 94].

Apart from the reflectance phenomenon discussed above, there is a group of algorithms which attempt to separate the foreground reflectance before the glass and the transmitted background after the glass. Most of these algorithms either require multiple images [28, 75, 152] or user interactions [74]. Recently, a new benchmark [137] for a single-image based method on this task has been proposed. However, the physic foundation of reflectance removal is totally different from our direct and global component separation task. The reflectance and transmission of former occurs on the glass-air interface while later is more complex than involves the volumetric scattering, inter-reflectance, subsurface scattering, and translucency *etc.*

According to the fact that SR models can often help other vision tasks, evaluating reconstruction performance by means of other tasks is another effective way for IQA. Specifically, researchers feed the original and the reconstructed HR images into a trained model and evaluate the reconstruction quality by comparing the impacts on the prediction performance. The vision tasks used for evaluation include object recognition, face recognition face alignment and parsing, etc.

To resolve the speed bottleneck of scanning based hyperspectral cameras, scanning-free devices have been proposed by using, for example, fiber bundles [88] and aperture masks with randomly [136, 33] or regularly [11] 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 [55, 171] or RGB [12, 61, 2, 69, 72, 3] image.

2.4 Hyperspectral Recovery and Analysis

2.4.1 Spectral reconstruction

Rather than making a hyperspectral imager directly, approaches for increasing the spectral resolution of a single RGB image has 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 [97], 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 [111] aimed at the same problem and proposed to learn a representative dictionary using a constrained sparse coding method. Arad and Ben-Shahar [5] focused on hyperspectral images of natural scenes and developed an RGB-to-spectrum mapping method using sparse coding. Very recently, Jia *et al.* [54] 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.* [32] 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 [5] and the shallow

network [97]. Alvarez-Gila *et al.* [4] applied a conditional generative adversarial framework to better capture spatial semantics. Xiong *et al.* [151] proposed a unified convolutional neural network for hyperspectral reconstruction from RGB images and compressively sensed measurements. Compared with pixel-wise operations [97, 111, 5], the imagewise operations in deep learning based methods [32, 4, 151] are more likely to incorporate spatial consistency in the reconstruction.

2.4.2 Spectral Classification and Segmentation

Recognizing materials, objects, land cover classes in the hyperspectral image can be viewed as a classification or segmentation task, which is widely studied for a long time. The used techniques can be divided into two parts: traditional methods and deep learning based methods. Concerning that data dimensions of hyperspectral is always too large and information redundant, the majority of existing traditional methods consist of feature extraction, e.g., PCA and kernel-based classifier and support vector machines (SVM) [8, 131, 29, 60, 35]. To integrate spatial context into consideration, probabilistic models like Markov random fields (MRF) are used [131, 35]. Meysam *et al.* [29] use morphological information to extract spatial relations in original data. Compared to conventional framework, deep learning methods [77] can lean the hierarchy of features automatically and shown promising results for hyperspectral classification. Wei Hu *et al.* [57] proposed 1-D deep convolution networks with a full connection layer to have better results than traditional methods.

Makantasis *et al.* [85] proposed a unified framework, which combines spectral and spatial information in 2D. Roy *et al.* [113] proposed a method using 3D and 2D convolution for hyperspectral image classification. Zhu *et al.* [170, 165] proposed a generative adversarial network for this task.

2.4.3 Spectral Response Functions Optimization

All the research above simulated RGB images using typical response functions from commercial RGB cameras. Very recently, Arad and Ben-Shahar [6] recognized the accuracy of the 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 [18], 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 [18] 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.

Chapter 3

Physical-Based Constraints in Network Loss for Global Direct Separation

3.1 Overview

In this chapter, we show how to separation global/direct components, which is one of the optical properties, by adding physical based constraint in the final loss of Generative Adversarial Network. Generative adversarial network (GAN) [36] has achieved impression result in image generation such as image reconstruction [4], biological image synthesis [101], image style transfer [52], shadow detection [98], and future video frame prediction [79]. In this paper, we propose a novel GAN based network architecture for recovering the direct and global component from a single image. Inspired by cycleGAN [168], as illustrated in Fig 3.1, our network introduces an inverse operation that imposes prior physical knowledge to enforce a physically plausible separation.

Instead of treating neural networks as 'black box', more and more research

[123, 30, 59, 99] attempted to embed domain knowledge in deep learning models. This would not only help to compress ample parameter searching space but also provide meaningful results. For example, by assuming an object is moving at a constant velocity, Stewart *et al.* [123] proposed a method to supervise a convolutional neural network to detect and track objects without any label. Video frames could be predicted by forcing pixels with physics dynamics [30]. Another example is to embed relationships between density, depth, and temperature of the lake with known physical equations in a physics-guided neural network [59].

Since our mapping from direct/global components to input image is defined by physical Equation 3.3, we replace the G_x of standard cycleGAN [168] that maps domain Y to domain X with a linear mapping layer as shown in Fig 3.1. This architecture allows us to reduce complexity needs and get a realistic solution for components separation. Our architecture is shown in Fig 3.1 for single-image components separation.

The input is a single RGB image X and output Y is a concatenation of global component Y_1 and direct component Y_2: $Y = Y_1 \| Y_2$. Let the G, D denote a generator(G) and a discriminator(D) respectively for the sake of simplicity. For our specific problem, we introduce a linear mapping layer L and encourages $L(G(x)) \simeq x$ to force the generated global and direct component to follow the physics model of Eq3.3.

3.1.1 Network Architecture

Our network module is formed as follows: 2D convolution-BatchNorm-Relu. The generator takes scene image of size 256×256×3 as input and finally produces the corresponding global and direct images of size 256×256×6. 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 paddings 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

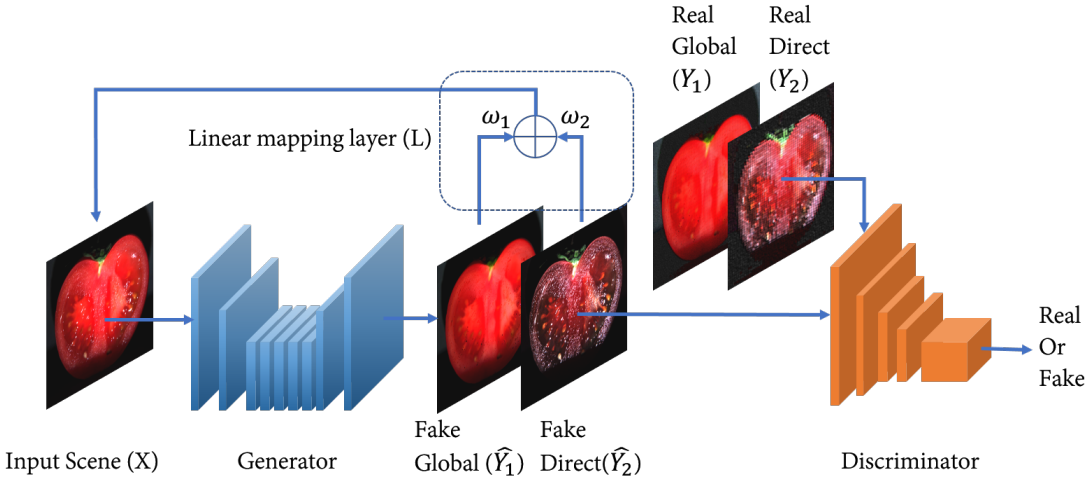


Figure 3.1 Our models contains three functions: a generator (G), a linear mapping layer (L), a discriminator (D). For a scene X, we concatenate global and direct component image together as Y. We add a linear mapping layer to regularise prediction Y to the physical constraint [95]: $\hat{X} = w1 * \hat{Y}_1 + w2 * \hat{Y}_2$ and add $Loss_L = L1(X, \hat{X})$ to the final loss function.

layer is replaced by the deconvolution layer, which upsamples the input by a factor of 2. A dropout layer with 50% dropout rate is added after each block. The generator architecture is composed as C64-C128-C256-C512-C512-C512-C512-CD512-CD512-CD512-CD256-CD128-CD64-CD6.

Compared to a standard U-net, we modify its final layer from 3 channels to 6 channels. The discriminator takes 256×256×9 as an input image, which is the concatenation of generator input and output. The final layer of discriminator adopts active sigmoid function. The structure is composed as: C9-C64-C128-C256-C512-C1.

3.1.2 Network Design for Components Separation

The objective of traditional GAN is defined as:

$$\begin{aligned} Loss_{GAN}(G,D) &= E_{Y\sim P_{data}(X,Y)}[logD(X,Y)] \\ +E_{X\sim P_{data}(X),Z\sim P_Z(Z)}&[log(1-D(X,G(X,Z)))], \end{aligned} \tag{3.1}$$

where Z is a noise vector.

Pix2pix, a generic GAN method [52] found that mixing $Loss_{GAN}(G,D)$ with generator loss $Loss_{L1}$ would be beneficial as L1 produces less blurring results:

$$Loss_{L1}(G) = E_{X,Y\sim P_{data}(X,Y),Z\sim P_Z(Z)}[||Y-G(X,Z)||_1]. \tag{3.2}$$

The generator loss requires G not only to fool discriminators D but also to provide more traditional loss, in order to get similar images compared to ground-truth images. In the current setting, we find L1 distance would deliver more clear result.

However, $Loss_{GAN}(G,D)$ and $Loss_{L1}(G)$ does not consider the physics relation between global image Y_1 and direct image Y_2. According to [95], the input image X is under the linear relation:

$$X = w_1Y_1 + w_2Y_2 \tag{3.3}$$

where w_1, w_2 is the weight of two components. Therefore, we add this prior knowledge by defining a linear mapping layer $L(Y) = \sum_{n=1}^{2} w_n Y_n$. We propose the inverse cost to impose the physics regulation:

$$Loss_I(G, L) = E_{X,Y \sim P_{data}(X,Y), z \sim P_Z(Z)}[||X - L(G(X, Z))||_1] \tag{3.4}$$

Finally, the objective function of network is defined as:

$$Loss = \arg\min_G \max_D Loss_{GAN}(G, D) + \lambda_1 Loss_{L1}(G) + \lambda_2 Loss_I(G, L) \tag{3.5}$$

Where λ_1 controls the relative importance of reconstruction loss while λ_2 determines the weight of inverse loss. In this paper, we set the λ_1 and λ_2 to 100.

3.2 Benchmark Dataset

For this research, we collect a dataset of 100 controlled indoor scenes along with their direct and global components. the captured scenarios cover a wide range of daily-life objects including plastic, food, and sweets (fresh fruits, vegetables, bread), synthetic fabrics and wooden object *etc.* There are 13 translucent items among 100 items, including common objects such as ceramic, jade, glass, various minerals, and candy *etc.* Each scene is of a triplet of images: 1. Scene image (Fig 1.2.(2)), 2. Direct component (Fig 1.2.(3)) and 3. Global component (Fig 1.2.(4)). In all, our dataset contains $3 \times 100 = 300$ images.

3.2.1 Data Collection

The data capture setup involves one projector, one camera, and a scene of one or multiple objects. For each scene, we collect the data in two steps: 1, We at first capture the scene image using a white background projected by the projector. 2. Then we measure the direct and global components in the way of [95].

As Nayar *et al.* [95] suggested, each scene was lit using a checkerboard pattern projected by the projector when calculating the global and direct com-

ponent. A checker pattern size of 8 x 8 pixels was used for the experiment with a shift of 2 pixels 8 times in each of the two dimensions.

We denote L^+ as the image under a high-frequency illumination that half of the image is lit. L^- is the image under the complementary illumination. For any pixel i lit in L^+ and deactivated in L^-, as proved in [95], it should follow $L^+[i] = L_d[i] + (1+b)\frac{L_g[i]}{2}$ and $L^-[i] = bL_d[i] + (1+b)\frac{L_g[i]}{2}$, where $L_d[i]$ is the direct component and $L_g[i]$ is the global one. b represents the deactivated source element brightness on the pixel i. In theory, two images are sufficient to calculate the separation if digital projector is able to project an ideal high-frequency pattern. In practise [96, 37, 1], capturing more images would significantly relieve this issue. For each pixel i, we use the minimum and maximum measured brightness $L_{min}[i]$, $L_{max}[i]$ instead of $L^-[c,i]$, $L^+[c,i]$ to compute the separated components. For each scene, we captured 64 images to ensure the reliable separation of direct and global components.

Throughout the data collection, two projectors and two cameras were utilized to simulate various capture settings. For example, we use BenQ PJ projector for half of the scene and DLP Light Commander for the remainder. Similarly, the Nikon 40S camera was used to capture half of the scene regardless of the selection of the projector, while the rest were captured by Grasshopper 3. To maximize diversity, we change the position of projectors, the camera, and the target objects for each scene.

3.3 Evaluation

3.3.1 Experiment Setting

In this evaluation, the whole dataset was randomly split into training/testing set. The training set includes 80 scenes while the testing set takes the rest 20. During the training, we at first resize each image into 1500*1500. Then, it was randomly cropped into 256*256 patches with affine transform, rotation, and shearing as data augmentation. The rotation range is -20 degree to 20 degree

while the shearing range is -10 degree to 10 degree.

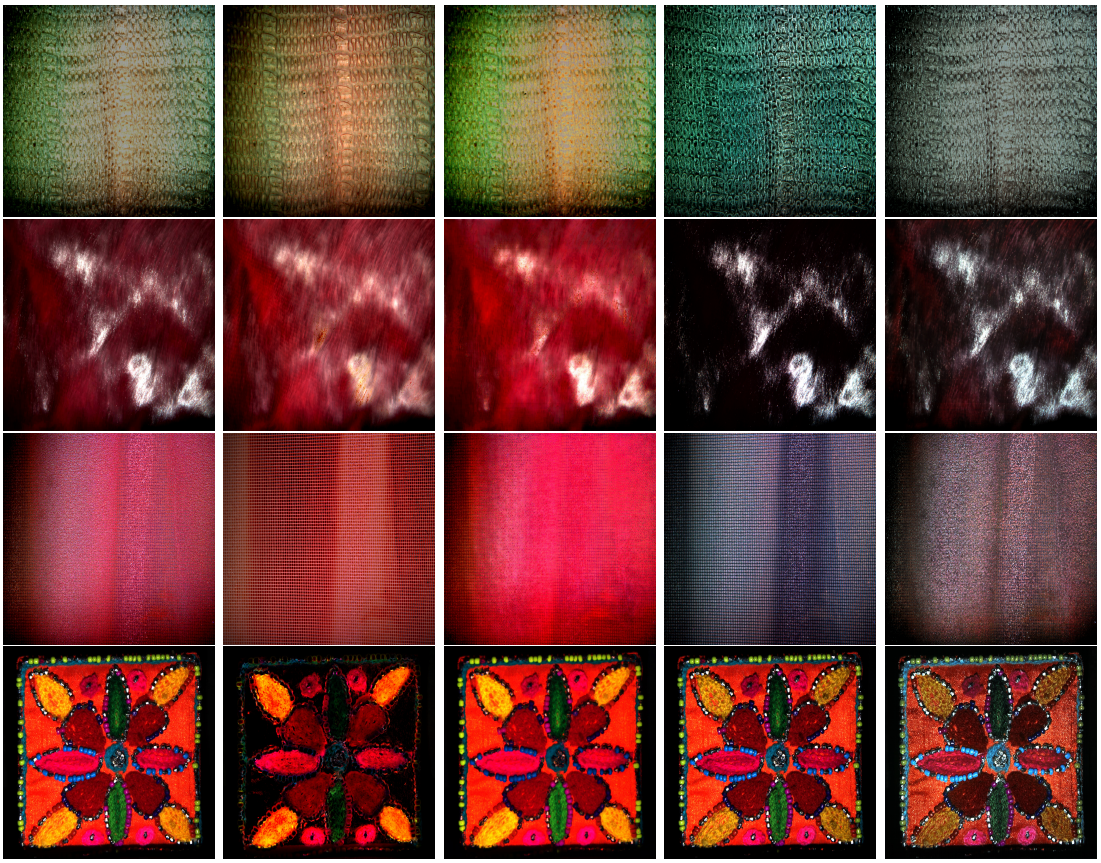
During the testing process, we at first crop the input scene image into overlapping 256*256 patches before feeding into the trained generator. The overlapping part of each patch is 200 pixels in horizontal and vertical direction separately. When knitting the patches back into images, we calculate the pixel value of the overlapping part by taking an average of corresponded image pieces.

3.3.2 Visual Quality Evaluation

We compare our separated components with the groundtruth in Fig 3.2, 3.3. The comparison shows that our method successfully separates most details of components from a single image.

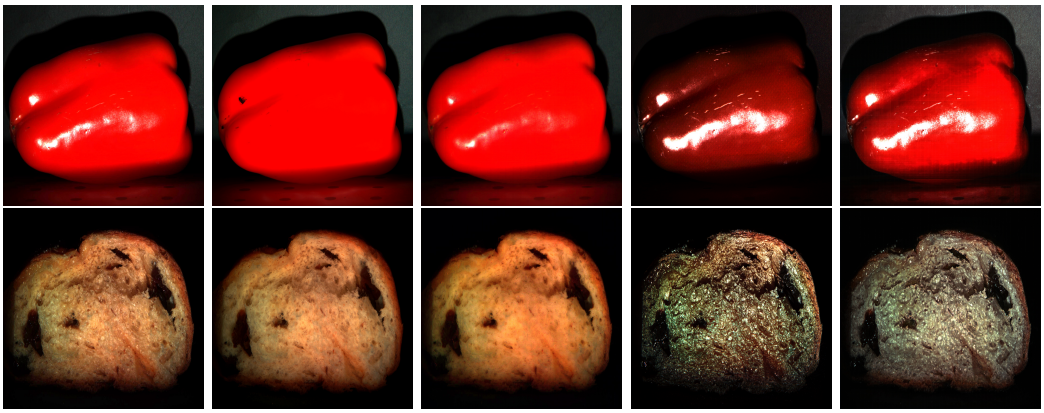
We also compare our result with the baseline pix2pix [52] that works without physical constraints. As shown in Fig 3.4, the baseline method pix2pix2 often delivers striped distortion or blurry results, especially at the object boundary and background area. However, with our proposed network, these artifacts are avoided, as shown in Fig 3.4.

With the proposed dataset, our method can work on the general images without a strict capture setting requirement. As illustrated in Fig 3.5, we also apply our trained model on the images in public dataset such as CAVE [153]. The images in the CAVE are captured under a neutral daylight illuminant (CIE Standard Illuminant D65). The image size are of 512*512 resolution, and it was cropped into 256*256 pieces to feed into our pre-trained network. The output is tilled with 100 pixels to fit the original input size. Fig 3.5 shows our method could achieve reasonable performance even for images captured under different setting from that of the training set. We also tested our method in improving the computer vision application, such as shape from shading (SFS) [108]. One example is given in Fig. 3.5. Our global component SFS results are more in accordance with real geometry by removing the specular part in the direct component.



(a) Input | **(b)** Global Groundtruth | **(c)** Predicted Global | **(d)** Direct Groundtruth | **(e)** Predicted Direct

Figure 3.2 The exemplar components separation of real scattering materials..



(a) Scene **(b)** Global Groundtruth **(c)** Predicted Global **(d)** Direct Groundtruth **(e)** Predicted Direct

Figure 3.3 The exemplar components separation on food.

3.3.3 Quantitative Result

In this analysis, we adopted three metrics when comparing recovered global and direct component image with provided ground-truth: structural similarity (SSIM) [139], structure index (SI) [127], Inception Score (IS) [115]. We did not use the evaluation metrics such as L2 (RMSE), L1, or PSNR because they prefer a blurring result rather than the one with highly accurate textures [53, 106, 158]. Mean pixel-wise Euclidean distance is minimized when result averages all plausible outputs. Therefore, we select SI, SSIM, Inception score, which are more widely used for quality assessment of generative model application such as [38, 100, 7, 86].

SSIM [139] evaluates the human visual perception of luminance, contrast, and structure. However, this does not consider correlations between pixels, which carries structure information of an object in a scene. Thus, we also suggest structure index (SI) [127] which only focus on structure relation between recovered image I and groundtruth I^*. The SI is defined as: $SI = \frac{2\sigma_{I,I^*}+c}{\sigma_I^2+\sigma_{I^*}^2+c}$, where σ_I, σ_{I^*} stands for the variance of I, I^*, σ_{I,I^*} stands for the covariance of I

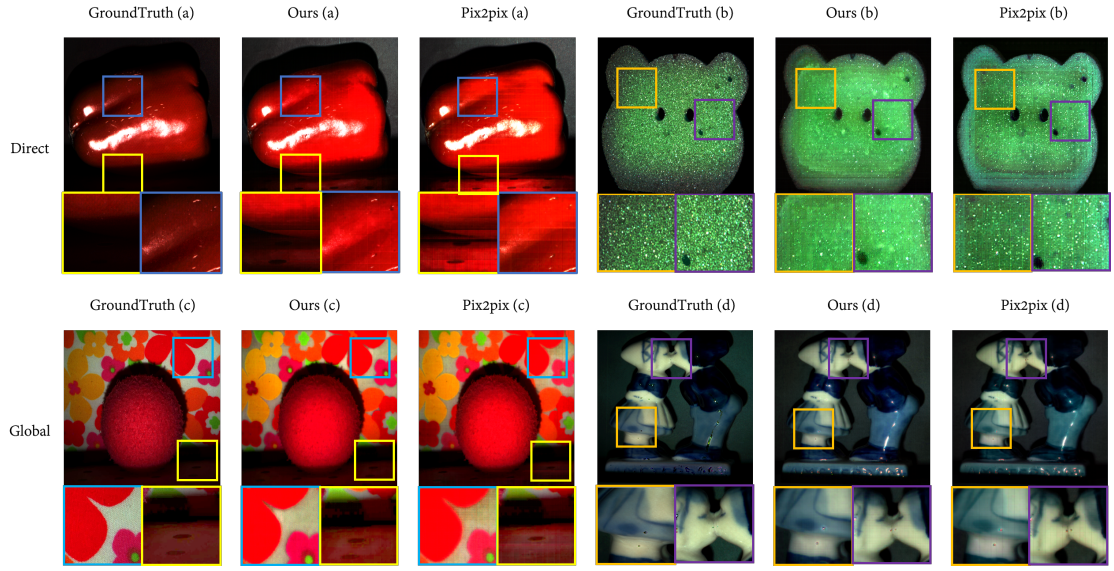


Figure 3.4 The first row shows the comparision of our method and pix2pix for direct generation, the second row shows the comparision on global generation. Note that pix2pix sometimes provides striped distortion as shown in (a,b,d) and more blurry result as shown in (c). We adjust the brightness and contrast for better visualization.

and I^*. c is a constant. The higher SSIM and SI score indicates less structure distortion and better quality.

The Inception Score (IS) is a metric for evaluating the quality of image generative models [115], which used an Inception v3 network [129] pretrained in ImageNet [114]. IS was shown to correlate well with the human judgment of realism [115]. A high score means a better result.

We report the baseline, which is the result of our proposed method compared with pix2pix [52] of SSIM, SI, IS on the test set of this dataset as shown in Table 3.3. We also report quantitative result for individual image in Table 3.1,3.2.

On all of the three metrics, our methods outperform pix2pix. This is in accordance with our observation in Fig 3.4 that the components separated by our method are with less structure distortion and of more natural looking.

number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SSIM(0.)	7.9	5.0	**9.0**	8.2	9.3	**9.5**	**8.8**	**9.2**	**9.4**	**9.3**	**9.3**	**8.5**	**8.7**	**7.8**	**9.0**
SSIM*(0.)	8.1	5.3	8.8	7.9	9.4	9.3	8.3	9.1	9.2	9.1	9.1	8.0	8.5	7.7	8.4
SI(0.)	**9.6**	**9.2**	9.8	**9.0**	9.8	9.9	9.5	9.4	9.9	9.8	9.7	9.6	**9.7**	9.5	**9.7**
SI*(0.)	9.5	9.1	9.9	8.9	9.8	9.9	9.5	9.5	9.9	9.9	9.7	9.6	9.6	9.5	9.5

Table 3.1 SSIM,SI for direct component reconstruction. Superscript * stands for the result of pix2pix baseline.

number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SSIM(0.)	**9.0**	**8.2**	**5.7**	8.4	**9.1**	9.3	5.5	9.6	8.5	**9.1**	**6.6**	9.4	7.5	**5.3**	**8.1**
SSIM*(0.)	9.1	8.2	5.5	8.7	9.0	9.3	5.5	9.7	8.8	8.9	6.3	9.5	7.5	5.2	8.0
SI(0.)	9.8	**9.2**	**9.6**	9.5	9.8	9.9	6.0	9.9	9.4	9.6	**8.2**	9.9	9.0	5.7	**9.4**
SI*(0.)	9.8	9.1	9.5	9.6	9.8	9.9	6.0	9.9	9.4	9.6	8.1	9.9	9.0	5.7	9.2

Table 3.2 SSIM,SI for global component reconstruction. Superscript * stands for the result of pix2pix baseline.

Method	Ours	Pix2pix[52]
SSIM	0.823	0.812
SI	0.924	0.922
IS	2.24	2.21

Table 3.3 Quantitative results of global and direct separation on our dataset.

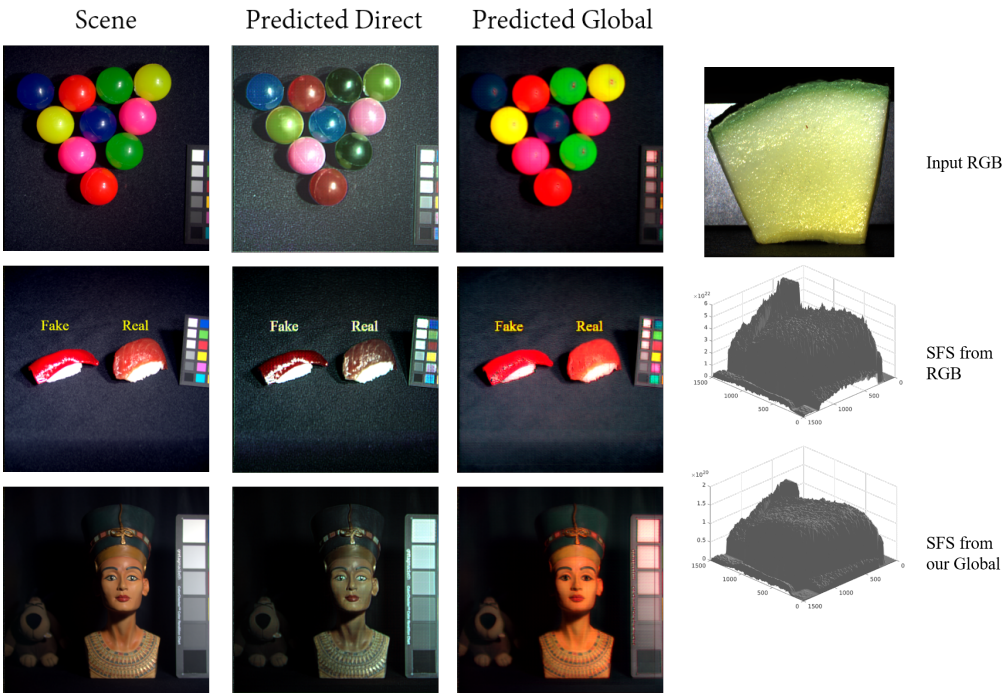


Figure 3.5 Left-hand part: We test our method to separate global and direct components in images from CAVE dataset [153]. From left to right: input single RGB scene under neutral illumination, predicted direct component, predicted global component. The right-hand part is shape from shading result compared with baseline.

3.4 Image Editing by Enhancing Direct and Global Components

Nayar *et al.* [96] showed that linearly mixing direct and global components with different weights is effective for image editing. In this paper, we further explore physically plausible material editing by manipulating our direct/global separation results with different linear weights.

Fig 3.6 shows image editing by manipulating the weights of direct and global components separated by our approach from the single input image. We

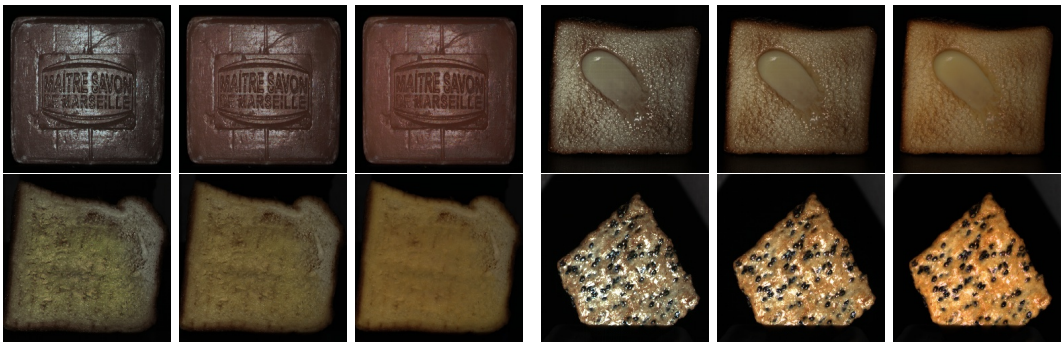
can see that object impression changes according to the weight. For instance, we show a hard-to-soft transition in Fig.3.6 (a): as we increase the weight of the global components, the objects look softer. Metallic / non-metallic transition in (b) where objects look more metallic with higher weights of direct components. Interestingly, the visual freshness of food can also be controlled with different weights in Fig.3.6(c). The proportion of specularity due to its subsurface scattering seems to be essential for us to recognize food freshness.

3.5 Conclusion

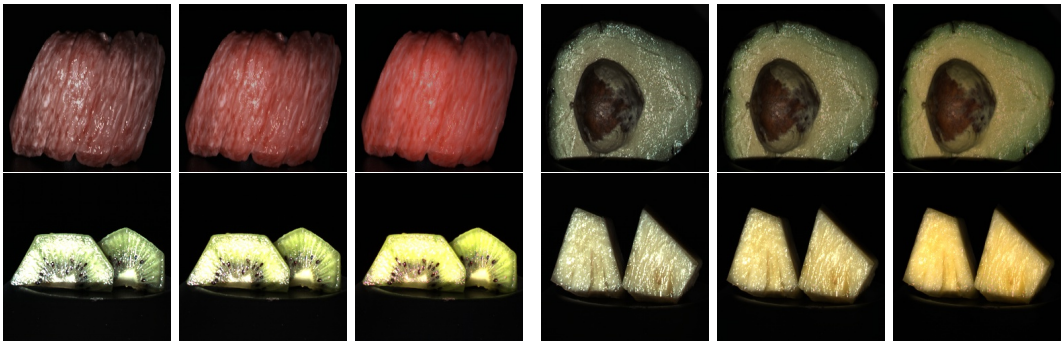
In this chapter, we propose the first method to separate direct and global components from a single image without hardware constraints. This model embeds substantial prior knowledge into the GAN based network to achieve single-image components separation. To train and evaluate this model, we also present the first dataset, which comprises of 100 scenes with their groundtruth direct and global components. Our method has been shown to work successfully on our own testing set and general images from the public dataset. Finally, we demonstrate how the separated components could be used for realistic image editing.

Unlike previous research relies on specific capture setting, we focus on extracting components from a single image with a simple capture request. Our dataset contains various indoor scenes with different capturing settings. Based on this dataset, we also propose the first method to estimate the direct and global components from a single image. We also illustrate some potential application of this separation in modifying the image in a physically plausible way.

(a) Hard to Soft



(b) Metallic to Non-Metallic



(c) Freshness of food

Figure 3.6 Image editing with direct and global enhancement. The appearance of objects changes according to different linear mixing applied to direct and global components. the weights (direct, global) used for (a) and (b) are (0.9, 0.1); (0.6, 0.4); (0.3, 0.7) from left to right. In (c) the weight used are (0.8, 0.2); (0.5, 0.5); (0.2, 0.8) from left to right.

Chapter 4

Physics-Based Design in Network Architecture for Hyperspectral Recovery

4.1 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 based on physical process of camera imaging onto it to learn the filter response functions as well.

4.1.1 Spectral Reconstruction Network

Noted that arbitrary end-to-end network could be used for our spectral reconstruction. Here, for the sake of generality, we compare two networks spectral

reconstruction network: Unet and HSCNN[118].

Unet as Backbone Network

The well-known U-net [112] architecture, which has been widely used for image-to-image translation applications, such as pix2pix [52], CycGAN [169], Semantic Segmentation [117] and hyperspectral reconstruction [4].

In recent years, deep convolutional neural network (CNN) has shown its power in the computer vision field. Although CNN has been proposed in 1989 [73], but due to the size of datasets and networks, the success is limited. However, due to the development of the Graphics Processing Unit (GPU), academics can build broader and deeper networks recently. When it comes to deep learning, the pioneering work was done by Krizhevsky *et al.* [67], with eight layers and training on the ImageNet dataset with 1000 classes. After that, VGG net [120] has been proposed and finally won the 2014 ImageNet Challenge.

The previous networks have outperformed the state of art methods in image classification problems. However, in many visual tasks like bioimage segmentation, each pixel should be assigned a label. The is the limitation of traditional deep learning image segmentation methods such as Fully Convolutional Network (FCN) [117]. Base on FCN, U-net [112] was proposed to give more precious segmentation. The critical contribution of U-net is to propose an elegant structure: skip-connections, which will allow information flow to be copied and concatenated in deeper layer. This method will avoid information loss during down-sampling operation, such as max-pooling. The idea of skip-connection has also been applied to image classification, such as Densely Connected Convolutional Networks (DCCN) [120], the difference is, every layer is connected to other layer to ensure maximum information flow. Also, the idea of skip connection has been used in Highway networks [122], ResNet [44], and so on.

Many previous encoder-decoder networks [105] pass the input through a series of down-sampling operations, such as max-pooling, until a bottleneck layer before reversing the process. Passing the information through these layers

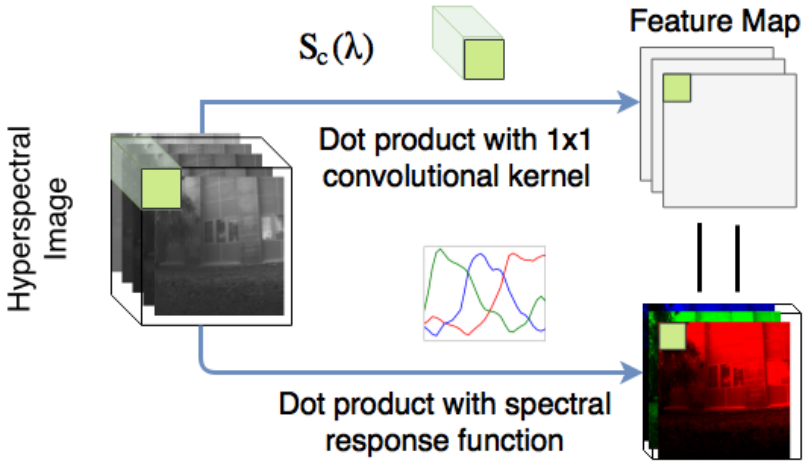


Figure 4.1 Similarity between the 1×1 convolution and the filter spectral response.

would inevitably sacrifice much of low-level details in the high-resolution input grid. Therefore, in the image-to-image application, 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 [50, 44, 27]. This structure can mitigate the issue with vanishing/exploding gradients when the model is "very deep" [44]. What is more, U-net also works well on small-sized training datasets [117]. As our dataset is limited in size, 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 paddings to edges. The α parameter in the leakyReLU layer is set to 0.2. *CDk* denotes the same block as *Ck*, except that the deconvolution layer replaces the convolution 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).

HSCNN as backbone net

Inspired by a previous paper that won the NTIRE2018 spectral reconstruction challenge [118], we use HSCNN comprised of residual blocks as a backbone network. Residual blocks, also called Resnet, first proposed by He et.al [44], this architecture makes deep learning network easier to train and also have great performance in image classification Inception-v4 net [128], image super-resolution [146], hyperspectral classification [166], manifold learning [78] etc. Based on Resnet, Huang [50] proposed a densely connected convolutional network (DenseNet) and get a competitive result on image recognition benchmark datasets. HSCNN is based on DenseNet and ResNet to produce state-of-art results and increased computational complexity.

4.1.2 Add a Physical Based Inverse Loss for Rgb to Spectrum Network

In order to keep RGB images consistent to the groundtruth, inspired by similar work [13], which makes predicted RGB image from a grayscale input image. We applied the same optical-based loss as global direct separation network, as a weighted sum of output (hyperspectral multiply existing camera). Suppose we have three color matching functions (R,G,B) for each channel, which will be write as $f_r(\lambda)$, $f_g(\lambda)$, and $f_b(\lambda)$ for any output hyperspectral image $\mathbf{g} = \{g(\lambda_1), g(\lambda_2), ..., g(\lambda_i)\}$ and input rgb image $\mathbf{i} = \{i_1, i_2, i_3\}$, the corresponding

rgb image will be:

$$rgb(\mathbf{g}) = \{\sum_{j=1}^{i} f_r(\lambda_j)g(\lambda_j), \sum_{j=1}^{i} f_g(\lambda_j)g(\lambda_j), \sum_{j=1}^{i} f_b(\lambda_j)g(\lambda_j)\} \tag{4.1}$$

To keep the rgb subspace of generated hyperspectral image consistent with input rgb, we need to add an additional L1 loss as a controller to make sure $rgb(\mathbf{g}) \approx \mathbf{i}$:

$$Loss_{L1}(G) = \beta(||rgb(\mathbf{g}) - \mathbf{i}||) \tag{4.2}$$

With this additional physical-based loss, the network encourages the input to be similar to a weighted sum of output.

4.1.3 Filter Spectral Response Design

one key novelty of this thesis 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.

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

$$I_c(x, y) = \int_{\lambda} S_c(\lambda)L(x, y, \lambda)d\lambda, \tag{4.3}$$

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

$$I_c(x, y) = \sum_{n=1}^{N} S_c(\lambda_n)L(x, y, \lambda_n), \tag{4.4}$$

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.

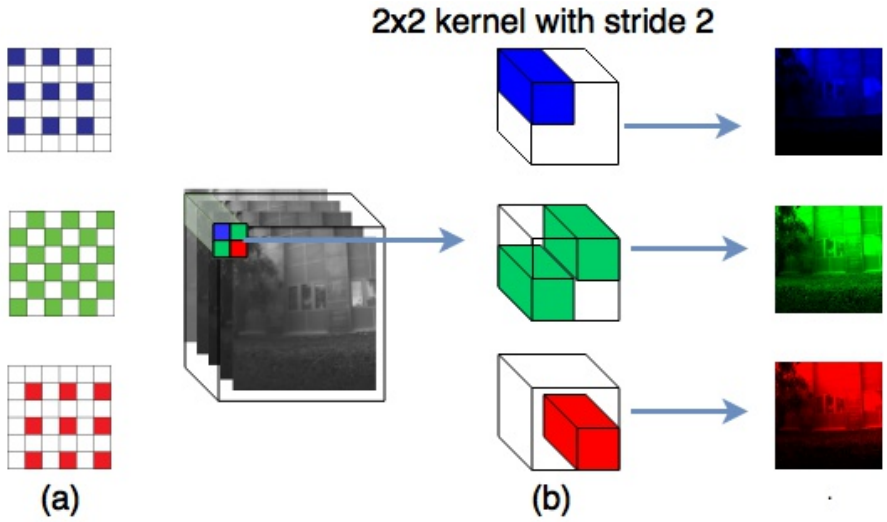


Figure 4.2 The typical Bayer filter array setup and our special convolution kernel for the Bayer-style 2×2 filter array design.

An interesting observation is that Eq. 4.4 is identical to the convolution operation of a 1x1 convolution kernel in forwarding propagation. By regarding the filter spectral response function, $\mathbf{S_c}$ as the weight of 1x1 convolution kernel, as shown in Fig. 4.1, 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. 3.1, 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.

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 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, as illustrated in Fig. 3.1, which is suitable for constructing multi-channel camera prototypes.

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

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. 4.2(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. 4.2(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.

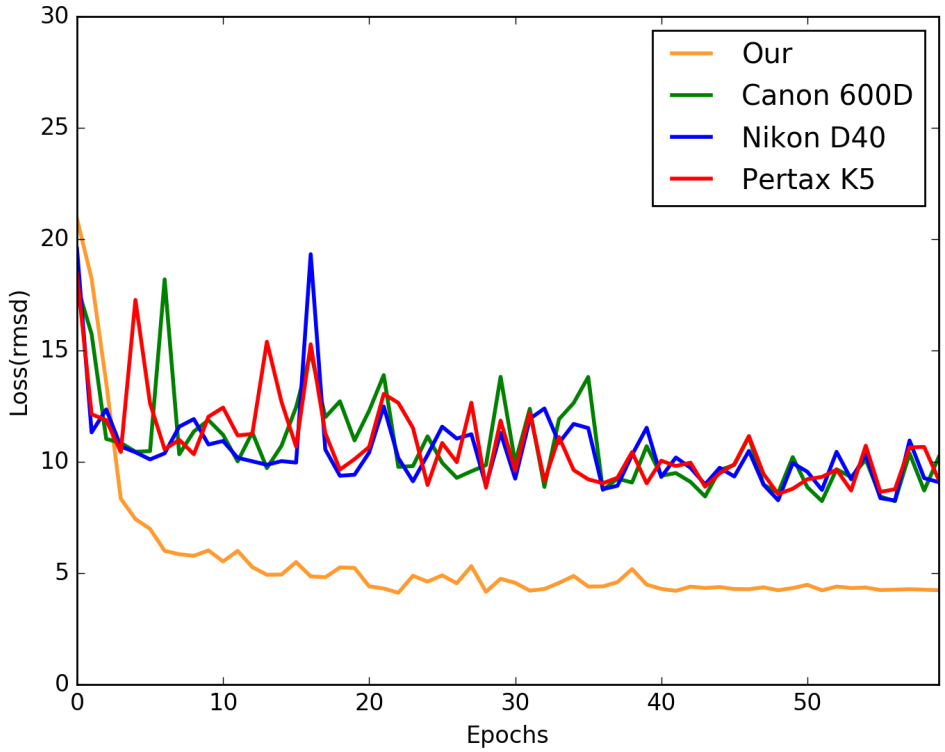


Figure 4.3 RMSE of each epoch of our designed and existing spectral response function on the CAVE dataset [154].

Non-destructive Filter Design

As hardware implement of designing CCD is costly and we want to investigate whether it is possible by simply depose a monochrome filter in front of camera lens to improve the hyperspectral reconstruction result.

In the experiment setting, if an input hyperspectral image is $X_{H,W,C}$, stands for height, width, channels separately, the filter sensitivity $S_{1,C}$ is learned during training, the rest network is $N(X)$, the final objective of autoencoder is $L = ||N(X \circ S) - X||_2$, where $\circ$ stands for hadamard product, which is element-wise multiplication.

L2 regularization can be derived from this assumption: layer weights obey Gaussian prior distribution. However, it does not ensure smoothness at any time; it only forces weight to choose smaller value and approach zero. In image processing, Gaussian blur is widely used as a low pass filter to smooth image. So we try to use 1D convolution with a fixed kernel like discrete Gaussian kernel. In our experiment setting, window length of that kernel is set to 3, and stand deviation is 1.

The framework function is shown in Fig. 4.4

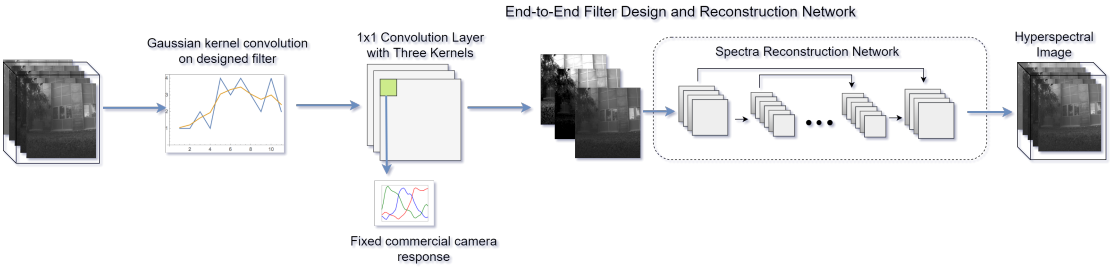


Figure 4.4 System framework for joint designing filter illumination response.

4.1.4 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.

Smooth Constraint Using L2 Norm Regulizer

There are various regularizers in the 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., nonnegative regularizer, and L1 norm.

To achieve nonnegative responses, we enforce a nonnegative regularizer on the kernel in our filter design convolution layer, such that $S_c(\lambda) \geqslant 0$. 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 $\eta\sqrt{\sum_{n=1}^{N}\left(S_c(\lambda_n)\right)^2}$, where η controls the smoothness. Throughout the experiment, η is set to 0.02.

In the subset of CAVE dataset, including real and fake objects, due to the limitation of dataset size, l2 norm can not produce a smooth spectral response. Thus, we propose a Fourier based method for a more strong constraint. We also implemented them into a network, as shown in subsection 5.2.2.

4.2 Reconstruction 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 [154, 15].

4.2.1 Training Data and Parameter Setting

The CAVE [154] dataset is a popular indoor hyper-spectral 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 [15] is a real-world hyperspectral dataset, including both outdoor and indoor scenarios. The image data are 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".

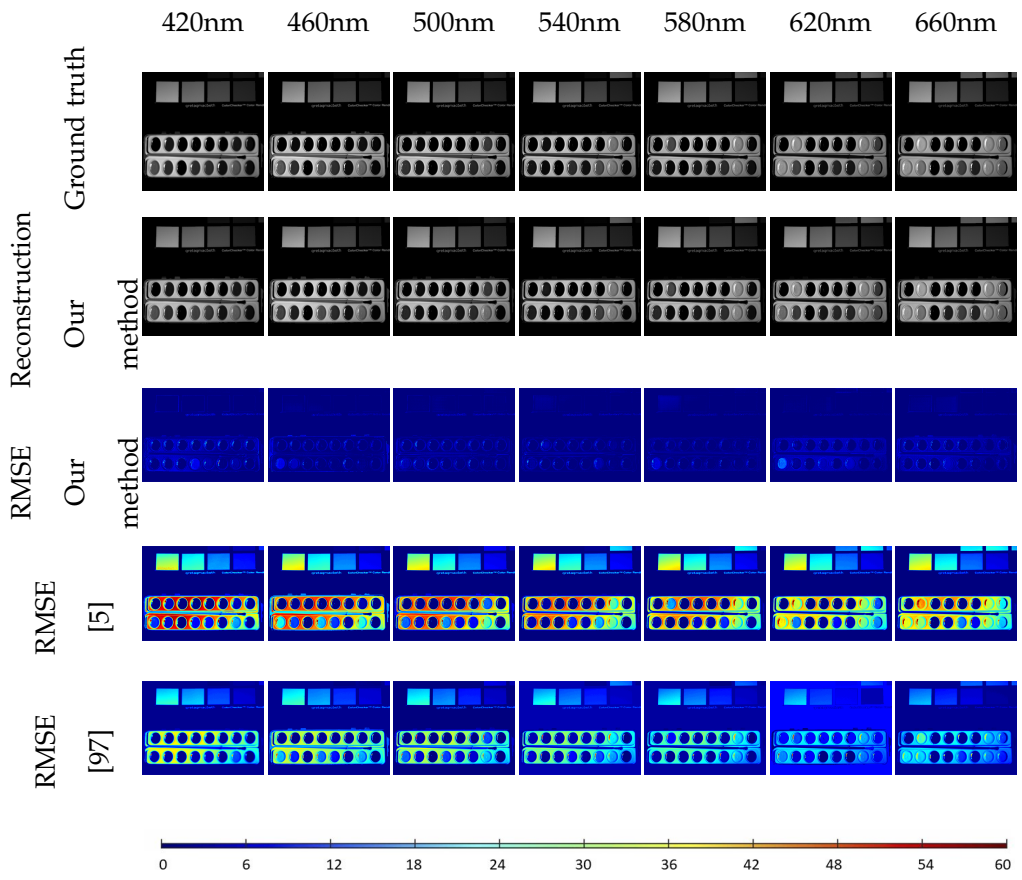


Figure 4.5 Sample Results from the CAVE Database [154]

In the training stage, we apply random jitter by randomly cropping 256×256 input patches from the training images. We trained our algorithm with a batch size of 2 and 50 iterations for each epoch. We trained the network with the Adam optimizer [66] 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 a mean 0 and a standard deviation 0.02.

We run our proposed algorithms on an NVIDIA GTX 1080 GPU. Our server is equipped with an Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz and 128GB of

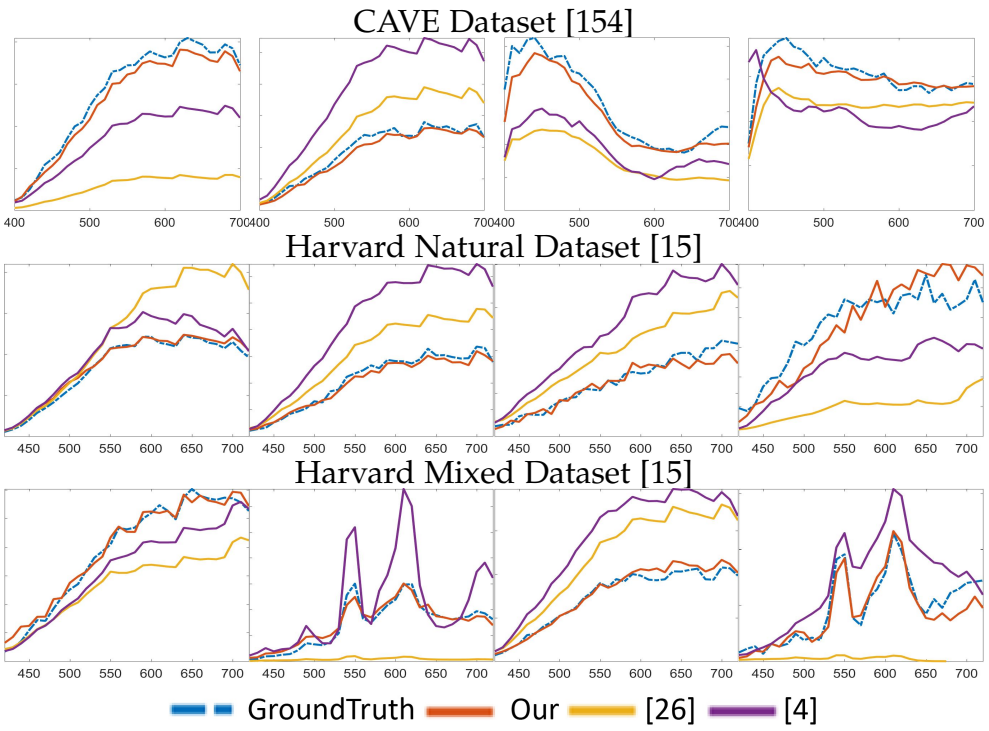


Figure 4.6 The reconstructed spectra samples for randomly selected pixels in the CAVE and Harvard Natural and Mixed datasets [154, 15]. Each row corresponds to its respective dataset.

memory. The training time for the CAVE [154], Harvard Natural and Mixed [15] 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 $[0, 255]$.

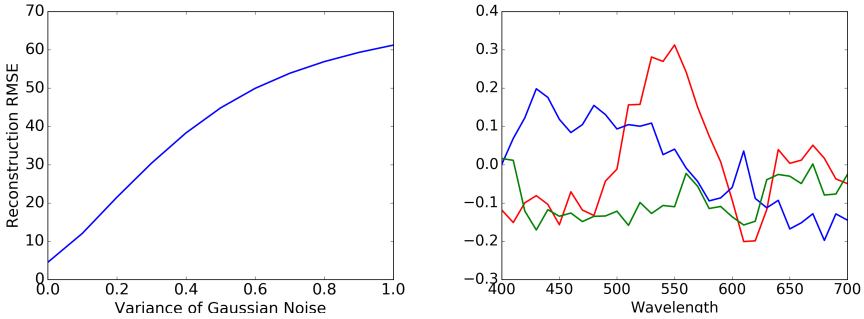


Figure 4.7 RMSE vs. Noise Level and Non-physical solutions for learned responses.

4.2.2 Results on 3 Channel Multiple-Chip Setting

At first, we evaluate the multi-chip setup described in Sec. 4.1.3. In this section, we evaluate the performance of the multi-chip setup with three sensors. The optimal spectral response function for the CAVE dataset [154] is given in Fig. 3.1.

Influence of Noise and Constraint

Specifically, we also simulated sensor noise by adding Gaussian noise to the data from the CAVE dataset and report the RMSE of the hyperspectral reconstruction, as shown in Fig . 4.7. The non-physical solutions for responses in Fig. 4.8. As expected, the non-physical responses are not smooth, and some portions are even negative, so they cannot be directly realized in film filters.

Influence of Inverse Camera Loss

We add an extra l1 loss as shown in Sec. 4.1.2, we evaluated the influence of β in equation 4.2. The evaluation metric is normalized L1 loss (Mean Absolute Error), as shown in Tab. 4.1

We exam the network performance with different β settings of the physical

Table 4.1 The β in physical based inverse loss in equation 4.2 vs. MAE, the smaller is better. $\beta = 0$ stands for a normal unidirectional inference with U-net network loss. $\beta = 1.0$ achieve the best evaluation result.

β	MAE
0.0	0.786561
0.2	0.757819
0.4	0.735643
0.6	0.782672
0.8	0.711089
1.0	**0.693925**

inverse loss term. Specifically, when we fix the parameter $\beta = 1.0$, the performance is significantly improved. It suggests a 12% improvement of MSE. Additionally, the MAS vs. β is not monotonically decreasing, when $\beta = 0.6$, there is no apparent difference between none physical-based inverse loss. We conclude that physical-based inverse loss for mapping output hyperspectral and compress to RGB keep the consistency with input RGB and select the weight for the loss term in the final loss is also essential.

The average and variance of the RMSE are shown in Table 4.2, which was compared with three baseline methods: [5], [97] and [54]. The RGB inputs of three baseline methods are generated from the spectral response function of Canon 600D. This table shows that the RMSE of our method outperforms the alternative methods in spectral reconstruction in all three datasets.

The learned spectral response function is shown in Fig4.9.

We also demonstrate the spatial consistency of the recovered hyperspectral images from CAVE datasets in Fig. 4.5, which shows images at seven different wavelengths.

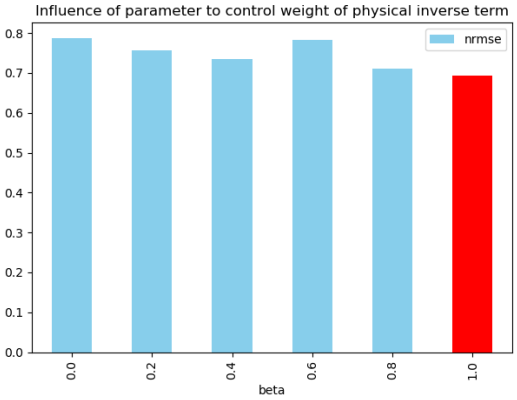


Figure 4.8 MAE of HSI reconstruction over six possible β in final loss term. Performance is the best when $\beta = 1.0$.

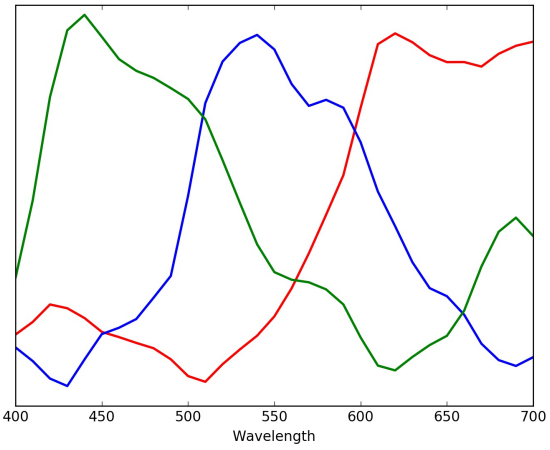


Figure 4.9 Learned optimal spectral response function trained on CAVE dataset[154]. Y axis stands for the amplitude.

Table 4.2 Average and Variance of RMSE of reconstruction on the hyperspectral databases [154, 15, 54].

	CAVE[154]	Harvard Natural[15]	Mixed[15]
Our	4.48 ± 2.97	7.57 ± 4.59	8.88 ± 4.25
[5]	8.84 ± 7.23	14.89 ± 13.23	9.74 ± 7.45
[97]	14.91 ± 11.09	9.06 ± 9.69	15.61 ± 8.76
[54]	7.92 ± 3.33	8.72 ± 7.40	9.50 ± 6.32

We also represent the recovered spectra for random points from three datasets in Fig. 4.6, which shows that 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. 4.3.

As shown in Fig. 4.3, 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 the supplementary material. Specifically, we first report performance in the case of added Gaussian noise. Then we report the response function trained without physical constraints. We also simulate sensor noise by the data from the CAVE dataset and report the RMSE of hyperspectral reconstruction. As expected, the non-physical responses are not smooth, and some portions are even negative, so they cannot be directly realized in film filters.

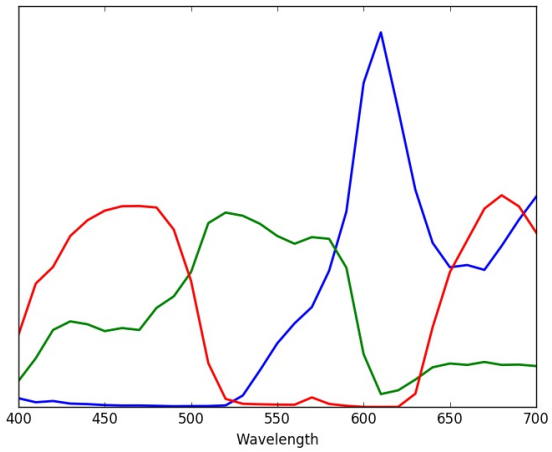


Figure 4.10 Optimal spectral response of filter array trained on CAVE dataset[154]. The corresponding array is shown in Fig4.2. Y axis stands for the amplitude.

4.3 Experiment Results on Harvard Datasets

Here, we compare the experimental results on the Harvard Nature and Harvard Mixed Datasets [15]. Comparisons between the generated hyperspectral images and ground truth are shown in Figure 4.11 and Figure 4.12 at different wavelengths. The error maps of the RMSE show our method is competitive with the alternatives. The recovered spectra of random samples are given in the main paper, in Figure 6, in the last two rows.

4.4 Experiment Results of Real-world Examples

In this section, we show more results taken by our multispectral camera using two channels with our optimized response functions. We conduct experiments on three indoor scenarios. The captured images and reconstructed spectra are shown in Figures 4.13, 4.14, and 4.15. Results show that our method achieves reasonably accurate performance.

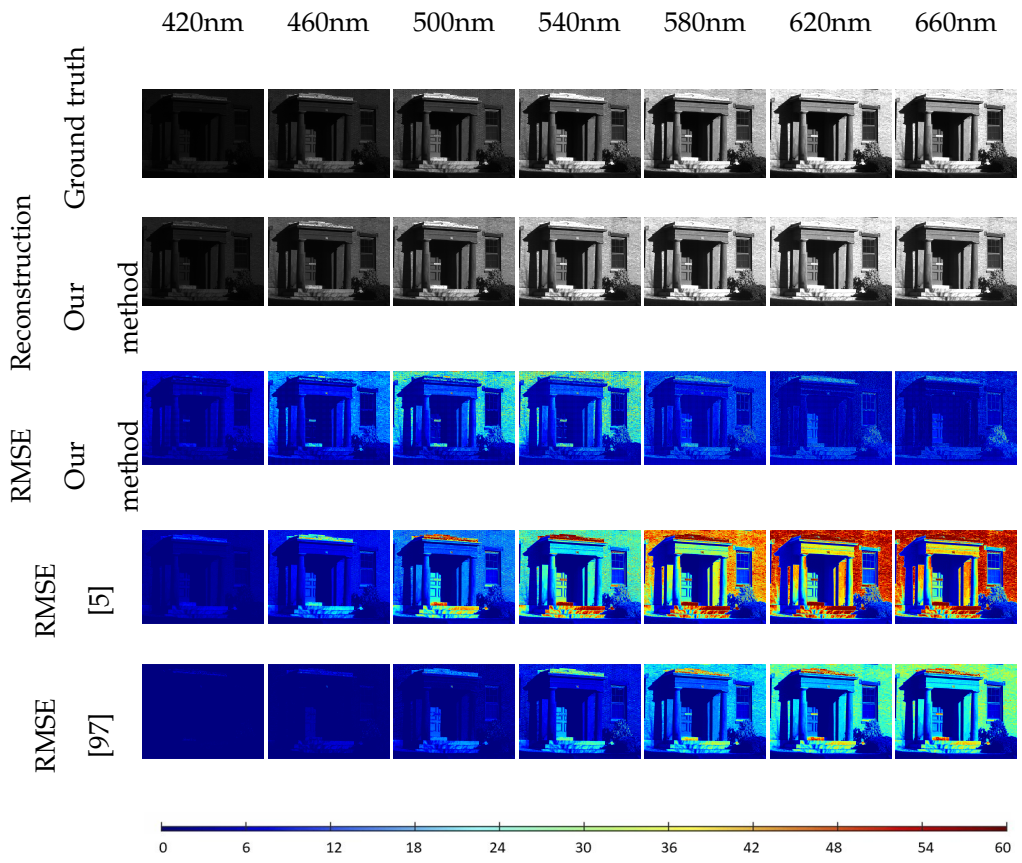


Figure 4.11 Sample Results from the Harvard Natural Database [15]

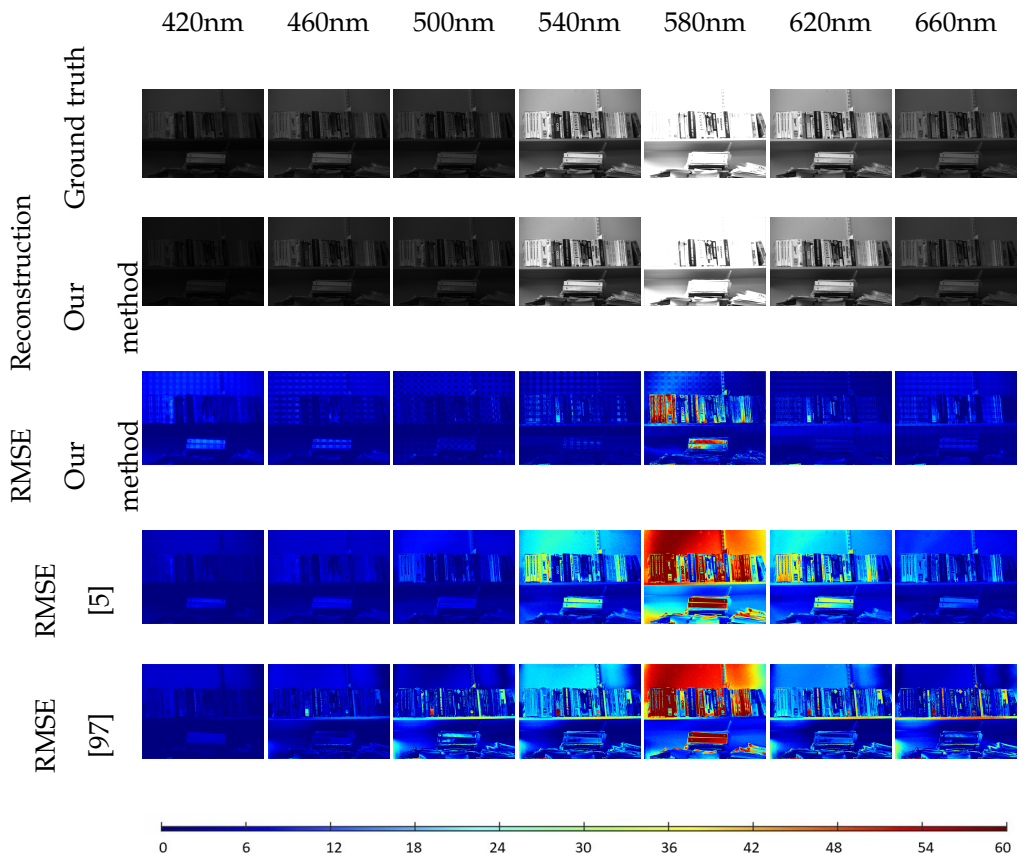


Figure 4.12 Sample Results from the Harvard Mixed Database [15]

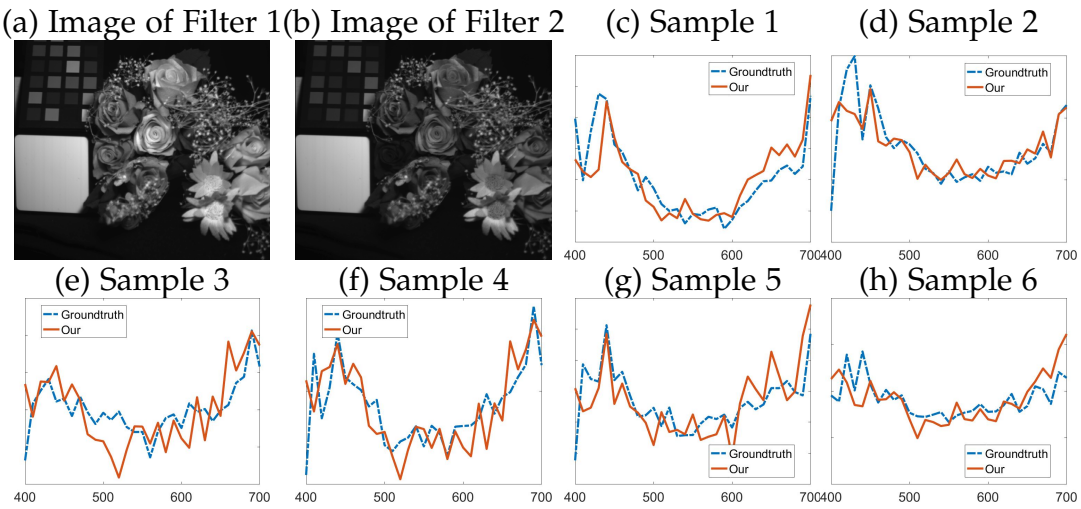


Figure 4.13 Results on flower and checkerboard from our multispectral camera. (a,b) The captured images of filters 1 and 2, respectively. (c to h) The reconstructed spectra of randomly selected single pixels.

4.4.1 Comparision Between Backbone Networks

We evaluate performance between a more recent network HSCNN [118, 150], and Unet. We concatenate a response design layer in front of an HSCNN model. As the HSCNN model does not provide a pre-trained model for CAVE dataset, we trained our network from scratch. During the experiment, we found it is hard to propagate gradient to optimize the camera design layer because of the network depth, so we decrease the original block number from 9 to 3.

The input image is cropped into 64*64 pieces to do data augmentation. Moreover, the train/test split is the same as CAVE official website. Network is optimized with Adam optimizer with learning rate 0.001, and $\beta 1$ and $\beta 2$ is set to 0.5 and 0.999. All of the weights were initialized from a Gaussian distribution with a mean 0 and a standard deviation 0.02. The RMSE of Unet or HSCNN as a backbone network are shown in Tab. 4.3

Compared to skip connection in Unet, the performance by HSCNN is significantly improved, due to the residual block and densely connected network

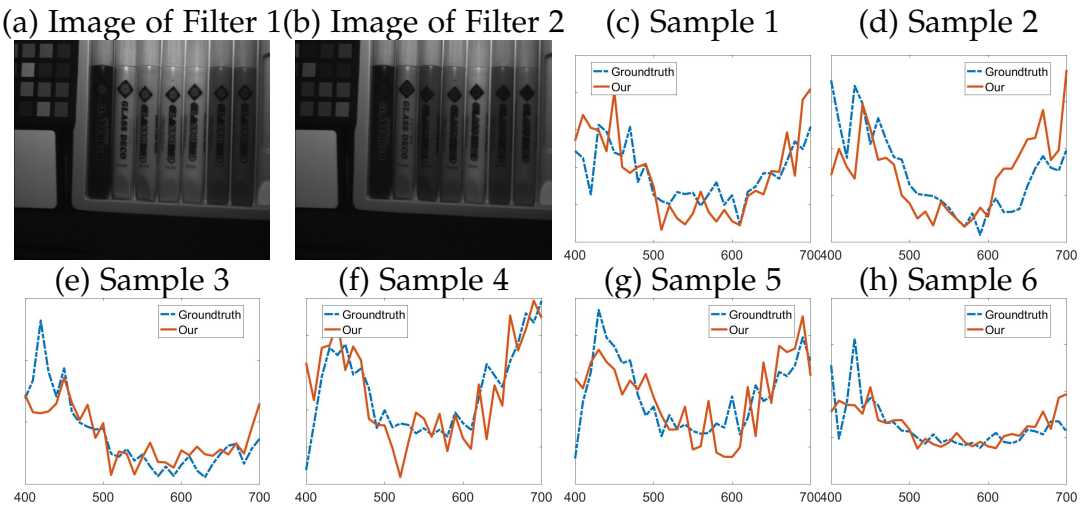


Figure 4.14 Results on books from our multispectral camera. (a,b) The captured images of filters 1 and 2, respectively. (c to h) The reconstructed spectra of randomly selected single pixels.

to minimize the loss of information flow.

4.4.2 Filter Array Design for Single Chip Setting

We also demonstrate our performance in designing the filter array (Sec. 4.1.3). When compared 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.

We present our quantitative analysis of 3 channel single-chip settings on the CAVE dataset in Table 4.4. The optimal spectral response function is given in 3.1, where the corresponding position of each spectral response function is illustrated in Fig. 4.2. 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

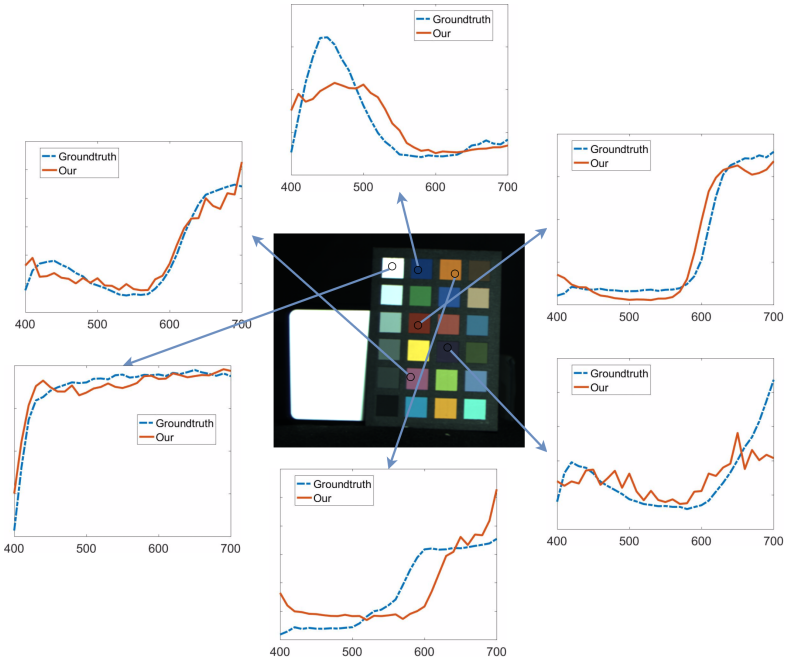


Figure 4.15 Results of single pixels spectra on color checker from our multispectral camera.

Table 4.3 RMSE for different backbone network: Unet and HSCNN. Performace evaluated under three datasets: CAVE, Harvard Natural and Mixed [154, 15, 54].

	Unet	HSCNN
CAVE	4.48	2.23
Harvard Natural	7.57	2.155
Mixed	8.88	2.69

Table 4.4 Average and Variance of RMSE of reconstruction with filter array on CAVE dataset [154].

Our	[5]	[97]
4.73 ± 3.12	13.25 ± 13.88	18.13 ± 9.33

deteriorates under the demosaicing process in the single chip setting.

4.4.3 Non-Invasive Filter Design

As shown in Table. 4.5, we use four different settings: (1) input is RGB image as lower bound; (2) Input is hyperspectral, and output is hyperspectral, network has response design layer; (3) Input is hyperspectral, and output is hyperspectral, network has response design layer and filter design layer. Canon 600D CSS function was initialized into the response design layer and keep frozen during training. (4) Input is hyperspectral, and output is hyperspectral as upper bound. We also report the relative improvement among RGB to hyperspectral reconstruction. Each of them is tested in three times with different random seed and take the average.

The learned camera response function after 300 epochs is shown in Fig. 4.16

As it is not physically plausible to implement CSS function and illumination response in Fig. 4.16, We add a smooth constraint during training. As the RMSE evaluation metric is shown in Table. 4.5

They demonstrate that smooth constraint has slightly suppressed the RMSE performance of CSS design and increased the performance of illumination design.

The learned CSS function and filter is shown in 4.17. Gaussian kernel smooth constraint is applied to smooth filter response.

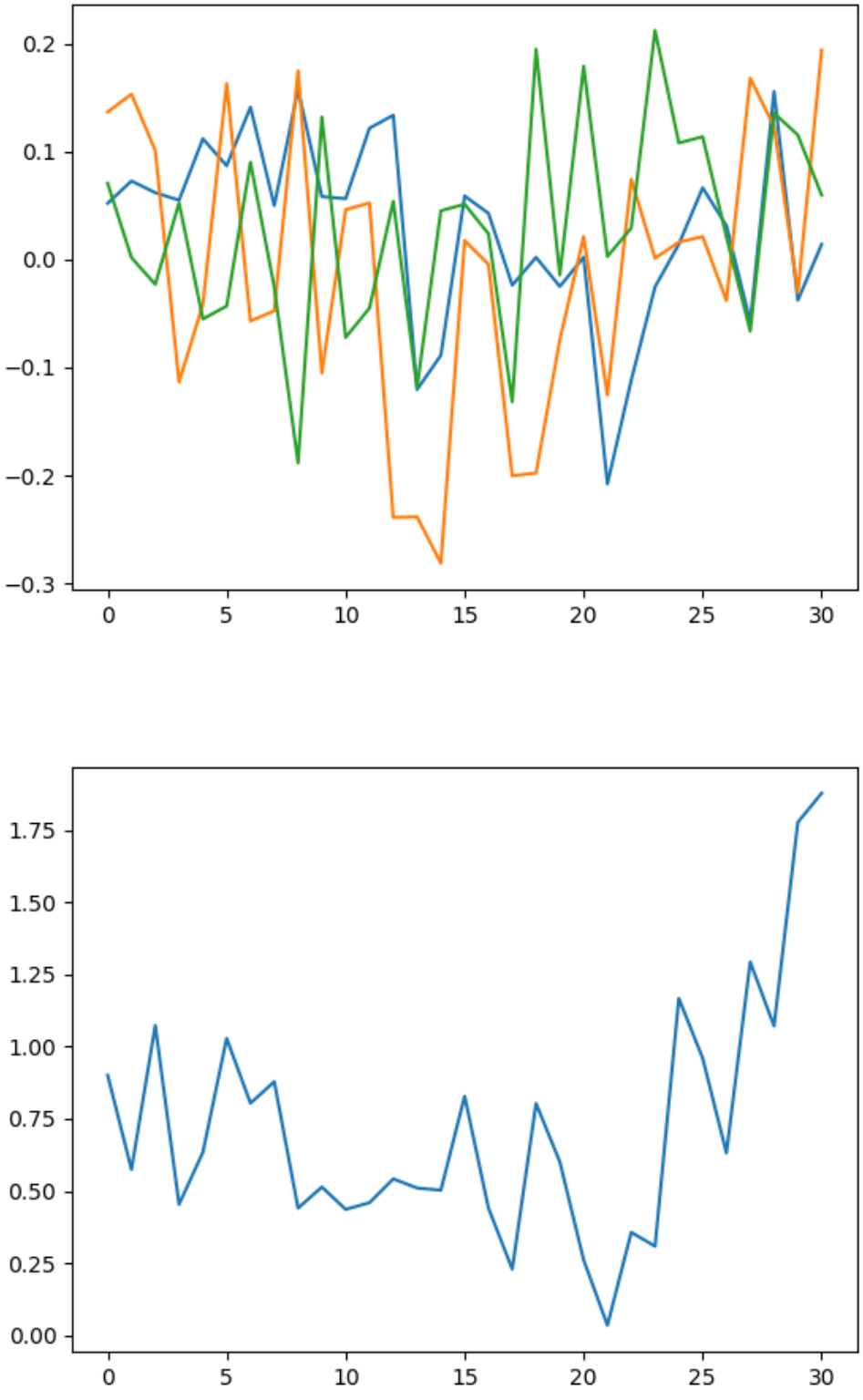


Figure 4.16 Learned camera response and filter response without constraint. x axis stands for channels number and y axis stands for amplitude.

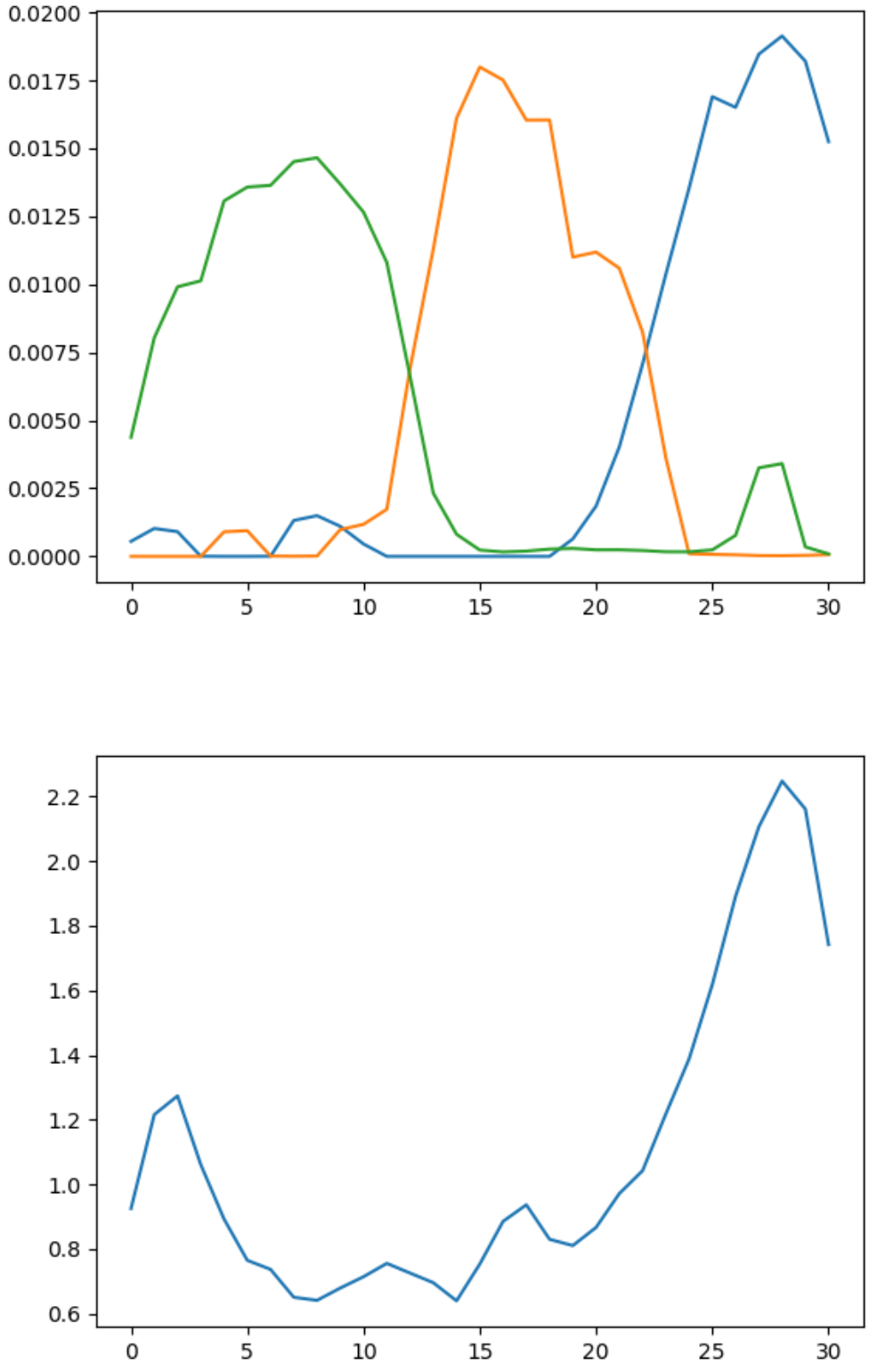


Figure 4.17 Learned camera response and filter response with smooth constraint. x axis stands for channels number and y axis stands for amplitude.

Table 4.5 RMSE on CAVE dataset for different settings. Smaller value leads to a better performance

Method	Without constraint	With constraint
RGB input	2.560	–
Response design	2.235	2.293
Filter design	2.468	2.408
Hyper input	1.006	–

Fig. 4.17 shows the learned response for fourier basis constraint.

4.5 Data-Inspired Multipectral Camera for Reconstruction

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 bandpass 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. 4.2.2. When evaluated on the CAVE dataset [154], the average RMSE of a 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 input. 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. 4.18(a). We turned to a leading optics company to

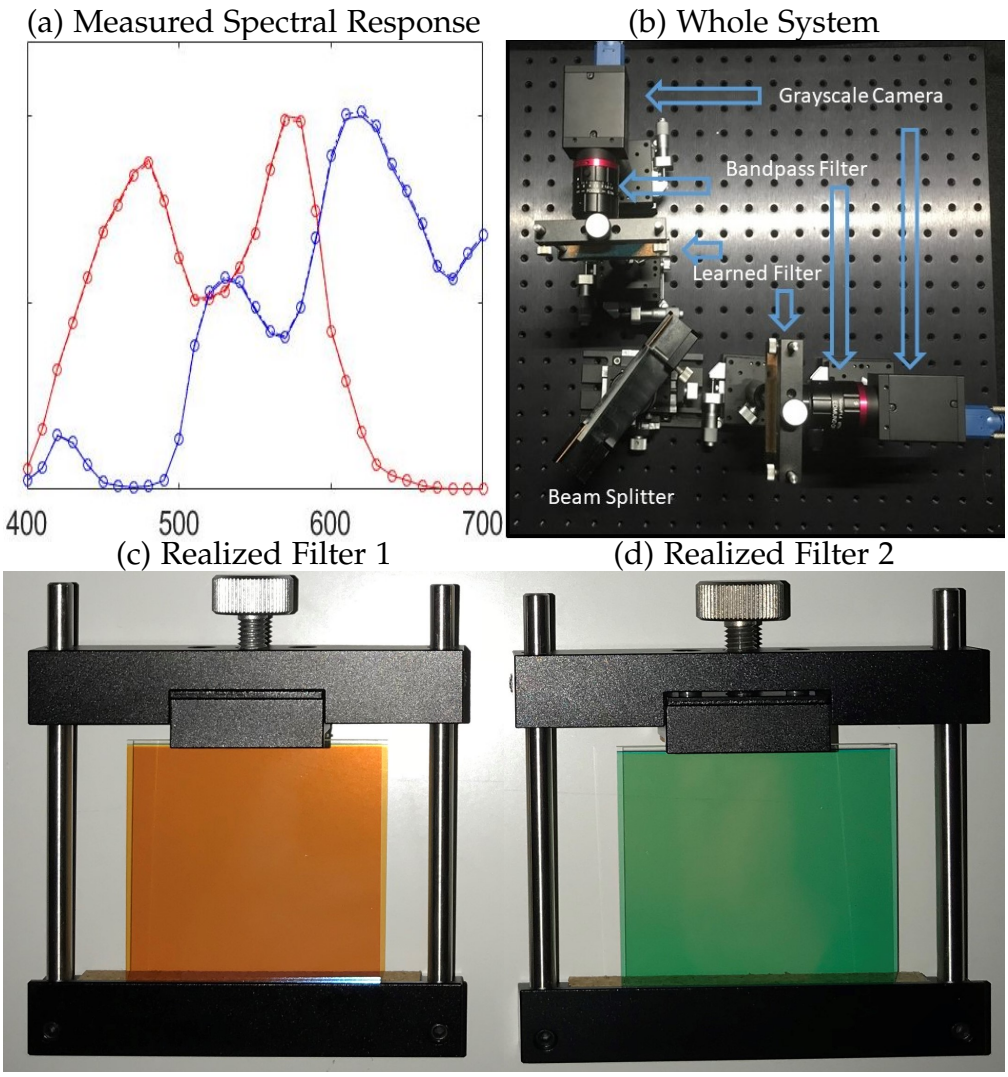


Figure 4.18 The realization of our multispectral camera. (a) The measured spectral response of our designed filter trained on CAVE [154]. Circles indicate the actual response while the solid lines are the designed spectral response function. (b) Our multispectral imaging system setup. (c) Filter of (a)'s red curve. (d) Filter of (a)'s blue curve.

implement the designed response functions. The realized film filters are of size 50mm×50mm×1mm (see 4.18(c,d)), and the measured spectral response

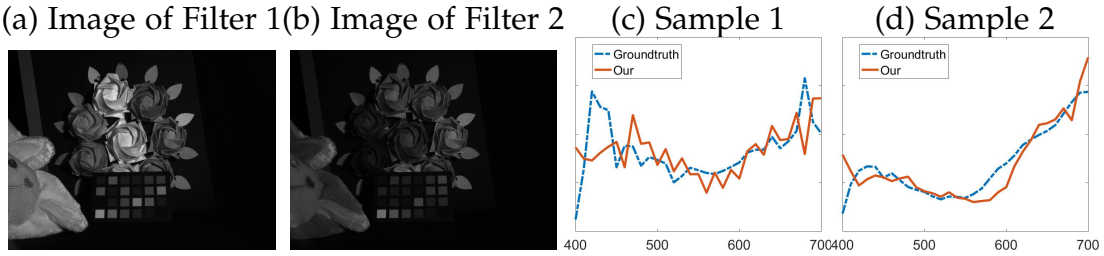


Figure 4.19 Results from our multispectral camera. (a,b) The captured images of filter 1 and 2, respectively. (c,d) The reconstructed spectra of randomly selected pixels.

functions are shown in Fig. 4.18(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. 4.18(b). Sample images captured through two filters are shown in Fig. 4.19(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. 4.19(c,d).

4.6 Computational Need

We run our proposed algorithms on NVIDIA GTX 1080 GPU. Our server is equipped with Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz and 128GB memory. The training and test costs are shown in Table 4.6

As PyTorch version 0.4 distributed training does not entirely support parallel training on four GPU cards in one machine, the time is analyzed on one single GPU. The DataParallel in PyTorch contributes to an unbalanced load, which will cause GPU 0 to memory overflow. The reason for the occurrence of load

Table 4.6 Time Consumption for training and testing on different hyperspectral datasets.

	CAVE	Harvard Natural	Mixed
Training time (hours)	1.84	8.88	8.52
Testing time per image (secs)	5.40	6.04	6.06

unbalancing is: GPU 0 is the host worker for distribute and gather gradient; this will occupy a considerable amount of memory. Our workstation provides 4 GPUs for parallel training, and third party packages are proposed recently. We suggest a distributed training framework names horovod by Uber for speed up training and an mix precision method apex to reduce GPU memory usage. Apex is a plugin for PyTorch developed by NVIDIA, which has an advantage of adaptive adjusting precision in the neural network. For example, the input image will be converted to float16 type instead of default float32 type to save space. However, when the final loss is gathered and prepare for gradient propagation, the precision is float32 to keep accuracy. This function is designed for tensor core in Volta architecture GPU like NVIDIA V100. If someone needs to implement this method in a production environment, please consider buying Volta architecture GPU and using distributed training packages mentioned above.

4.7 Joint Optimize Camera Spectral Response Selection, Sensor Multiplexing, and HSI Recovery

In our previous work, we presented a method for simultaneous learn camera spectral response and hyperspectral image reconstruction using an autoencoder like structure. In this section, we proposed an add-on layer that can also learn sensor patterns through training iterations. This layer is utilized to generate the best sensor measurements that are fed into a tail reconstruction network.

4.7.1 Black-white Pattern

Just like Coded aperture snapshot spectral imaging (CASSI), which encode the 3D hyperspectral image (HSI) into a 2D compressive image, we proposed our learned patterns, which is different at every channels. The learned pattern size is 8*8 and is repeated eight times each dimension to fit image size 64*64.

The image capturing process is supposed to be: setting a prism after an HSI cube and each channel was masked by its corresponding pattern, then using lens group to gather lights after that pattern and sum them together into a 2D compressive image. By feeding this image into a reconstruction network, the output is the original HSI cube

By encoding this process into a deep learning network, the critical challenge is to learn the binary pattern. We use a binaryConnect method [20], which helps deep learning networks to train binary weight during propagations. The critical idea of binaryConnect is only binarize the weights during the forward and backward propagations but not during the parameter update.

The learned pattern are shown in Fig. 4.20

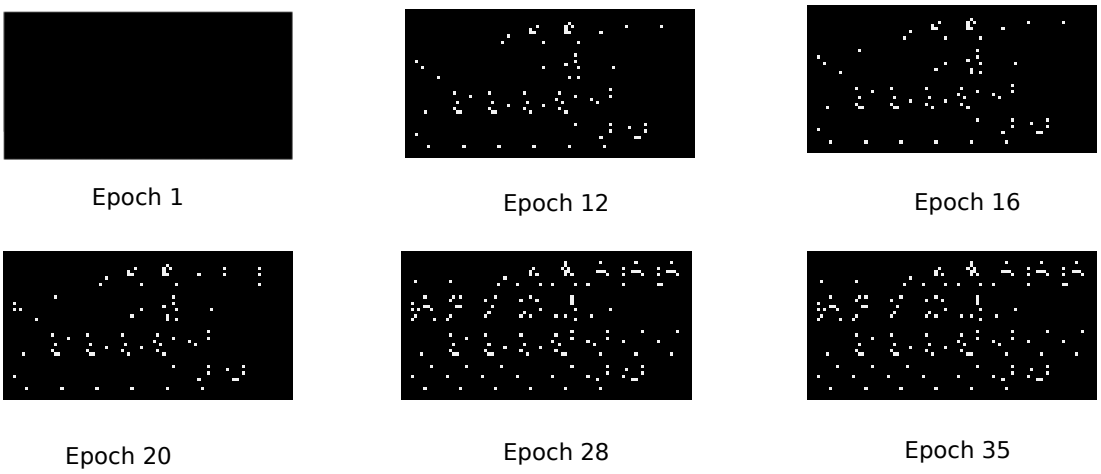


Figure 4.20 Evolution of binary pattern learning from epoch from 1 to 35.

Table 4.7 loss on validation set between learned pattern and all one (panchromatic) pattern

Panchromatic $(\times 10^{-3})$	Learned Pattern $(\times 10^{-3})$
5.731	2.642

4.7.2 Color Pattern

Previous work [14] shows designed sensor multiplexing has better performance than Bayer Pattern in raw image reconstruction. However, camera response on each sensor was constrained to be one of ['Red', 'Green', 'Blue']. In our experiments, we also learned the color matching functions for C channels.

The key challenge of learning the color pattern layer lies in choosing proper receptors among C different candidates using hard non-differentiable decision between C possibilities. To solve this challenge, despite of using softmax function directly, we use a training time dependent parameter α_t to constrain gradient of softmax function. Moreover, the designed sensor pattern layer has different behavior between training and testing stages [121]. In the training stage, the activation $I(n)$ is defined as:

$$I(n) = Softmax[\alpha_t w(n)] \tag{4.5}$$

Where $w_n \in \mathbb{R}^{\mathbb{C}}$ is a learnable parameter for each location n of multiplexing pattern. In the testing stage, the $I(n)$ was replaced with a binary version as $I^c(n) = 1$ for

$$c = \underset{c}{\text{argmax}}\ w^c(n)$$

and 0 otherwise. The output channel $c = soft\,\text{argmax}(\alpha x) = I(n)^T x(n)$

The difference between $soft\,\text{argmax}(\alpha x, 0)$ (i.e $softmax(\alpha x, 0)$) and $max(x, 0)$ is shown in Fig.4.21:

when learning the sensor pattern, we set the softmax parameter according to a quadratic schedule as $\alpha_t = 1 + (\sigma t)^2$ where $\sigma = 2.5 \times 10^{-2}$ The results on CAVE dataset was shown in Table. 4.8. We observe that our method outperforms the white sensor significantly.

The learned 8*8 pattern in show in Fig. 4.22

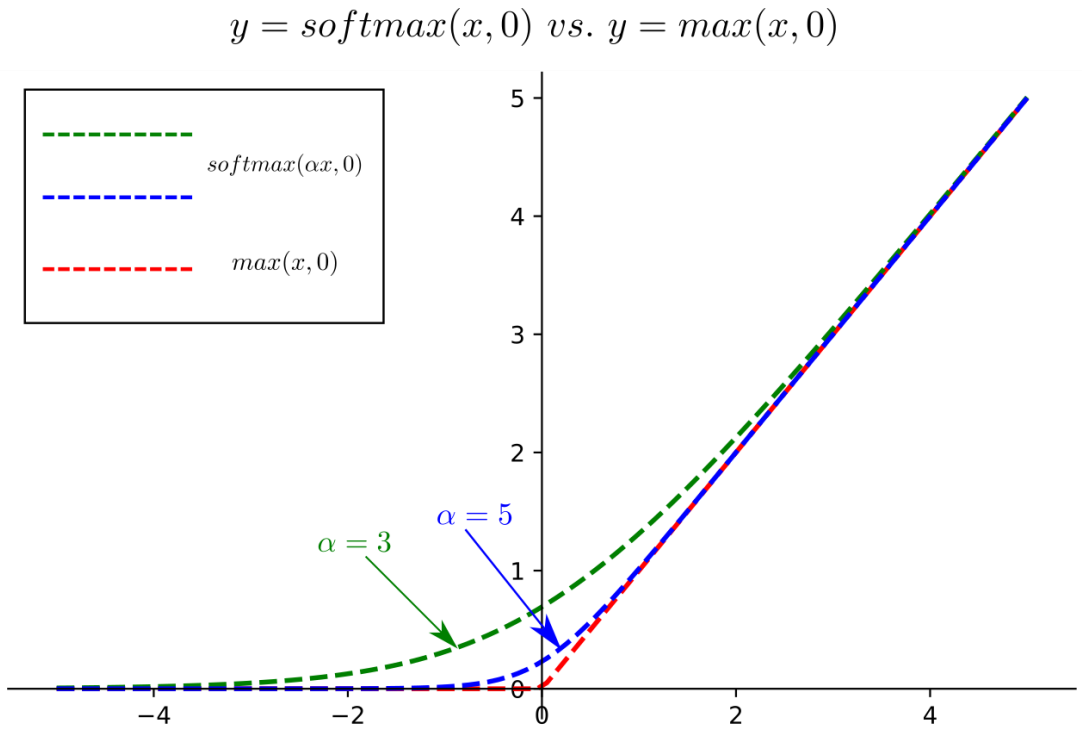


Figure 4.21 Sofxmax(x,0) vs. max(x,0). As the temperature parameter of softmax increase, it converges to hardmax function.

Table 4.8 loss on validation set for learning multiplexing pattern.

White Sensor $(\times 10^{-4})$	Multiplexing Sensor $(\times 10^{-4})$
56	9.3

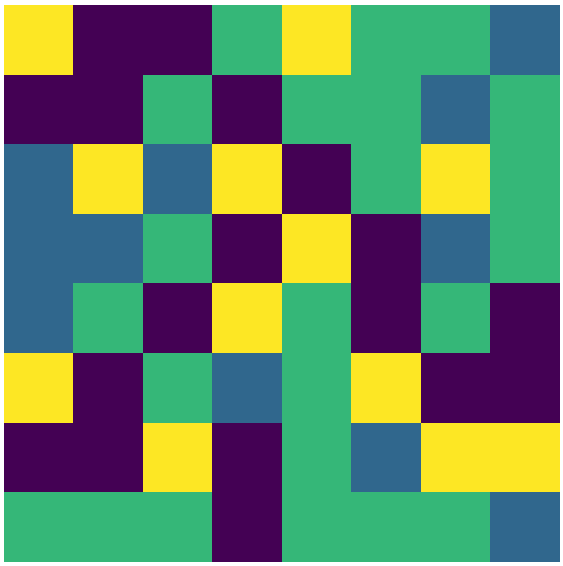


Figure 4.22 Learned multiplexing pattern

4.7.3 Ablation Study

To evaluate the effectiveness of our proposed method of joint design pattern and CSS response, we have experimented with doing these settings individually. The upper left of the Table. 4.9 shows PSNR that CSS function and pattern are all frozen into Canon 600D and Bayer Pattern. The lower right shows the PSNR of joint learn pattern and CSS response. The other part of that table is freezing one and learn another.

We can see that joint learn patterns and response can effectively improve the reconstruction results.

Relative MSE Loss vs. MSE Loss

Previous experiment on EBA Japan data provides us some insights about choosing proper object function for the network. In EBA Japan Dataset, hyperspectral images were taken from the outdoor scene where illuminate level varies significantly, which makes network focus on learning highlight part, for example,

sky. To solve this, Relative MSE was introduced and had better performance on that dataset. Similar observations are provided in Z. et al. [118]. However, in CAVE dataset, an indoor scene dataset, most images were taken in front of the black background; using Relative MSE will force the network to focus on leaning black part, which is negligible. During the training part, we found it is hard for network to converge, and final loss is much more significant than one using MSE loss.

Table 4.9 PSNR, SSIM, MSE loss, RMSE of different settings in test set.

	Bayer_600D	Bayer_learn	Learn_600D	Lean_Learn
PSNR	32.0723	32.4156	33.3676	**33.4827**
SSIM	0.9493	0.9341	**0.9538**	0.9479
val_loss(E-5)	9.9	9.03	6.51	**6.28**
RMSE	0.03141	0.2995	0.02538	**0.02491**

4.8 Conclusion

In this chapter, 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 also extend this work to optimize the camera for a broader range of vision tasks such as classification [9].

We also jointly learn a non-destructive filter, a coded pattern for hyperspectral reconstruction. To learn a coded pattern, we inspired by binary deep learning networks and adjust the slope of softmax function during training. Ablation study shows our method outperforms Bayer patterns in hyperspectral reconstruction.

Chapter 5

Physics-Based Constraints for Hyperspectral Classification

In this chapter, I will show the problem definition and our solutions in Section 5.1. Then, the network details is proposed in Section 5.2. Moreover, the dataset processing and experiments results are presented in Section **??**. Finally, the conclusion of this chapter is discussed.

5.1 Hyperspectral classification

The hyperspectral classification problem can be described as the following: Take an input B-bands hyperspectral cube, which can be formulated as a set of n pixels vectors $\mathbf{X} = \{\mathbf{x_j} \in \mathbb{R}^B, j = 1, 2, 3, ...n\}$. Let $\Omega = \{\omega_1, \omega_2, \omega_3, ..., \omega_K\}$ be a set of information classes in the scene. Classification consists of assigning each pixel of the scene to one of the K classes of interest. A typical information class of a remote sensing is (snow, water, wheat, trees, roof, etc.). Intuitively, one may believe that spectrum for each pixel could represent the sufficient information for corresponding class. However, as I mentioned in Chapter 2,

the analysis of hyperspectral image suffers from the curse of dimensionality, prevent robust statistical estimations. Therefore, to take full advantage of the rich information and get rid of redundant information, researcher developed many algorithms. For example, hyperspectral bands selection, PCA analysis and related algorithms are proposed to do dimension reduction. However, these methods are not designed for the classification task. For instance, the optimal object of PCA analysis is designed for finding the components which contribute most to the variance of original spectrum. The limitation is the optimal object of PCA is not for classification. Inspired by this, the aim of this chapter is to find a proper mapping, which project the original spectrum to three components. The optimal object is classification defined and constrained by a hyperspectral classification network. As RGB camera acquisition is designed to mimic the human eye, I will show the superior performance of our method than RGB image in the following sections.

5.2 Spectral Classification Network

5.2.1 Network Structure

Most image classification method works on RGB image shown in Fig 5.1 that takes the image of three channels (red, blue, and green) as input. On the contrary, the input of hyperspectral classification, as shown in Fig 5.2, is the hyperspectral image with multiple channels. Though hyperspectral images contain much more spectral information than RGB images, capturing it is much more expensive and inconvenience than RGB images.

To explore information conveyed in hyperspectral images, we propose a novel network in Fig 5.3 whose input is mere RGB image. As described in Chapter 3, the physical process of camera response function converts the multiple bands of hyperspectral imaging into RGB channels. We can embed this process into a 1x1 convolution layer with three kernels. Therefore, in the training stage, our training input is hyperspectral imaging, which would be projected

into three-channel images. In the actual testing stage, we could directly capture this three-channel image with specifically designed CCD filter. We then can treat it as an RGB image with a standard RGB classification network.

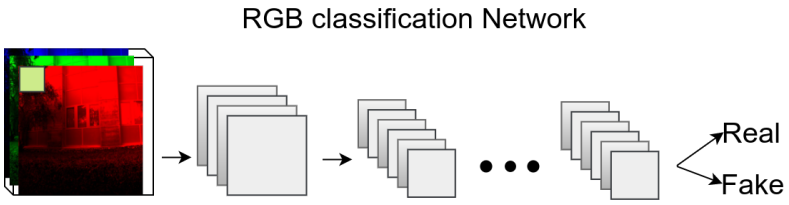


Figure 5.1 RGB classification net

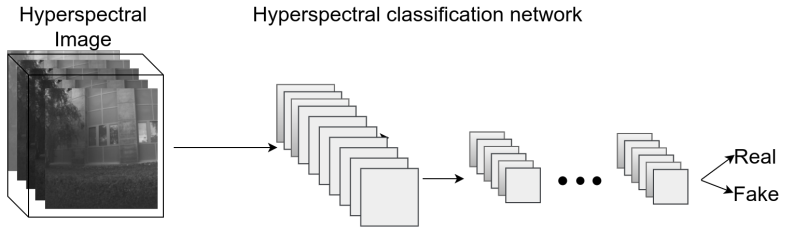


Figure 5.2 Hyper classification net

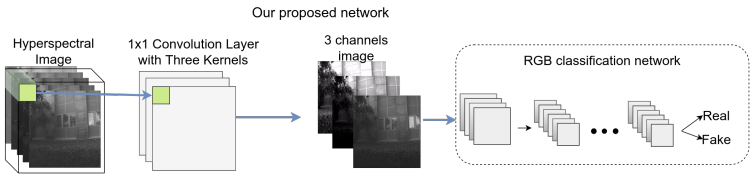


Figure 5.3 Our proposed net

Noted that arbitrary end-to-end network could be used for our spectral classification. In the previous reconstruction network, we use a well-known U-net [112], which has been widely used for image-to-image translation applications, such as pix2pix, CycGAN, Semantic Segmentation [117], and hyperspectral reconstruction [4]. However, existing classification net-like VGGnet [119] designed for normal RGB image is too large for 31 input channels. Theoretically

speaking, the depth of VGGnet is too large to be a proper choice for small datasets. Inspired by the structure design of a generative adversarial network for hyperspectral image classification [170] which has shallow net layers, we propose network as shown in Fig5.3. We use modules formed as follows: 2D convolution-Relu. The network takes images of size n×n×3 as input and finally produces the corresponding spectral images of size n×n×31. Let *Ck* denote a convolutional block including one convolutional layer with k filters, one ReLU activation layer. We removed the batchNormalization layer and got a better performance. The α parameter in the leakyReLU layer is set to 0.2.

All the convolutional layers are set to 3*3 kernel size, stride 1, with max-pooling factor 2, and proper zero paddings to edges. The network structure is C31-C3(1*1 convolution)-C64-C16-D32-D16-D8-D1. D stands for the dense layer. Specifically, the first layer is the embedded bottle-neck layer that embeds camera response function.

5.2.2 Smooth Constraint Using Fourier Basis

During the training stage, we observed that the smoothness of designed camera response is not sufficient even with l2 regularization, as shown in Fig. 5.5. This would increase the cost of manufacturing the actual filter. To further impose the smoothness, we proposed a method to learn a combination of orthogonal basis. By using cut-off frequency, we could control the smoothness of learned weight. A typical set of orthogonal basis is Fourier series $\{1, \sin x, \cos x, \sin 2x, \cos 2x, ..., \sin nx, \cos nx\}$. By assuming camera response $f(x)$ is a linear combination of these basis, we have:

$$f(x) = a_0 + \sum_{n=1}^{N} \left\{ a_n \cos(\frac{2\pi nx}{P}) + b_n \sin(\frac{2\pi nx}{P}) \right\} \tag{5.1}$$

We use gradient descent method to optimize a_n, and b_n, the maximum frequency was constrained to N/P. By encoding this Fourier Series into our network, the learned camera spectral response is shown in Fig. 5.4, details of learning is show in section 5.

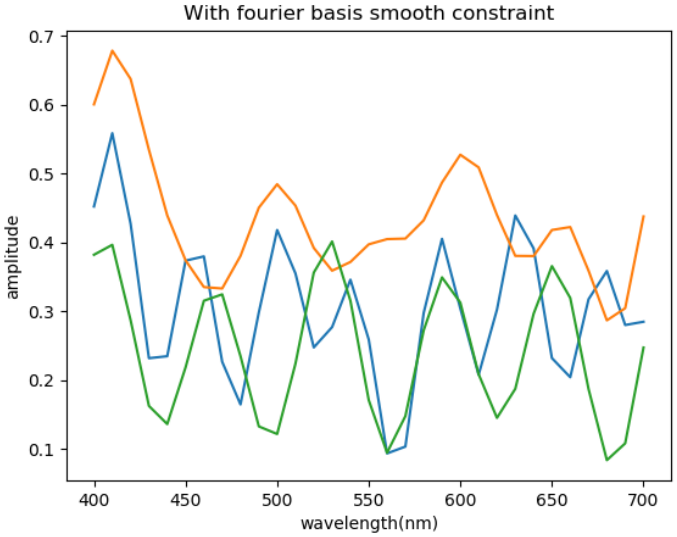


Figure 5.4 Camera spectral response with cut-of frequency, n=15. This is a more strong constraint than l2 norm.

As a potential application, we can also learn the filter that is mounted on existing cameras with known spectral response function.

5.3 Dataset and Experiment Results

5.3.1 Cave Real and Fake Pepper

As far as we know, there is no public dataset of labeled hyperspectral segmentation dataset in visible wavelength range. However, we found there is one image in CAVE described in sec. 4.2.1; the experiment was conducted on that image. In this image, a real pepper and a false pepper are put in front of a black background. We extract the foreground and analyze the appearance. As the dataset size is too small, we split original images into several patches; Yan *et al.* also used this approach in remote sensing segmentation [82]. The patches are

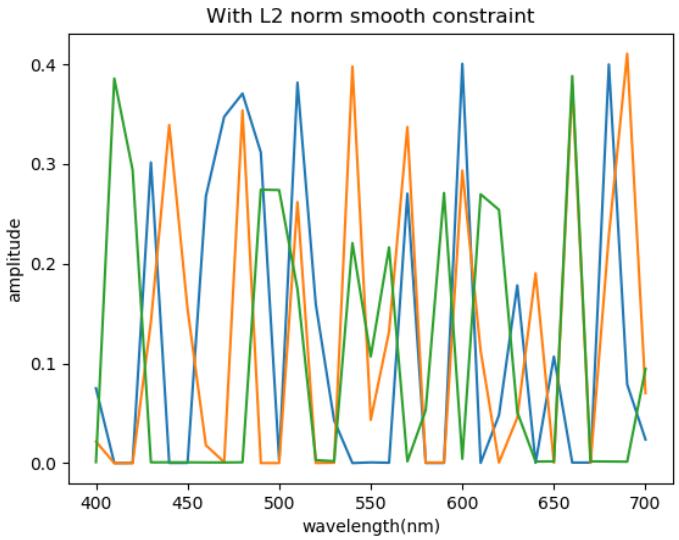


Figure 5.5 Generated camera spectral response with l2 norm regularization. Response function is jagged and physically implausible

split into train and test sets with ratio 0.7. The patch size is 128*128 with stride 1. Another challenge is small dataset size leads to a jagged learned response even if l2 constraint is applied. Thus we use a more strong constraint using Fourier basis described Equation 4.4. Parameter n in that equation is set to 15.

As shown in Fig 5.6, we compared our method with five settings:

1. Input is an RGB image, network structure is C3-C64-C16-D32-D16-D8-D1.
2. Input is a hyperspectral image, and the weights in the filter design layer are frozen to commercial camera spectral response. Network structure is C31-C3(1*1 convolution)-C64-C16-D32-D16-D8-D1.
3. Input is a hyperspectral image, filter design layer is trained without constraint. Network structure is C31-C3(1*1 convolution)-C64-C16-D32-D16-D8-D1.
4. Input is a hyperspectral image, with no filter design layer as bottleneck

layer. Network structure is C31-C64-C16-D32-D16-D8-D1.

5. Input is a hyperspectral image; filter design layer is trained with smooth constraint using Fourier basis to control maximum frequency. This makes trained response physically plausible for manufacturers. Network structure is C31-(Fourier layer)-C64-C16-D32-D16-D8-D1.

Intuitively, in this dataset, classification accuracy by hyperspectral image works much better than RGB image and using our learned response, accuracy is largely improved. However, due to the dataset size, learned response is jagged shown in Fig 5.5 and cannot be smoothed using l2 norm. To make a trade-off between smoothness and final accuracy, a Fourier basis is introduced. The learned response is shown in Fig 5.4.

5.3.2 Remote Sensing Dataset

India Pines [113] dataset is gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in northwestern Indiana. This dataset has 220 spectral channels from 400 nm to 2499 nm include visible and infrared spectrum. To make a fair comparison, we discard infrared wavelength and generate three channels image with re-sampling on existing camera response. The same sensor of India Pines collected the Salinas dataset at the place of Salinas Valley, California. As there are so many bands in original data, and wavelength range between each band is not fixed, calibration information and resampling the camera response function using 3-order B-spline interpolation are applied as a pre-processing method to get a 30 bands multispectral image in visible wavelength range. The AVIRIS sensor calibration information is from Purdue University Research Repository (PURR) [1]. Documnet name is Calibration_Information_for_220_Channel_Data_Band_Set.txt. We plot the center wavelength (nm) in this document shown in Fig. 5.7 and using this information to resampling the camera response function shown in Fig. 5.8. After we got

[1] https://purr.purdue.edu/publications/1947/supportingdocs/1

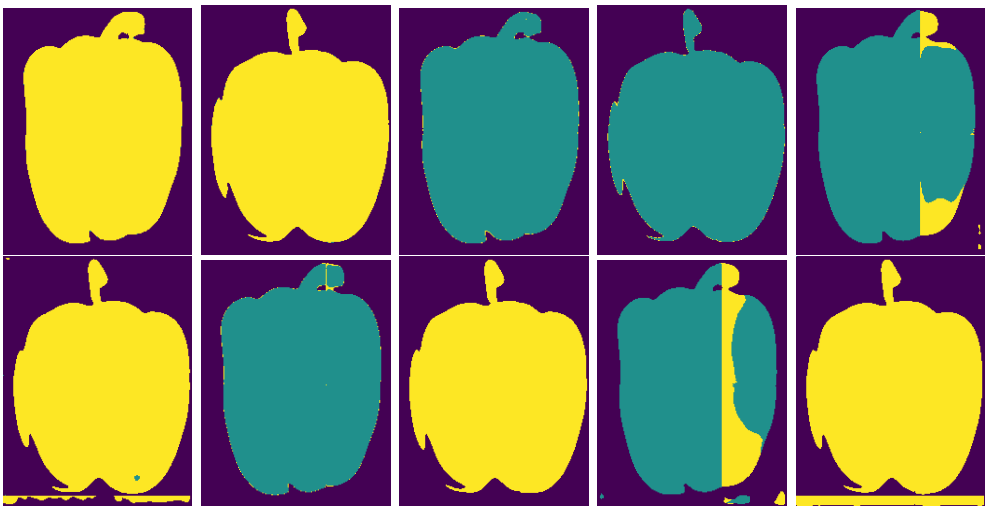


Figure 5.6 Predicted segmentation mask for real and fake pepper. Yellow label stands for fake label and blue one stands for real label. The first and second row stands for the corresponded mask for real and fake pepper image input. Each column stands for: (1) rgb input image as lower bound. (2) first layer (camera response layer) initialized by canon 600D and freezed during training. (3) proposed method to set all the weights trainable to design camera response. (4) hyperspectral input image as upper bound. (5) same setting as (2), but to learn the linear combination parameters of fourier basis.

the resampled weight of AVIRIS sensor, we can get the 'RGB' image of remote sensing dataset for comparing the final results with our proposed method.

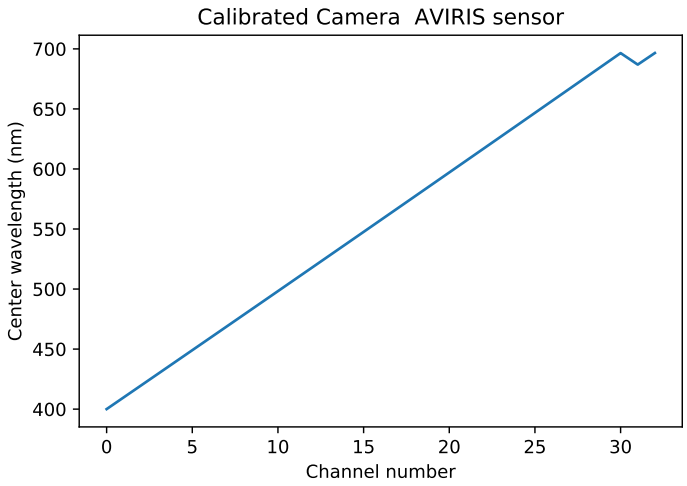


Figure 5.7 The AVIRIS sensor calibration information taken from Purdue University Research Repository (PURR): 220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3.

The overall accuracy is shown in Table. 5.1. Experiments result show that the proposed method is significantly improved compared to traditional RGB image.

Pavia University

Pavia Univ dataset was acquired using ROSIS (Reflective Optics System Imaging Spectrometer) over the Pavia University.It has 610*340 pixels and 103 bands, wavelength from 0.43 to 0.86μm. This dataset has ten classes. The training and testing set was randomly separated, with a ratio of 7:3. The visible wavelength band is top 68 channels. Note that the calibration information is missing compared to AVIRIS sensor. So we assume that the narrow-bands images are taken in wavelengths with same intervals.

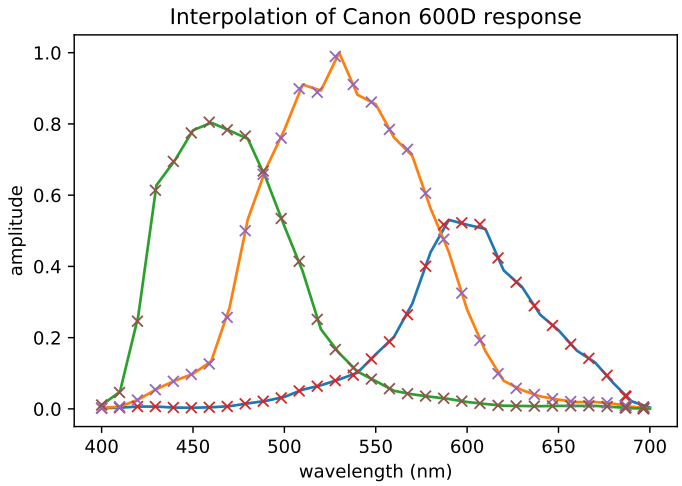


Figure 5.8 Using resampling method from AVIRIS sensor calibration information on canon 600D response. This is used for generating RGB image of remote sensing datasets.

Classification result are shown in Fig. 5.11. Filter was learned using the top 68 channels, and the RGB image is calculated using the interpolation of Canon 600D. Note that the Response design and RGB results are generated from visible wavelength in original spectrum. In our design, we have about 3 % overall accuracy improvement using response design on the India Pines dataset and 10 % improvement on the Salinas dataset. Classification result is shown in Fig. 5.9. and Fig. 5.10

5.4 Conclusion

We build a framework for material classification, in the training stage, our training input is hyperspectral imaging which would be projected into 3 channel images. In the actual testing stage, we could directly capture this 3 channel image with specifically designed CCD filter. We then can treat it as an RGB

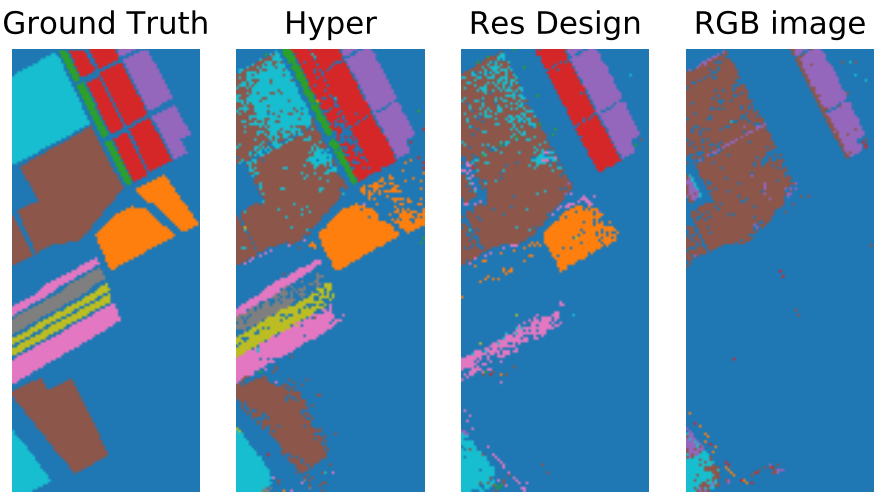


Figure 5.9 Classification result in Salinas Data Set. From left to right stands for ground truth, hyperspectral input, hyperspectral input with filter learning, RGB input respectively.

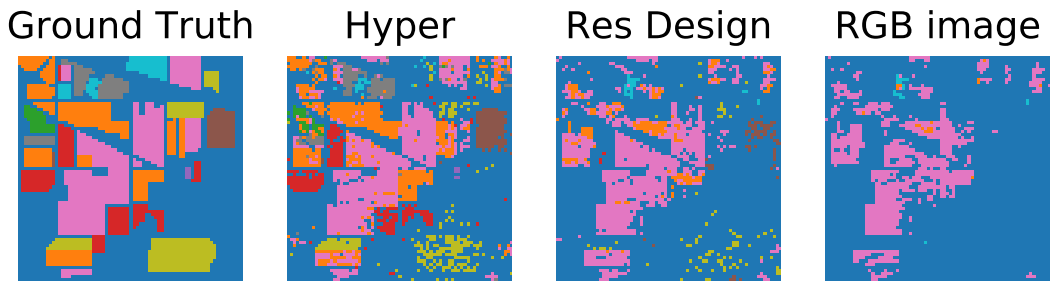


Figure 5.10 Classification result in Indian Data Set.From left to right stands for ground truth, hyperspectral input, hyperspectral input with filter learning, RGB input respectively.

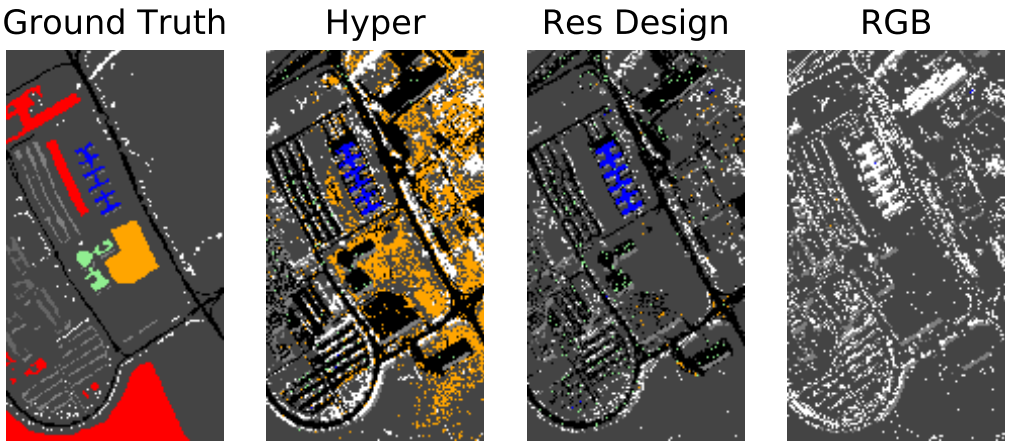


Figure 5.11 Classification result in Pavia University Data set.From left to right stands for ground truth, hyperspectral input, hyperspectral input with filter learning, RGB input respectively.

Table 5.1 Overall Accuracy in three public hyperspectral datasets

OA(Overall Accuracy)	Hyper	RGB	Response design	Random
India Pines	0.7616	0.5609	**0.5907**	0.0571
Salinas	0.8660	0.6250	**0.7296**	0.0587
Pavia University	0.7549	0.4974	**0.6598**	0.1023

image with a standard RGB classification network. With this setting, we can build a snapshot hyperspectral classification system. During the training stage on CAVE pepper dataset, we observed that l2 regularization is no sufficient for controlling smoothness in this small dataset. To further impose the smoothness, we propose a fourier basis layer in classification network and learn the parameter of Fourier series.

Chapter 6

Conclusion

This thesis propose a new physical-based deep learning framework for analyzing optical properties for real scenes. Optical property has two essential parts: global/direct separation and hyperspectral reconstruction. For recovering global and direct illuminations, we add a physical constraint as a term on the network loss. For recovering and analyzing pixel spectral reflectance, we add a smooth constraint using l2 norm and Fourier basis as a term on the network loss in conjunction with particular designed convolution neural network layer.

In Chapter 1, I explain the drawback of current approaches for optical property analysis. Firstly, separating direct and global components requires multiple images taken under a specific setting, such as high-frequency light patterns. 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). Unfortunately, these devices are extremely limited in spatial resolution. To analyzing optical properties, several deep learning and machine learning based methods were proposed, but the transparency of deep learning method is lacking, which is highly parameter fitting. We build the connection between camera sensor spectral/spatial distribution and convolution layer of neural network. By simulating the camera response as a convolution

layer and tailed with network for analyzing, the performance is better than state-of-the-art.

In Chapter 2, I provide a literature review of previous works related on the application of machine learning algorithms on optical properties analysis and their limitations.

In Chapter 3, I design a physical-based framework to separate direct and global component, which is an important part of optical property analysis. We propose the first method to separate direct and global components from a single RGB image without hardware constraints. This model embeds physical prior knowledge into the GAN based network to achieve single-image components separation. To train and evaluate this model, we also present the first dataset, which comprises of 100 scenes with their ground truth direct and global components. Our method has been shown to work successfully on our own testing set and general images from the public dataset. Finally, we demonstrate how the separated components could be used for realistic image editing.

In Chapter 4, I discuss how to impose physical prior constraints to spectral reconstruction, which is anather important task of optical preperty analysis. For RGB to hyperspectral reconstruction, although previous work can recover spectral reflectance using input RGB image [97, 111, 5, 54, 32], and shows camera spectral spectral is important for hyperspectral reconstruction [6]. At first, we feed RGB image into a network and add a new physical inverse loss as a weighted sum of output hyperspectral image, where weights are equivalent to existing cameras. After adding this loss into the final objective function, the reconstruction error is reduced. By considering searching for the best filter response function, we build and end-to-end network simultaneously learn the optimized filter response functions and the mapping for spectral reconstruction and classification. With the designed filters, we propose a data-inspired multispectral camera for snapshot hyperspectral imaging.

We also jointly learn a non-destructive filter, a coded pattern for hyperspectral reconstruction. To learn a coded pattern, we inspired by binary deep learning networks [14] and adjust the slope of softmax function during training. Ablation study shows out the method outperforms Bayer patterns in

hyperspectral reconstruction.

In Chapter 5, I explore the application of optical property analysis such as material classification. In the training stage, our training input is hyperspectral imaging which would be projected into 3 channel images. In the actual testing stage, we could directly capture this 3 channel image with specifically designed CCD filter. We then can treat it as an RGB image with a standard RGB classification network. With this setting, we can build a snapshot hyperspectral classification system, and significantly improve the performance.

Chapter 7

Publications

1. S Nie, L Gu, Y Zheng, A Lam, N Ono, I Sato. (2018). Deeply Learned Filter Response Functions for Hyperspectral Reconstruction. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 4767–4776).
2. S Nie, L Gu, A Subpa-Asa, I Kacher, K Nishino, I Sato. "A Data-Driven Approach for Direct and Global Component Separation from a Single Image." Asian Conference on Computer Vision. Springer, Cham, 2018.

Bibliography

[1] S. Achar, S. T. Nuske, and S. G. Narasimhan. Compensating for Motion during Direct-Global Separation. In *IEEE Int. Conf. Comput. Vis.*, pages 1481–1488, 2013. ↰ pages 5, 19, and 30

[2] N. Akhtar, F. Shafait, and A. Mian. Sparse Spatio-spectral Representation for Hyperspectral Image Super-resolution. In *Proc. Eur. Conf. Comput. Vis.*, pages 63–78, 2014. ↰ page 21

[3] N. Akhtar, F. Shafait, and A. Mian. Hierarchical Beta Process with Gaussian Process Prior for Hyperspectral Image Super Resolution. In *Proc. Eur. Conf. Comput. Vis.*, pages 103–120, 2016. ↰ page 21

[4] A. Alvarez-Gila, J. van de Weijer, and E. Garrote. Adversarial Networks for Spatial Context-Aware Spectral Image Reconstruction from RGB. *IEEE Int. Conf. Comput. Vis. Work. (ICCVW 2017)*, 2017. ↰ pages 7, 22, 25, 40, and 77

[5] B. Arad and O. Ben-Shahar. Sparse Recovery of Hyperspectral Signal from Natural RGB Images. *ECCV*, pages 19–34, 2016. ↰ pages 7, 21, 22, 49, 52, 54, 56, 57, 61, and 90

[6] B. Arad and O. Ben-Shahar. Filter selection for hyperspectral estimation. In *ICCV*, pages 3172–3180, 2017. ↰ pages 7, 10, 23, and 90

[7] D. Berthelot, T. Schumm, and L. Metz. BEGAN: boundary equilibrium generative adversarial networks. *arXiv Prepr. arXiv1703.10717*, 2017. ↰ page 33

[8] B. Bigdeli, F. Samadzadegan, and P. Reinartz. A Multiple SVM System for Classification of Hyperspectral Remote Sensing Data. *J. Indian Soc. Remote Sens.*, 41(4):763–776, 2013. ↰ pages 13, 14, and 22

[9] H. Blasinski, J. Farrell, and B. Wandell. Designing illuminant spectral power distributions for surface classification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, volume 2017-Janua, pages 2682–2691, 2017. ↰ pages 9, 15, and 74

[10] I. Boyadzhiev, K. Bala, S. Paris, and E. Adelson. Band-Sifting Decomposition for Image-Based Material Editing. *ACM Trans. Graph.*, 34(5):163:1—-163:16, 2015. ↰ page 6

[11] 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. ↰ pages 7 and 21

[12] X. Cao, X. Tong, Q. Dai, and S. Lin. High resolution multispectral video capture with a hybrid camera system. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 297–304, 2011. ↰ page 21

[13] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised Diverse Colorization via Generative Adversarial Networks. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 10534 LNAI:151–166, 2017. ↰ page 42

[14] A. Chakrabarti. Learning Sensor Multiplexing Design through Back-propagation. (Nips):1–9, 2016. ↰ pages 70 and 90

[15] A. Chakrabarti and T. Zickler. Statistics of Real-World Hyperspectral Images. In *Proc.˜IEEE Conf.˜on Comput. Vis. Pattern Recognit.*, pages 193–200, 2011. ↰ pages xiii, xiv, xvii, 48, 50, 54, 55, 56, 57, and 60

[16] C. Chang, Q. Du, T. S. I. transactions on . . . , and undefined 1999. A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification. *ieeexplore.ieee.org*. ↰ page 15

[17] C. C. Chang and C. J. Lin. LIBSVM: A Library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–39, 2011. ↰ pages 13 and 14

[18] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 903–912, 2016. ↰ pages 10 and 23

[19] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in neural information processing systems*, pages 1736–1744, 2014. ↰ page 18

[20] M. Courbariaux, Y. Bengio, and J.-P. David. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. pages 1–9, 2015. ↰ page 68

[21] P. Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, pages 32:1—-32:10, New York, NY, USA, 2008. ACM. ↰ page 6

[22] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 914–922, 2017. ↰ page 18

[23] N. Doulamis. Adaptable deep learning structures for object labeling/tracking under dynamic visual environments. *Multimedia Tools and Applications*, 77(8):9651–9689, 2018. ↰ page 18

[24] N. Doulamis and A. Voulodimos. Fast-mdl: Fast adaptive supervised training of multi-layered deep learning models for consistent object tracking and classification. In *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 318–323. IEEE, 2016. ↰ page 18

[25] M. T. Eismann. *Hyperspectral Remote Sensing*. 2012. ↰ pages 2 and 7

[26] et al. Bernhard E. Boser. A training algorithm for optimal margin classifiers. 2010. ↰ page 13

[27] K. S. et al. D Silver J Schrittwieser and K. S. et al. D Silver J Schrittwieser. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2017. ↰ pages 2 and 41

[28] H. Farid and E. H. Adelson. Separating reflections and lighting using independent components analysis. In *Proceedings. 1999 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (Cat. No PR00149)*, volume 1, page 267 Vol. 1, 1999. ↰ page 20

[29] M. Fauvel, J. A. Benediktsson, J. Chanussot, J. R. Sveinsson, and J. A. Benediktsson. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles Spectral and Spa-tial Classification of Hyperspectral Data Using SVMs and Morphological Profiles. IEEE Transactions on Geoscience and Remote Sensing, Institute. *Electron. Eng.*, 46(11):3804–3814, 2008. ↰ pages 13, 14, and 22

[30] C. Finn, I. Goodfellow, and S. Levine. Unsupervised Learning for Physical Interaction through Video Prediction. *Neural Inf. Process. Syst.*, (Nips):64–72, 2016. ↰ page 26

[31] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999. ↰ page 18

[32] S. Galliani, C. Lanaras, D. Marmanis, E. Baltsavias, K. Schindler, S.Galliani, C. Lanaras, D. Marmanis, E. Baltsavias, and K. Schindler. Learned Spectral Super-Resolution. *CoRP*, arXiv:1703, 2017. ↰ pages 7, 21, 22, and 90

[33] L. Gao, R. Kester, N. Hagen, and T. Tomasz. Snapshot Image Mapping Spectrometer ({IMS}) with high sampling density for hyperspectral microscopy. *Opt. Express*, 18(14):14330–14344, 2010. ↰ pages 7 and 21

[34] S. Georgoulis, K. Rematas, T. Ritschel, M. Fritz, T. Tuytelaars, and L. Van Gool. What Is Around the Camera? In *IEEE Int. Conf. Comput. Vis.*, 2017. ↰ page 6

[35] M. Golipour, H. Ghassemian, and F. Mirzapour. Integrating Hierarchical Segmentation Maps With MRF Prior for Classification of Hyperspectral Images in a Bayesian Framework. *IEEE Trans. Geosci. Remote Sens.*, 54(2):805–816, 2016. ↰ pages 9, 13, 14, and 22

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst. 27*, pages 2672–2680, 2014. ↰ pages 19 and 25

[37] J. Gu, T. Kobayashi, M. Gupta, and S. K. Nayar. Multiplexed Illumination for Scene Recovery in the Presence of Global Illumination. In *IEEE Int. Conf. Comput. Vis.*, pages 1–8, 2011. ↰ pages 5, 19, and 30

[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Adv. Neural Inf. Process. Syst.*, pages 5767–5777, 2017. ↰ page 33

[39] Y. Guo, S. Wang, C. Gao, D. S. . . . . , R. Sensing, and undefined 2015. Wishart RBM based DBN for polarimetric synthetic radar data classification. *ieeexplore.ieee.org*. ↰ page 3

[40] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan. A Practical Approach to 3D Scanning in the Presence of Interreflections, Subsurface Scattering and Defocus. *Int. J. Comput. Vis.*, 102(1-3):33–55, 2013. ↰ page 20

[41] M. Gupta, S. G. Narasimhan, and Y. Y. Schechner. On controlling light transport in poor visibility environments. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1–8, 2008. ↰ page 20

[42] M. Gupta, Y. Tian, S. Narasimhan, and L. Zhang. A Combined Theory of Defocused Illumination and Global Light Transport. *Int. J. Comput. Vis.*, 98(2):146–167, 2012. ↰ pages 6 and 19

[43] W. Havard, L. Besacier, and O. Rosec. SPEECH-COCO: 600k Visually Grounded Spoken Captions Aligned to MSCOCO Data Set. 2017. ↰ page 2

[44] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016. ↰ pages 2, 40, 41, and 42

[45] V. Heikkinen. Spectral Reflectance Estimation Using Gaussian Processes and Combination Kernels. *IEEE Trans. Image Process.*, 27(7):3358–3373, 2018. ↰ page 15

[46] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. ↰ page 17

[47] Y. C. Ho, D. L. Pepyne, and M. A. Simaan. Simple Explanation of the No-Free-Lunch Theorem and Its Implications 1. Technical Report 3, 2002. ↰ page 2

[48] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. ↰ page 17

[49] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou. Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. ↰ page 3

[50] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017. ↰ pages 41 and 42

[51] C. Igel and M. Toussaint. A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions. Technical report, 2004. ↰ page 2

[52] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv*, page 16, 2016. ↰ pages 25, 28, 31, 34, 35, and 40

[53] P. Isola, J.-Y. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR*, 2017-Janua:5967–5976, 2017. ↰ page 33

[54] 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, 2017. ↰ pages xvii, 7, 21, 52, 54, 60, and 90

[55] C. Jiang, H. Zhang, H. Shen, and L. Zhang. A Practical Compressed Sensing Based Pan Sharpening Method. *IEEE Geosci. Remote Sens. Lett.*, 9(4):629–633, 2012. ↰ page 21

[56] J. Jiang, D. Liu, J. Gu, S. Süsstrunk, and S. Susstrunk. What is the Space of Spectral Sensitivity Functions for Digital Color Cameras? In *WACV*, pages 168–179, 2013. ↰ page 7

[57] S. Kalita and M. Biswas. Improved Convolutional Neural Networks for Hyperspectral Image Classification. *Adv. Intell. Syst. Comput.*, 740:397–410, 2019. ↰ pages 9 and 22

[58] X. Kang, S. Li, L. Fang, M. Li, and J. A. Benediktsson. Extended random walker-based classification of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.*, 53(1):144–153, 2015. ↰ pages 13 and 15

[59] A. Karpatne, W. Watkins, J. Read, and V. Kumar. Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. 2017. ↰ pages 3 and 26

[60] K. Kavitha, S. Arivazhagan, and B. Suriya. Classification of Pavia University Hyperspectral Image using Gabor and SVM Classifier. *Int. J. New Trends Electron. Commun.*, 2(3):9–14, 2014. ↰ pages 9, 13, 14, and 22

[61] R. Kawakami, J. Wright, Y.-W. W. Tai, Y. Matsushita, M. Ben-Ezra, K. Ikeuchi, J. Wright, M. Ben-Ezra, Y.-W. W. Tai, and K. Ikeuchi. High-resolution hyperspectral imaging via matrix factorization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2329–2336, 2011. ↰ pages 16 and 21

[62] R. Kemker, R. Luu, and C. Kanan. Low-Shot Learning for the Semantic Segmentation of Remote Sensing Imagery. pages 1–10, 2018. ↰ pages 9 and 13

[63] R. Kemker, C. Salvaggio, and C. Kanan. High-Resolution Multispectral Dataset for Semantic Segmentation. 2017. ↰ page 15

[64] M. Khamis, W. Gomaa, and B. Galal. Deep learning is competing random forest in computational docking. aug 2016. ↰ page 3

[65] M. A. Khamis. DEEP LEARNING IS COMPETING RANDOM FOREST IN COMPUTATIONAL DOCKING. Technical report, 2016. ↰ page 3

[66] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. *Int. Conf. Learn. Represent. 2015*, pages 1–15, 2014. ↰ page 49

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Technical report. ↰ page 40

[68] H. Kubo, S. Jayasuriya, T. Iwaguchi, T. Funatomi, Y. Mukaigawa, and S. G. Narasimhan. Acquiring and characterizing plane-to-ray indirect light transport. In *Comput. Photogr. (ICCP), 2018 IEEE Int. Conf.*, pages 1–10. IEEE, 2018. ↰ page 5

[69] H. Kwon and Y.-W. Tai. RGB-Guided Hyperspectral Image Upsampling. In *Proc. Int. Conf. Comput. Vis.*, pages 307–315, 2015. ↰ page 21

[70] A. Lam, A. Subpa-Asa, I. Sato, T. Okabe, and Y. Sato. Spectral imaging using basis lights. *BMVC 2013 - Electron. Proc. Br. Mach. Vis. Conf. 2013*, pages 1–11, 2013. ↰ page 9

[71] C. Lanaras, E. Baltsavias, and K. Schindler. Hyperspectral super-resolution by coupled spectral unmixing. *Proc. IEEE Int. Conf. Comput. Vis.*, 2015 Inter:3586–3594, 2015. ↰ page 17

[72] C. Lanaras, E. Baltsavias, and K. Schindler. Hyperspectral SuperResolution by Coupled Spectral Unmixing. In *Proc. Int. Conf. Comput. Vis.*, pages 3586–3594, 2015. ↰ page 21

[73] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990. ↰ pages 17, 18, and 40

[74] A. Levin and Y. Weiss. User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1647–1654, 2007. ↰ page 20

[75] Y. Li and M. S. Brown. Exploiting Reflection Change for Automatic Reflection Removal. In *2013 IEEE Int. Conf. Comput. Vis.*, pages 2432–2439, 2013. ↰ page 20

[76] Y. Li, H. Lu, J. Li, X. Li, Y. Li, S. S. C. &. E. Engineering, and undefined 2016. Underwater image de-scattering and classification by deep neural network. *Elsevier*. ↰ page 3

[77] Y. Li, W. Xie, and H. Li. Hyperspectral image reconstruction by deep convolutional neural network for classification. *Pattern Recognit.*, 63:371–383, 2017. ↰ page 22

[78] Z. Li and Z. Shi. Deep Residual Learning and PDEs on Manifold. pages 1–9, 2017. ↰ page 42

[79] X. Liang, L. Lee, W. Dai, E. P. Xing, T. Takatani, T. Aoto, Y. Mukaigawa, and X. Zhang, Kaibing and Gao, Xinbo and Tao, Dacheng and Li. Dual Motion GAN for Future-Flow Embedded Video Prediction. *Proc. IEEE Int. Conf. Comput. Vis.*, 2017-Octob:1762–1770, 2017. ↰ page 25

[80] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz. Exploring features in a Bayesian framework for material recognition. In *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 239–246, 2010. ↰ page 6

[81] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning End-to-end Multimodal Sensor Policies for Autonomous Navigation. Technical report. ↰ page 3

[82] Y. Liu, Q. Ren, J. Geng, M. Ding, and J. Li. Efficient patch-wise semantic segmentation for large-scale remote sensing images. *Sensors (Switzerland)*, 18(10):1–16, 2018. ↰ page 79

[83] G. Loukas, T. Vuong, R. Heartfield, G. S. I. . . . , and undefined 2017. Cloud-based cyber-physical intrusion detection for vehicles using deep learning. *ieeexplore.ieee.org*. ↰ page 3

[84] Lu Guolan and B. Fei. Medical Hyperspectral Imaging: A Review. *J. Biomed. Opt.*, 2016. ↰ pages 2 and 7

[85] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. *Int. Geosci. Remote Sens. Symp.*, 2015-Novem:4959–4962, 2015. ↰ pages 9 and 22

[86] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Comput. Vis. (ICCV), 2017 IEEE Int. Conf.*, pages 2813–2821. IEEE, 2017. ↰ page 33

[87] J. Marçais, J.-R. de Dreuzy, and C. Author. Prospective Interest of Deep Learning for Hydrological Inference. *Prospect. Interes. Deep Learn. Hydrol. Infer-ence. Groundw.*, 55(5), 2017. ↰ page 3

[88] H. Matsuoka, Y. Kosai, M. Saito, N. Takeyama, and H. Suto. Single-cell viability assessment with a novel spectro-imaging system. *J. Biotechnol.*, 94(3):299–308, 2002. ↰ pages 7 and 21

[89] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. ↰ page 17

[90] L. Mo, F. Li, Y. Zhu, A. H. . I. International, and undefined 2016. Human physical activity recognition based on computer vision with deep learning model. *ieeexplore.ieee.org*. ↰ page 3

[91] N. J. Morris and K. N. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–8, 2007. ↰ page 6

[92] Y. Mukaigawa, K. Suzuki, and Y. Yagi. Analysis of Subsurface Scattering Based on Dipole Approximation. *Inf. Media Technol.*, 4(4):951–961, 2009. ↰ page 20

[93] Y. Mukaigawa, Y. Yagi, and R. Raskar. Analysis of light transport in scattering media. In *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 153–160, 2010. ↰ page 20

[94] A. Munoz, J. I. Echevarria, F. J. Seron, J. Lopez-Moreno, M. Glencross, and D. Gutierrez. BSSRDF Estimation from Single Images. *Comput. Graph. Forum*, 30(2):455–464, 2011. ↰ page 20

[95] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph.*, 25:935, 2006. ↰ pages xii, 2, 27, 28, 29, and 30

[96] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM Trans. Graph. (also Proc. ACM SIGGRAPH)*, 2006. ↰ pages 5, 19, 30, and 36

[97] R. M. H. Nguyen, D. K. Prasad, and M. S. Brown. Training-based spectral reconstruction from a single RGB image. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 8695 LNCS(PART 7):186–201, 2014. ↰ pages 7, 21, 22, 49, 52, 54, 56, 57, 61, and 90

[98] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, D. Samaras, and S. Brook. Shadow Detection with Conditional Generative Adversarial Networks. *Iccv 2017*, pages 4510–4518, 2017. ↰ page 25

[99] S. Nie, L. Gu, Y. Zheng, A. Lam, N. Ono, and I. Sato. Deeply Learned Filter Response Functions for Hyperspectral Reconstruction. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4767–4776, 2018. ↰ page 26

[100] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv Prepr. arXiv1610.09585*, 2016. ↰ page 33

[101] A. Osokin, A. Chessel, R. E. C. Salas, and F. Vaggi. GANs for Biological Image Synthesis. 2017. ↰ page 25

[102] M. O'Toole, S. Achar, S. G. Narasimhan, and K. N. Kutulakos. Homogeneous codes for energy-efficient illumination and imaging. *ACM Trans. Graph.*, 34(4):35, 2015. ↰ page 5

[103] M. O'Toole, R. Raskar, and K. N. Kutulakos. Primal-dual coding to probe light transport. *ACM Trans. Graph.*, 31(4):31–39, 2012. ↰ page 5

[104] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, et al. Deepid-net: Object detection with deformable part based convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1320–1334, 2016. ↰ page 18

[105] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016. ↰ page 40

[106] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2016. ↰ page 33

[107] N. Phan, D. Dou, H. Wang, D. Kil, B. P. I. Sciences, and undefined 2017. Ontology-based deep learning for human behavior prediction with explanations in health social networks. *Elsevier*. ↰ page 3

[108] T. Ping-Sing and M. Shah. Shape from shading using linear approximation. *Image Vis. Comput.*, 12(8):487–498, 1994. ↰ page 31

[109] S. Prasad, L. B. I. G. Sensing, Remote, and undefined 2008. Limitations of principal components analysis for hyperspectral target recognition. *ieeexplore.ieee.org*. ↰ page 15

[110] D. Reddy, R. Ramamoorthi, and B. Curless. Frequency-Space Decomposition and Acquisition of Light Transport under Spatially Varying Illumination. *Eur. Conf. Comput. Vis.*, 2012. ↰ page 20

[111] A. Robles-Kelly. Single Image Spectral Reconstruction for Multimedia Applications. In *23rd ACM Int. Conf. Multimed.*, pages 251–260, 2015. ↰ pages 7, 21, 22, and 90

[112] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Med. Image Comput. Comput. Interv.*, 2015. ↰ pages 40 and 77

[113] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri. HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.*, pages 1–5, 2019. ↰ pages 22 and 81

[114] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and Others. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. ↰ pages 2 and 34

[115] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Adv. Neural Inf. Process. Syst.*, number Nips, pages 2234–2242, 2016. ↰ pages 33 and 34

[116] V. Saragadam and A. C. Sankaranarayanan. Programmable Spectrometry – Per-pixel Classification of Materials using Learned Spectral Filters. 2019. ↰ pages 9, 13, and 14

[117] E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017. ↰ pages 40, 41, and 77

[118] Z. Shi. HSCNN + : Advanced CNN-Based Hyperspectral Recovery from RGB Images. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work.*, 2018. ↰ pages 40, 42, 58, and 73

[119] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014. ↰ pages 2 and 77

[120] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* International Conference on Learning Representations, ICLR, 2015. ↰ page 40

[121] J. Song. Binary Generative Adversarial Networks for Image Retrieval. pages 394–401, 2017. ↰ page 70

[122] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway Networks. *arXiv1505.00387 [cs]*, 2015. ↰ page 40

[123] R. Stewart and S. Ermon. Label-Free Supervision of Neural Networks with Physics and Domain Knowledge. 1(1), 2016. ↰ pages 2 and 26

[124] A. Subpa-asa, Y. Fu, Y. Zheng, T. Amano, and I. Sato. Direct and Global Component Separation from a Single Image Using Basis Representation. In S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, editors, *Comput. Vis. – ACCV 2016*, pages 99–114, Cham, 2017. Springer International Publishing. ↰ pages 5 and 20

[125] A. Sun, B. Scanlon, Z. Z. W. R. . . . , and undefined 2019. Combining Physically Based Modeling and Deep Learning for Fusing GRACE Satellite Data: Can We Learn From Mismatch? *Wiley Online Libr.* ↰ page 3

[126] A. Y. Sun, B. R. Scanlon, Z. Zhang, D. Walling, S. N. Bhanja, A. Mukherjee, and Z. Zhong. Combining Physically Based Modeling and Deep Learning for Fusing GRACE Satellite Data: Can We Learn From Mismatch? *Water Resour. Res.*, 2019. ↰ page 3

[127] S. H. Sun, S. P. Fan, and Y. C. F. Wang. Exploiting image structural similarity for single image rain removal. *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pages 4482–4486, 2014. ↰ page 33

[128] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *31st AAAI Conf. Artif. Intell. AAAI 2017*, 2017. ↰ page 42

[129] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pages 2818–2826, 2016. ↰ page 34

[130] K. Tanaka, Y. Mukaigawa, H. Kubo, Y. Matsushita, and Y. Yagi. Recovering Inner Slices of Layered Translucent Objects by Multi-Frequency Illumination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):746–757, 2017. ↰ page 20

[131] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson. SVM- and MRF-based method for accurate classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.*, 7(4):736–740, 2010. ↰ pages 9, 13, 14, and 22

[132] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 61(3):611–622, 1999. ↰ page 15

[133] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. ↰ page 18

[134] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-Based Color Constancy. *IEEE Trans. Image Process.*, 16(9):2207–2214, 2007. ↰ page 6

[135] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. pages 1–15, 2016. ↰ page 2

[136] A. Wagadarikar, N. Pitsianis, X. Sun, and B. David. Video rate spectral imaging using a coded aperture snapshot spectral imager. *Opt. Express*, 17(8):6368–6388, 2009. ↰ pages 7 and 21

[137] R. Wan, B. Shi, L. Y. Duan, A. H. Tan, and A. C. Kot. Benchmarking Single-Image Reflection Removal Algorithms. In *2017 IEEE Int. Conf. Comput. Vis.*, pages 3942–3950, 2017. ↰ page 20

[138] T. Wang, C. Wen, H. Wang, F. G. C. . . . , and undefined 2017. Deep learning for wireless physical layer: Opportunities and challenges. *ieeexplore.ieee.org*. ↰ page 3

[139] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. ↰ page 33

[140] J. Wei, G. M. . J. W. on Cyber-Physical, and undefined 2016. A deep learning-based cyber-physical strategy to mitigate false data injection attack in smart grids. *ieeexplore.ieee.org*. ↰ page 3

[141] Y. Wei, Y. Zhou, and H. Li. Spectral-spatial response for hyperspectral image classification. *Remote Sens.*, 9(3):1–31, 2017. ↰ pages 9 and 15

[142] D. H. Wolpert. The Supervised Learning No-Free-Lunch Theorems. ↰ page 2

[143] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. Technical report, 1996. ↰ page 2

[144] D. Wu, M. O'Toole, A. Velten, A. Agrawal, and R. Raskar. Decomposing global light transport using time of flight imaging. In *2012 IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 366–373, 2012. ↰ page 20

[145] D. Wu, A. Velten, M. O'Toole, B. Masia, A. Agrawal, Q. Dai, and R. Raskar. Decomposing global light transport using time of flight imaging. In *Int. J. Comput. Vis.*, volume 107, pages 123–138, 2014. ↰ page 20

[146] F. Wu, C. Chen, D. Liu, Z. Shi, Z. Xiong, Z.-J. Zha, C. Chen, Z. Xiong, D. Liu, Z.-J. Zha, and F. Wu. Deep residual attention network for spectral image super-resolution. In *Eur. Conf. Comput. Vis.*, pages 214–229. Springer, 2018. ↰ page 42

[147] J. Wu, I. Yildirim, J. J. Lim, W. T. Freeman, and J. B. T. Bcs. Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning. Technical report. ↰ page 3

[148] H. Xiong, B. Alipanahi, L. Lee, H. B. . . . , and undefined 2015. The human splicing code reveals new insights into the genetic determinants of disease. *science.sciencemag.org*. ↰ page 2

[149] W. Xiong and B. Funt. Independent Component Analysis and Nonnegative Linear Model Analysis of Illuminant and Reflectance Spectra. *Proc. Tenth Congr. Int. Colour Assoc.*, 2(1):503–506, 2005. ↰ page 15

[150] Z. Xiong, Z. Shi, H. Li, L. Wang, D. Liu, and F. Wu. HSCNN: CNN-Based Hyperspectral Image Recovery from Spectrally Undersampled Projections. In *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017*, volume 2018-Janua, pages 518–525, 2018. ↰ page 58

[151] Z. W. Xiong, Z. Shi, H. Q. Li, L. Z. Wang, D. Liu, and F. Wu. HSCNN: CNN-Based Hyperspectral Image Recovery From Spectrally Undersampled Projections. *IEEE Int. Conf. Comput. Vis.*, pages 518–525, 2017. ↰ pages 7 and 22

[152] J. Yang, H. Li, Y. Dai, and R. T. Tan. Robust Optical Flow Estimation of Double-Layer Images under Transparency or Reflection. In *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1410–1419, 2016. ↰ page 20

[153] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE Trans. image Process.*, 19(9):2241–2253, 2010. ↰ pages xiii, 31, and 36

[154] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized Assorted Pixel Camera: Postcapture Control of Resolution, Dynamic Range, and Spectrum. *IEEE Trans. IMAGE Process.*, 19(9), 2010. ↰ pages xiii, xiv, xvii, 46, 48, 49, 50, 51, 53, 54, 55, 60, 61, 64, and 65

[155] N. Yokoya, T. Yairi, and A. Iwasaki. Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Trans. Geosci. Remote Sens.*, 50(2):528–537, 2012. ↰ page 16

[156] D. Zhang, J. Lin, Q. Peng, D. Wang, T. Y. J. of . . . , and undefined 2018. Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm. *Elsevier*. ↰ page 3

[157] H. Zhang, L. Zhang, and H. Shen. A super-resolution reconstruction algorithm for hyperspectral images. *Signal Processing*, 92(9):2082–2096, 2012. ↰ page 15

[158] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Eur. Conf. Comput. Vis.*, volume 9907 LNCS, pages 649–666. Springer, 2016. ↰ page 33

[159] Y. Zhang, C. P. Huynh, N. Habili, and K. N. Ngan. Material segmentation in hyperspectral images with minimal region perimeters. In *Proc. -*

Int. Conf. Image Process. ICIP, volume 2016-Augus, pages 834–838, 2016. ↰ pages 9 and 15

[160] Y. Zhang, K. N. Ngan, C. P. Huynh, and N. Habili. Learning Deep Spatial-Spectral Features for Material Segmentation in Hyperspectral Images. *DICTA 2017 - 2017 Int. Conf. Digit. Image Comput. Tech. Appl.*, 2017-Decem:1–7, 2017. ↰ page 9

[161] Z. Zhang, C. Zhang, and K. P. Lam. A Deep Reinforcement Learning Method for Model-based Optimal Control of HVAC Systems. Technical report. ↰ page 3

[162] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. Deep Learning and Its Applications to Machine Health Monitoring: A Survey. Technical report. ↰ page 3

[163] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. *Proc. IEEE Int. Conf. Comput. Vis.*, 2015 Inter:1529–1537, 2015. ↰ page 9

[164] Y. Zheng, I. Sato, and Y. Sato. Illumination and reflectance spectra separation of a hyperspectral image meets low-rank matrix factorization. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 07-12-June:1779–1787, 2015. ↰ pages 15 and 16

[165] Z. Zhong and J. Li. Generative Adversarial Networks and Probabilistic Graph Models for Hyperspectral Image Classification. pages 1–18, 2018. ↰ page 22

[166] Z. Zhong, J. Li, Z. Luo, and M. Chapman. Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.*, 56(2):847–858, 2018. ↰ pages 9 and 42

[167] X. Zhou, Q. Wan, W. Zhang, X. Xue, Y. Wei, Z. Wei, X. Xue, and Y. Wei. Model-based Deep Hand Pose Estimation, 2016. ↰ page 3

[168] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. 2017. ↰ pages 25 and 26

[169] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. 2017. ↰ page 40

[170] L. Zhu, Y. Chen, P. Ghamisi, and J. A. J. A. Benediktsson. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.*, 56(9):5046–5063, 2018. ↰ pages 22 and 78

[171] X. X. Zhu and R. Bamler. A Sparse Image Fusion Algorithm With Application to Pan-Sharpening. *IEEE Trans. Geosci. Remote Sens.*, 51(5):2827–2836, 2013. ↰ page 21