DEEP LEARNING-BASED UNROLLED RECONSTRUCTION METHODS FOR COMPUTATIONAL IMAGING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CAN DENİZ BEZEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

DEEP LEARNING-BASED UNROLLED RECONSTRUCTION METHODS FOR COMPUTATIONAL IMAGING

submitted by **CAN DENİZ BEZEK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Sevinç Figen Öktem
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Sevinç Figen Öktem
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Tolga Çukur
Electrical and Electronics Engineering, Bilkent University _____

Date: 08.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Can Deniz Bezek

Signature :

ABSTRACT

DEEP LEARNING-BASED UNROLLED RECONSTRUCTION METHODS FOR COMPUTATIONAL IMAGING

Bezek, Can Deniz
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Sevinç Figen Öktem

September 2021, 98 pages

Computational imaging is the process of forming images from indirect measurements using computation. In this thesis, we develop deep learning-based unrolled reconstruction methods for various computational imaging modalities. Firstly, we develop two deep learning-based reconstruction methods for diffractive multi-spectral imaging. The first approach is based on plug-and-play regularization with deep denoisers whereas the second one is an end-to-end learned reconstruction based on unrolling. Secondly, we consider general multidimensional imaging systems whose measurements involve convolution and superposition. We formulate a realistic truncated image formation model and develop a deep learning-based unrolled reconstruction method to solve the associated inverse problem. The performance is illustrated for single-frame deconvolution and diffractive multi-spectral imaging applications. Thirdly, we extend the deep learning-based unrolled reconstruction methods to a compressive spectral imaging modality by considering both the conventional and truncated image formation models. The performance of all the developed methods is comparatively evaluated for sample applications using different convolutional neural network (CNN) architectures, including U-Net. Results illustrate the superior per-

formance of the developed unrolled learned reconstruction methods over purely analytical methods for all simulation settings. We also demonstrate the generalization capability of the developed unrolled reconstruction methods through extensive simulations and perform model mismatch analyses to illustrate the improvement gained with the truncated image formation model.

Keywords: computational imaging, spectral imaging, inverse problems, image reconstruction, deep learning, algorithm unrolling, truncated convolution

ÖZ

HESAPLAMALI GÖRÜNTÜLEME İÇİN DERİN ÖĞRENME TABANLI GERİÇATIM YÖNTEMLERİ

Bezek, Can Deniz
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sevinç Figen Öktem

Eylül 2021 , 98 sayfa

Hesaplamalı görüntüleme, görüntülerin hesaplama kullanılarak dolaylı ölçümlerden elde edilmesidir. Bu tezde, çeşitli hesaplamalı görüntüleme metotları için derin öğrenme tabanlı geriçatım yöntemleri geliştirilmektedir. İlk olarak, kırınımlı çoklu spektral görüntüleme için derin öğrenme tabanlı iki geriçatım yöntemi geliştirilmektedir. İlk yaklaşım, derin gürültüsüzleştiriciler ile koy ve oynat düzenlileştirmesine dayanan özyineli bir yöntem ve ikinci yaklaşım, özyineli yöntemin iterasyonlar boyunca açılmasına dayanan baştan sonra öğrenilmiş bir yöntemdir. İkinci olarak, ölçümleri evrişim ve üstdüşüm içeren genel çok boyutlu görüntüleme sistemleri dikkate alınmaktadır. Gerçekçi kırpılmış görüntüleme modeli formüle edilmektedir ve ilgili ters problemi çözmek için derin öğrenme tabanlı geriçatım yöntemi geliştirilmektedir. Yöntemin performansı, tek-çerçeve ters evrişim ve kırınımlı çoklu spektral görüntüleme uygulamaları için gösterilmektedir. Üçüncü olarak, derin öğrenme tabanlı geriçatım yöntemleri hem klasik hem de kırpılmış görüntüleme modelleri kullanılarak sıkıştırılmış spektral görüntüleme metoduna uygulanmaktadır. Geliştirilen bütün yöntemlerin performansları, U-Net'i de içeren farklı evrişimsel sinir ağı mimarileri kullanılarak

örnek uygulamalar için karşılaştırmalı olarak değerlendirilmektedir. Sonuçlar, tüm benzetim durumlarında, geliştirilen baştan sona öğrenilmiş yöntemin tamamen analitik olan yöntemlere karşı olan üstünlüğünü göstermektedir. Ayrıca, kapsamlı benzetimlerle, geliştirilen baştan sona öğrenilmiş yöntemin genelleştirme yeteneği ve model uyumsuzluğu analizleriyle, kırpılmış görüntüleme modeli kullanılarak elde edilen iyileşmeler gösterilmektedir.

Anahtar Kelimeler: hesaplamalı görüntüleme, spektral görüntüleme, ters problemler, görüntü geriçatımı, derin öğrenme, algoritma açma, kırpılmış evrişim

To my lovely family

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Figen S. Öktem for her guidance, patience, motivation, and continuous support. I have learned a lot from her about scientific research, writing, and critical thinking. She has been not just a supervisor but also a mentor. I could not imagine having a better supervisor for my master's studies.

I am grateful to members of my thesis committee, Prof. Dr. Aydın Alatan, Assoc. Prof. Dr. Yeşim Serinağoğlu Doğrusöz, Assoc. Prof. Dr. Elif Vural, and Assoc. Prof. Dr. Tolga Çukur, for showing interest in my studies, reviewing and commenting on my thesis.

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under grants 117E160 (3501 Research Program) and 120E505. I also would like to thank TUBITAK for their support under 2210/A scholarship program.

I would like to express my deepest gratitude to my family. They always loved, supported, cared for, and believed in me. I am very lucky to have the best siblings, my brother Serhat and my sister Diren, in my life. I would also like to thank my father İsmet, and my mother Hüsna, the strongest woman I know, for all the sacrifices they made for us.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xxi

CHAPTERS

1 INTRODUCTION 1

2 EFFICIENT DEEP LEARNING BASED RECONSTRUCTION FOR DIFFRACTIVE MULTI-SPECTRAL IMAGING 7

2.1 Introduction 7

2.2 Forward Model 9

2.3 Inverse Problem 11

2.4 Developed Reconstruction Methods 11

2.4.1 Plug-and-Play Regularization 13

2.4.2 End-to-End Learned Reconstruction with Unrolling 13

2.5 Numerical Results 16

2.5.1 Performance Comparison of Different Methods 18
2.5.2 SNR Analysis 23
2.5.3 Model Mismatch Analysis 25
2.6 Conclusion 28
3 DEEP LEARNING-BASED UNROLLED RECONSTRUCTION FOR IMAGING PROBLEMS INVOLVING TRUNCATED CONVOLUTIONS 31
3.1 Introduction 31
3.2 Forward Model 33
3.2.1 Single-Frame Deconvolution 34
3.2.2 Snapshot Case 35
3.2.3 Multi-Frame Deconvolution 36
3.2.4 Multiple Measurement Multiple Image Case 37
3.3 Inverse Problem 38
3.4 Developed Reconstruction Method 39
3.5 Numerical Results 42
3.5.1 Performance Analysis for the Multiple Measurement Multiple Image Case 42
3.5.2 Model Mismatch Analysis for the Multiple Measurement Multiple Image Case 48
3.5.3 Performance Analysis for Single-Frame Deconvolution 50
3.6 Discussion 56
3.7 Conclusion 60
4 DEEP LEARNING-BASED UNROLLED RECONSTRUCTION FOR COMPRESSIVE SPECTRAL IMAGING WITH DIFFRACTIVE LENSES . . . 63
4.1 Introduction 63

4.2 Forward Model 65
4.2.1 Simplified Model with Zero End Conditions 66
4.2.2 Truncated Model 67
4.3 Inverse Problem 69
4.4 Developed Reconstruction Methods 69
4.4.1 Method for the Simplified Model 71
4.4.2 Method for the Truncated Model 72
4.5 Numerical Results 74
4.5.1 Results for the Simplified Model 74
4.5.2 Results for the Truncated Model 79
4.5.3 Model Mismatch Analysis 83
4.6 Conclusion 84
5 CONCLUSION 87
REFERENCES 91

LIST OF TABLES

TABLES

Table 2.1 Average PSNR (dB) and SSIM values for TV and deep learning-based reconstructions at 25 dB input SNR for different test datasets (with best values shown in bold). 19

Table 2.2 Number of Trainable Parameters for Different Methods. 23

Table 2.3 PSNR (dB) and SSIM values for TV and learned reconstruction methods at different input SNRs and for different spectral datasets (with best values shown in bold). 24

Table 2.4 Average PSNR (dB) and SSIM values for Unrolled-II at different input SNRs for different spectrums. 28

Table 3.1 Average PSNR (dB), SSIM and runtime (second) values for TV and learned reconstructions for different spectral test data (with best values shown in bold). 45

Table 3.2 Average PSNR (dB) and SSIM values for unrolled reconstruction method with and without model mismatch for different spectral test data. . 50

Table 3.3 Average PSNR (dB) and SSIM values for TV and learned reconstructions and for different blurs (with best values shown in bold). 52

Table 4.1 Comparison of average PSNR (dB), SSIM and SAM values for different measurement SNRs (with best values shown in bold). 75

Table 4.2 Comparison of average runtime (second) for different measurement SNRs. 76

Table 4.3 Comparison of PSNR (dB), SSIM and SAM values for different test images (with best values shown in bold). 76

Table 4.4 Comparison of average PSNR (dB), SSIM and SAM values (with best values shown in bold). 79

Table 4.5 Comparison of average runtime (second) for different measurement SNRs. 80

Table 4.6 Comparison of PSNR (dB), SSIM and SAM values for different test images using truncated model (with best values shown in bold). 80

Table 4.7 Average PSNR (dB), SSIM, and SAM values for learned reconstruction with and without mismatch. 83

LIST OF FIGURES

FIGURES

Figure 2.1 Schematic view of the diffractive lens based spectral imaging modality. 8

Figure 2.2 The developed plug-and-play regularization method. 13

Figure 2.3 The developed end-to-end learned reconstruction with unrolling. 13

Figure 2.4 CNN architectures. Network-I (U-Net) (a) (figure adapted from [51]), Network-II (b) (figure adapted from [39]). conv, convolutional filter; BN, batch normalization; ReLU, rectified linear unit; up-conv 2., transposed convolutional layer with a stride of two. 15

Figure 2.5 Measured intensities for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (f), and λ_3 (k); the contribution of the 1st, 2nd, and 3rd spectral images to the measurement with the PS design focusing wavelength λ_1 (b)-(d), λ_2 (g)-(i), λ_3 (l)-(n); sampled and zoomed PSFs of the system with the PS design focusing solar emission line at the central wavelength λ_1 for the 1st, 2nd, and 3rd spectral images, (e), (j), and (o), respectively. 17

Figure 2.6 Top to bottom: diffraction-limited images, TV, Unrolled-I, Unrolled-II, Plug&Play-I, and Plug&Play-II reconstructions for Dataset 2 (with false-color representations on the right). 20

Figure 2.7 Top to bottom: diffraction-limited images, TV, Unrolled-I, Unrolled-II, Plug&Play-I, and Plug&Play-II reconstructions for Dataset 5 (with false-color representations on the right). 21

Figure 2.8 PSNR (a) and SSIM (b) histograms for TV, Unrolled-I, and Unrolled-II reconstructions. Red, green, and blue vertical lines show the mean values, respectively. 22

Figure 2.9 Continuous spectrum consisting of three Gaussian spectral lines with central wavelengths 33.28, 33.42, and 33.54 nm. 26

Figure 2.10 Measured intensities for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (b), and λ_3 (c); with the largest contributions from 11 sampled spectral points. . . . 26

Figure 2.11 Diffraction-limited original images (top), reconstructed spectral images from continuous spectrum measurements with learned-reconstruction method (bottom) for dataset 2 at 25 dB SNR. For each case, false-color representations are also provided on the right. 27

Figure 3.1 Measured intensities using truncated image formation model for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (h), and λ_3 (o); the contribution of the 1st, 2nd, and 3rd spectral images to the measurement with the PS design focusing solar emission line at the central wavelength λ_1 (b)-(d), λ_2 (i)-(k), λ_3 (p)-(r); sampled and zoomed PSFs of the system for the 1st, 2nd, and 3rd spectral images with the PS design focusing solar emission line at the central wavelength λ_1 (e)-(g), λ_2 (l)-(n), λ_3 (s)-(u) , respectively. . 43

Figure 3.2 Top to bottom: diffraction-limited images, deep learning-based unrolled reconstruction, TV reconstruction for test data 2 (with false-color representations on the right). 46

Figure 3.3 Top to bottom: diffraction-limited images, deep learning-based unrolled reconstruction, TV reconstruction for test data 5 (with false-color representations on the right). 47

Figure 3.4 PSNR (a) and SSIM (b) histograms of TV and learned reconstructions. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively. 48

Figure 3.5 Top to bottom: diffraction-limited images, reconstructions without model mismatch, reconstructions with model mismatch for test data 4 (with false-color representations on the right). 49

Figure 3.6 Sample image (a), sample measurements at 25 dB SNR for the Gaussian blur with a variance of 2 (b), Gaussian blur with a variance of 8 (c), and uniform blur (d). All blurs have 9×9 support. Sampled and zoomed versions of the Gaussian blur with a variance of 2 (e), Gaussian blur with a variance of 8 (f), and uniform blur (g); magnitudes of the Fourier transforms of the Gaussian blur with a variance of 2 (h), Gaussian blur with a variance of 8 (i), and uniform blur (j). 51

Figure 3.7 PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for Gaussian blur with a variance of 2. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively. 53

Figure 3.8 PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for Gaussian blur with a variance of 8. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively. 53

Figure 3.9 PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for uniform blur. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively. 54

Figure 3.10 Top to bottom: Sample image (a); TV reconstructions for the Gaussian blur with a variance of 2 (b), Gaussian blur with a variance of 8 (c), uniform blur (d); learned reconstructions for the Gaussian blur with a variance of 2 (e), Gaussian blur with a variance of 8 (f), uniform blur (g). Reconstruction PSNR (dB) and SSIM values from (b) to (g) are respectively $24.67/0.77, 22.60/0.61, 23.35/0.66, 25.03/0.80, 23.05/0.67,$ and $23.71/0.71$. 55

Figure 4.1 CSID System. 65

Figure 4.2 3D U-Net architecture. conv, convolutional filter; ReLU, rectified linear unit; up-conv 2., transposed convolutional layer with a stride of two. This figure is adapted from [70]. 71

Figure 4.3 Architecture for the developed deep learning-based unrolled reconstruction method. 72

Figure 4.4 Reconstructed spectral images from compressive measurements for test image 1 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 77

Figure 4.5 Reconstructed spectral images from compressive measurements for test image 2 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 77

Figure 4.6 Reconstructed spectral images from compressive measurements for test image 3 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 77

Figure 4.7 PSNR (a) and SSIM (b), and SAM (c) histograms of analytical and learned reconstructions using the simplified model. Red and green vertical lines show the mean values for analytical and learned reconstructions, respectively. 78

Figure 4.8 Reconstructed spectral images from compressive measurements for test image 1 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 81

Figure 4.9 Reconstructed spectral images from compressive measurements for test image 2 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 81

Figure 4.10 Reconstructed spectral images from compressive measurements for test image 3 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown). 81

Figure 4.11 PSNR (a) and SSIM (b), and SAM (c) histograms of analytical and learned reconstructions using the truncated model. Red and green vertical lines show the mean values for analytical and learned reconstructions, respectively. 82

LIST OF ABBREVIATIONS

1D	1 Dimensional
2D	2 Dimensional
3D	3 Dimensional
CNN	Convolutional Neural Network
CSID	Compressive Spectral Imaging with Diffractive Lenses
DFT	Discrete Fourier Transform
EUV	Extreme Ultra-Violet
FFT	Fast Fourier Transform
FOV	Field of View
GPU	Graphics Processing Unit
HQS	Half Quadratic Splitting
NASA	National Aeronautics and Space Administration
PS	Photon Sieve
PSF	Point Spread Function
PSNR	Peak Signal-to-Noise Ratio
SAM	Spectral Angle Mapper
SD	Steepest Descent
SDO	Solar Dynamics Observatory
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity Index Metric
TV	Total Variation

CHAPTER 1

INTRODUCTION

Imaging has been a prominent field with widespread applications in physical and life sciences [1, 2]. Images acquired can be either two-dimensional (2D) or higher dimensional (multidimensional). In conventional 2D imaging, the aim is to measure the distribution of light over two-dimensional space. On the other hand, in multidimensional imaging, the goal is to form the image of a scene in more than two dimensions, to acquire much richer information about the chemical, physical, and biological properties of the targeted scene [1–5]. The additional dimensions can contain the spatial, spectral, and/or temporal distribution of the light (or the electromagnetic field). Therefore, the images are hypercubes involving spatial coordinates (x, y, z), wavelength (λ), and/or time (t).

Capturing the multidimensional image data with the conventional 2D detectors is challenging. Traditional imaging techniques form the high-dimensional data from a series of 2D measurements. For example, in spectral imaging, three-dimensional spectral datacube is obtained either by scanning the scene spectrally through a series of wavelength filters or by scanning the scene spatially through a spectrometer with a long slit. These methods suffer from low signal-to-noise ratio (SNR), long acquisition time, and temporal artifacts for dynamic scenes [6–8]. Moreover, attainable spatial, spectral, and temporal resolutions are inherently limited by the availability and cost of the optical elements involved. Computational imaging techniques overcome these limitations by passing some of the burden to a reconstruction algorithm [2, 4, 5, 7, 9–13]. These approaches reconstruct the desired image data from superimposed measurements by solving an inverse problem.

In various multidimensional computational imaging systems, the relationship be-

tween the measurements (sensor data) and the unknown image data can be characterized by the sum of multiple convolutions. These imaging systems involve an image reconstruction task from superimposed convolutions. Such image reconstruction problems are encountered in various computational imaging modalities such as computational photography, wide-field astronomical imaging, three-dimensional (volumetric) microscopy, spectral imaging, ultrafast imaging, radio interferometric imaging, magnetic resonance imaging, and ultrasound imaging [5,7,9,14–25].

Recently deep learning-based reconstruction methods have been successfully applied to various inverse problems in imaging [26,27]. These methods are generally based on either direct inversion [28–30], plug-and-play regularization [31–33], or unrolling [34–39]. Direct inversion methods aim to find a direct mapping from the measurements to the desired image solely using training data. Therefore, these methods do not use the forward model of the physical system. The key idea in plug-and-play regularization and unrolling-based approaches is to replace the analytical prior with a deep prior. Hence, these approaches incorporate deep learning into the reconstructions while also using the forward model.

In this thesis, we develop novel deep learning-based unrolled reconstruction methods for various computational imaging modalities. We focus on diffractive multi-spectral imaging in Chapter 2. In Chapter 3, we address the limitations of the conventional image formation model used in Chapter 2. For this, we focus on general multidimensional imaging systems whose measurements involve convolution and superposition. We formulate a realistic truncated image formation model and develop a deep learning-based unrolled reconstruction method to solve the associated inverse problem. The performance is illustrated for single-frame deconvolution and diffractive multi-spectral imaging applications. Then, we extend the deep learning-based unrolled reconstruction methods to a compressive spectral imaging modality by considering both the conventional and truncated image formation models. In each chapter, we first formulate the forward model and the inverse problem for the imaging system of interest. Then, we develop deep learning-based unrolled reconstruction methods using the half quadratic splitting (HQS) method. The performance of all the developed methods is comparatively evaluated for sample applications using different CNN architectures, including U-Net. Results illustrate the superior performance of the de-

veloped unrolled learned reconstruction methods over purely analytical methods for all simulation settings. We also demonstrate the generalization capability of the developed unrolled reconstruction methods through extensive simulations and perform model mismatch analyses to illustrate the improvement gained with the truncated image formation model.

In Chapter 2, we develop deep learning-based reconstruction methods for a diffractive multi-spectral imaging system [36] based on plug-and-play regularization and unrolling. This imaging system offers higher spatial and spectral resolutions than the traditional multi-spectral imagers [7, 36, 40, 41] but additionally requires an inverse problem to be solved for image reconstruction. To develop deep learning-based reconstruction methods, we first formulate the inverse problem as an optimization problem and then split it into two sub-problems using the HQS method. The sub-problem containing the data fidelity term has a closed-form solution and is solved efficiently using fast Fourier transforms (FFTs). The second sub-problem corresponds to a denoising problem. In the plug-and-play approach, to solve this denoising problem, we train a deep CNN denoiser with noisy spectral images. Spectral images are then reconstructed iteratively by alternating between these two sub-problems. In the second approach, we unroll the algorithm into an end-to-end trainable network. Spectral images are then simply reconstructed by feeding this network with the measurements. For a sample application in astrophysical imaging, we comparatively evaluate the performance of the developed methods using different CNN architectures and loss functions. Results illustrate that unrolled reconstruction with U-Net provides a significant performance improvement over the other learned and analytical reconstruction methods used in the comparison. We also evaluate the performance of the unrolled method under different signal-to-noise ratios (SNRs). Results show that the unrolled method consistently outperforms the analytical reconstruction for all tested cases. Lastly, we perform a model mismatch analysis to understand the effect of discretization along spectral dimension.

In Chapter 3, we turn our attention to the limitations of the conventional image formation model used in Chapter 2. For this, we focus on general multidimensional imaging systems whose measurements involve convolution and superposition, and formulate a realistic truncated image formation model to overcome these limitations.

The formulated model involves summations of truncated linear convolutions. This is compatible with real-world imaging systems since the detectors have finite support and can only acquire the system's response in a limited region. This image formation model does not make any assumptions about the support of the unknown images outside the field of view (FOV) (i.e., end conditions). In fact, the truncated model produces the same measurements regardless of the end conditions. Therefore, the truncated image formation model can represent the physical reality better and reduces the discrepancies between experimental measurements and the forward model. Then, we develop deep learning-based unrolled reconstruction methods using the truncated image formation model. Unlike the inverse problem in Chapter 2, the sub-problem containing the data fidelity term does not have a fast-to-compute closed-form solution since it requires forming, storing, and inverting huge matrices. As a result, we solve it using the conjugate gradient method with an efficient implementation involving FFTs. However, using the conjugate gradient method increases the training time. Therefore, we also develop an alternative solution approach using the HQS method with additional variable splittings. In the alternative approach, each sub-problem has a closed-form solution that can be computed efficiently without forming, storing, and inverting any huge matrix. The performance of the developed deep learning-based reconstruction method is illustrated for single-frame deconvolution and diffractive multi-spectral imaging applications. Results demonstrate that the developed method outperforms the method with analytical prior. Through a model mismatch analysis, we also illustrate the performance improvement gained with the truncated image formation model over the conventional image formation model in Chapter 2.

Lastly, in Chapter 4, we extend the deep learning-based unrolled reconstruction methods to a compressive spectral imaging modality by using both a simplified model, like the one in Chapter 2, and a realistic truncated model involving truncated convolutions, like the one in Chapter 3. We efficiently solve the sub-problem containing the data fidelity term using the steepest descent and conjugate gradient methods with FFTs. Because we need to exploit spectral correlations in a compressive spectral imaging setting in addition to spatial correlations, we solve the denoising sub-problem using a 3D CNN to exploit both spatial and spectral correlations. For a sample imaging application, we comparatively evaluate the performance of the developed reconstruction

methods, which shows superior results compared to analytical priors. The methods developed in this chapter are applicable to similar multidimensional imaging problems that involve the reconstruction of a correlated image cube from its superimposed, coded, and blurred measurement. We also perform a model mismatch analysis to illustrate the significance of the truncated model.

The rest of this thesis is organized as follows. In Chapter 2, we present a diffractive multi-spectral imaging system and develop various deep learning-based reconstruction methods. In Chapter 3, we address the limitations and challenges of the conventional image formation model used in Chapter 2, and formulate a realistic truncated image formation model. We also develop deep learning-based unrolled reconstruction methods. In Chapter 4, we extend deep learning-based unrolled reconstruction methods to a compressive spectral imaging modality by using both conventional and truncated image formation models. Finally, we conclude the thesis and discuss future research directions in Chapter 5.

CHAPTER 2

EFFICIENT DEEP LEARNING BASED RECONSTRUCTION FOR DIFFRACTIVE MULTI-SPECTRAL IMAGING

2.1 Introduction

Spectral imaging is a useful technique with applications in remote sensing, astronomy, physics, chemistry, biology, and industrial inspection. However, capturing the three-dimensional (3D) spectral data cube using 2D detectors is challenging. Traditional methods achieve this by scanning the scene either spatially or spectrally, which results in long acquisition times, and low spatial or spectral resolutions. Computational spectral imaging aims to overcome these limitations by incorporating computation [6,10]. Several computational spectral imaging modalities have been proposed such as compressive modalities enabling reduced measurements (reduced acquisition time) [9, 34, 42, 43] and non-compressive modalities enabling improved spatial and spectral resolutions [7, 40, 41] using diffractive lenses and filters.

In this chapter, we develop learning-based reconstruction methods for a multi-spectral imaging modality with diffractive lenses. The imaging system is illustrated in Fig. 2.1 [7]. This system offers higher spatial and spectral resolutions than the traditional spectral imagers [7, 36, 40, 41], but additionally requires an inverse problem to be solved for image reconstruction. Because the diffractive lens in this system has wavelength-dependent focusing behavior, each measurement is a superposition of differently blurred spectral components. Hence the goal in the inverse problem is

Some parts of this chapter have been published in [36].

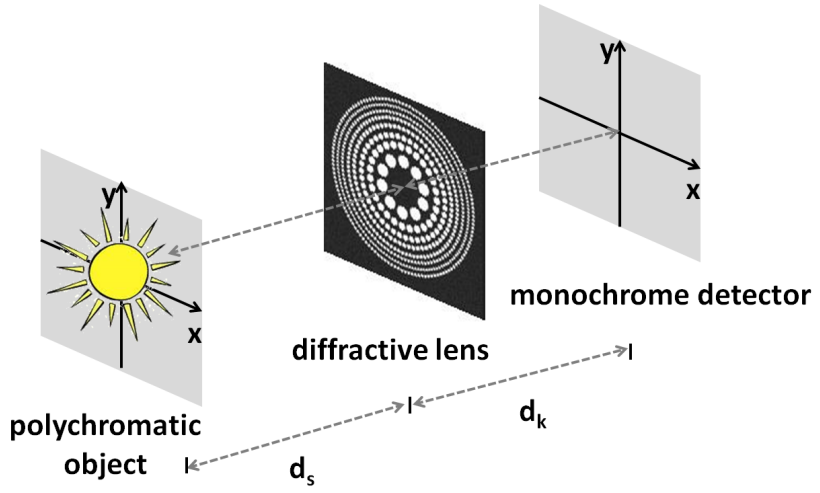


Figure 2.1: Schematic view of the diffractive lens based spectral imaging modality.

to reconstruct the multiple spectral images from their superimposed and blurred versions. We develop different learning-based reconstruction methods to solve this inverse problem.

Recently deep learning-based reconstruction have been successfully applied to various inverse problems in imaging [26, 27] including spectral imaging [34, 35, 38]. These methods are generally based on plug-and-play regularization [31–33] or unrolling [34, 35, 38, 39], both of which are considered in this work. In our development, we first formulate the inverse problem as an optimization problem and then split it into two sub-problems using half quadratic splitting (HQS) method. The sub-problem containing the data fidelity term has a closed-form solution and is solved efficiently using FFTs. The second sub-problem corresponds to a denoising problem. In the plug-and-play approach, we solve the denoising problem using a deep CNN denoiser trained separately with noisy spectral images. Spectral images are then reconstructed iteratively by alternating between these two sub-problems. In the second approach, we unroll the algorithm into an end-to-end trainable network. After training, spectral images are directly reconstructed by feeding this network with the measurements.

For a sample imaging application with practical significance [40], we comparatively evaluate the performance of the developed reconstruction methods using different

CNN architectures and loss functions (i.e. l_2 and l_1 losses). Results illustrate that unrolled reconstruction method with U-Net provides a significant performance improvement over the other learned and analytical reconstruction methods used in the comparison. We then evaluate the performance of the unrolled reconstruction method under different signal-to-noise ratios (SNRs). Results show that unrolled reconstruction method consistently outperforms the reconstruction method with analytical prior for all SNR levels. Lastly, we perform a model mismatch analysis to understand the effect of discretization along spectral dimension. Another type of model mismatch can arise from the assumptions made for the forward model, which will be investigated thoroughly in Chapter 3. The methods developed in this section are applicable to similar imaging problems that involve reconstruction of images from their superimposed and blurred versions.

2.2 Forward Model

In this section, we present the discretized forward model. The continuous forward model can be found in [36]. Measurements are obtained through the spectral imaging system shown in Fig. 2.1. For the input, we consider a polychromatic incoherent illumination consisting of P different wavelengths λ_p for $p = 1, \ldots, P$. The optical field is first passed through a diffractive lens and measured by a monochrome detector. The focusing behavior of the diffractive lens is wavelength-dependent. As a result, each spectral component is exposed to a different blur. Then each measurement is a superposition of differently blurred spectral components. To achieve measurement diversity, we can take R measurements using different approaches. For example, multiple measurements can be obtained by fixing the detector distance, i.e., d_k, and changing the diffractive lens design. In this case, discretized measurements can be expressed as:

$$t_r[m,n] = \sum_{p=1}^{P} x_p[m,n] * g_{r,p}[m,n]. \tag{2.1}$$

Here $t_r[m,n], x_p[m,n], g_{r,p}[m,n]$ are respectively the sampled versions of the r^{th} measurement, the p^{th} spectral image, and the point spread function (PSF) of the diffractive lens acting to the p^{th} spectral image in the r^{th} measurement. PSF is the 2D impulse response of an imaging system. The exact form of these PSFs can

be found in [36]. Let the PSF $g_{r,p}[m,n]$ has $M \times M$ support, i.e. $g_{r,p}[m,n] = 0$ for $m,n \notin [0, M-1]$. We assume that the supports of the spectral images are slightly smaller than the detector range covered by N pixels, i.e. $x_p[m,n] = 0$ for $m,n \notin [0, N-M]$. Then the linear convolution in Eq. (2.1) can be replaced with a circular convolution of N points, which will be useful to develop fast image reconstruction methods.

Due to the linearity of the convolution operator, the discrete model can be formulated in the following matrix-vector form using appropriate lexicographic ordering:

$$\mathbf{t}_r = \sum_{p=1}^{P} \mathbf{H}_{r,p}\mathbf{x}_p. \tag{2.2}$$

Here $\mathbf{H}_{r,p}$ is an $N^2 \times N^2$ block circulant matrix with circular blocks corresponding to the circular convolution operation with the PSF $g_{r,p}[m,n]$. The vectors $\mathbf{t}_r$ and $\mathbf{x}_p$ correspond to lexicographically ordered versions of the observation, $t_r[m,n]$, and spectral image, $x_p[m,n]$, respectively. By combining all the observation vectors, $\mathbf{t}_r$, into a single vector, $\mathbf{t}$, and all the image vectors, $\mathbf{x}_p$, into the vector $\mathbf{x}$, we obtain

$$\mathbf{t} = \mathbf{H}\mathbf{x}, \tag{2.3}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \dots & \mathbf{H}_{1,P} \\ \vdots & & \vdots \\ \mathbf{H}_{R,1} & \dots & \mathbf{H}_{R,P} \end{bmatrix}, \tag{2.4}$$

where $\mathbf{t} = [\mathbf{t}_1^T | \dots | \mathbf{t}_R^T]^T$, $\mathbf{x} = [\mathbf{x}_1^T | \dots | \mathbf{x}_p^T]^T$, and $\mathbf{H}$ is a $RN^2 \times PN^2$ matrix. The final observation model can be expressed with noise as follows:

$$\mathbf{y} = \mathbf{t} + \mathbf{w} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{2.5}$$

where $\mathbf{w} = [\mathbf{w}_1^T | \dots | \mathbf{w}_R^T]^T$ is the additive noise vector. Here, white Gaussian noise is assumed, where $(\mathbf{w}_r)_i \sim N(0, \sigma_r^2)$ is uncorrelated over different pixels i and measurements r, with σ_r representing the noise standard deviation for the r^{th} measurement. This noise assumption is generally valid for practical multi-spectral imaging scenarios of interest.

2.3 Inverse Problem

In the inverse problem, the goal is to recover the unknown spectral images, $\mathbf{x}$, from the measurements, $\mathbf{y}$, obtained with the considered spectral imaging system. This inverse problem can be considered as a multi-frame deconvolution problem where the measurements are in the form of superimposed blurred images. That is, each measurement is a superposition of focused or defocused versions of different spectral images. This problem is inherently ill-posed; as the PSFs, $g_{r,p}[m,n]$, of different wavelengths, p, and measurements, r, become more similar (for example, when the difference between different wavelengths decreases), the conditioning of the problem gets worse due to the increased dependency of the columns or rows of $\mathbf{H}$, respectively.

There are various approaches for solving ill-posed linear inverse problems. A systematic approach to regularization uses prior knowledge about the unknown solution and leads to the minimization of an appropriately formulated cost function [44, 45]. The prior information can be introduced in a deterministic or a statistical setting, with the latter related to the Bayes framework [46]. A general formulation of the inverse problem can be expressed as:

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \; ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{x}). \tag{2.6}$$

Here the first term controls consistency of the forward model with the measurements, the second term controls consistency of the solution with our prior knowledge. The parameter λ determines the trade-off between these data fidelity and prior terms. As the analytical priors have recently been shown to be outperformed by deep priors [26, 27, 36, 39], there is significant interest in learning the prior through a deep network that is trained offline or end-to-end with an unrolling approach. In this work, we develop both approaches to solve the inverse problem with deep priors.

2.4 Developed Reconstruction Methods

In this section, we develop fast and efficient deep learning-based reconstruction methods using the HQS framework to solve the optimization problem in Eq. (2.6). We first re-formulate Eq. (2.6) as a constrained optimization problem by introducing the aux-

iliary variable $\mathbf{z}$:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}. \tag{2.7}$$

By using the HQS method, this constrained optimization problem can be turned into a non-constrained one as follows:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}||_2^2, \tag{2.8}$$

where ν is a penalty parameter. We can solve this problem by performing minimization over $\mathbf{x}$ and $\mathbf{z}$ alternatingly as follows:

$$\mathbf{x}^{l+1} = \arg\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \nu||\mathbf{z}^l - \mathbf{x}||_2^2, \tag{2.9a}$$

$$\mathbf{z}^l = \arg\min_{\mathbf{z}} \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}^l||_2^2. \tag{2.9b}$$

The data fidelity update step in Eq. (2.9a) has a closed-form solution given by

$$\mathbf{x}^{l+1} = (\mathbf{H}^H\mathbf{H} + \nu\mathbf{I})^{-1}(\mathbf{H}^H\mathbf{y} + \nu\mathbf{z}^l). \tag{2.10}$$

This can be computed efficiently in the frequency domain [9, 47] using the diagonalizability of the convolution matrices involved. In fact, the system matrix $\mathbf{H}$ can be expressed as $\mathbf{H} = \bar{\mathbf{F}}^H\mathbf{\Lambda}\tilde{\mathbf{F}}$, where $\bar{\mathbf{F}} = \mathbf{I}_R \otimes \mathbf{F}_{2D}$ and $\tilde{\mathbf{F}} = \mathbf{I}_P \otimes \mathbf{F}_{2D}$ with $\otimes$ denoting the Kronecker product, $\mathbf{I}_n$ denoting the identity matrix of size $n \times n$, and $\mathbf{F}_{2D}$ denoting the unitary 2D discrete Fourier transform (DFT) matrix. Multiplication by $\tilde{\mathbf{F}}$ or $\tilde{\mathbf{F}}^H$ corresponds to taking the 2D DFT or inverse DFT of all spectral bands. Similarly, multiplication by $\bar{\mathbf{F}}$ corresponds to taking the 2D DFT of all measurements. The matrix $\mathbf{\Lambda}$ is a block matrix of diagonal matrices whose diagonals consist of the 2D DFT of the PSFs [9, 47]. Using this representation of $\mathbf{H}$ in Eq. (2.10) yields

$$\mathbf{x}^{l+1} = \tilde{\mathbf{F}}^H(\mathbf{\Lambda}^H\mathbf{\Lambda} + \nu\mathbf{I})^{-1}(\mathbf{\Lambda}^H\bar{\mathbf{F}}\mathbf{y} + \nu\tilde{\mathbf{F}}\mathbf{z}^l). \tag{2.11}$$

Here the inverse of $\mathbf{\Lambda}^H\mathbf{\Lambda} + \nu\mathbf{I}$ can be computed efficiently through a recursive block matrix inversion approach that only requires the inversion of diagonal matrices [47].

The problem in Eq. (2.9b) corresponds to a denoising problem and can be solved using deep denoisers. Here we solve this problem using a CNN that is trained in two different ways. Details of these two methods are given in the following sections.

2.4.1 Plug-and-Play Regularization

In the developed plug-and-play regularization method, we train the denoiser CNN using noisy and clean versions of the ground truth spectral images (without using the forward model and the measurements). For training, we first obtain noisy images for a single SNR by adding white Gaussian noise to the clean spectral images. Then, we train this network using noisy spectral images as input of the network, and minimizing the loss between the output of the network (denoised images) and clean spectral images. Lastly, we reconstruct the spectral images by alternating between the denoising and data fidelity steps as shown in Fig. 2.2.

2.4.2 End-to-End Learned Reconstruction with Unrolling

In the developed end-to-end learned reconstruction, after fixing the iteration number L, we unroll the algorithm described by Eq. (2.9a) and (2.9b) into an end-to-end trainable network [39] shown in Fig. 2.3. For training, we first simulate the measurements $\mathbf{y}$ using the forward model in Eq. (2.5). Then, we train this network using simulated noisy measurements as input of the network, and minimizing the loss between the output of the network (reconstructed images) and ground truth spectral images. Spectral images are then reconstructed by feeding this network with the available measurements. Weights between different iterations can be shared to decrease the

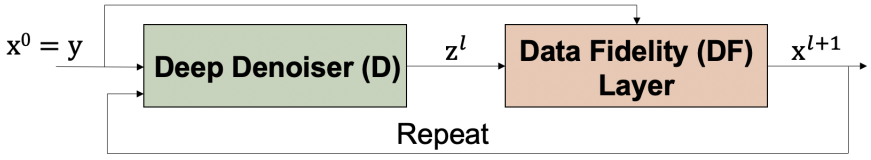


Figure 2.2: The developed plug-and-play regularization method.

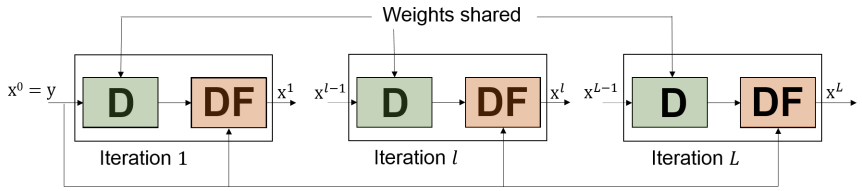
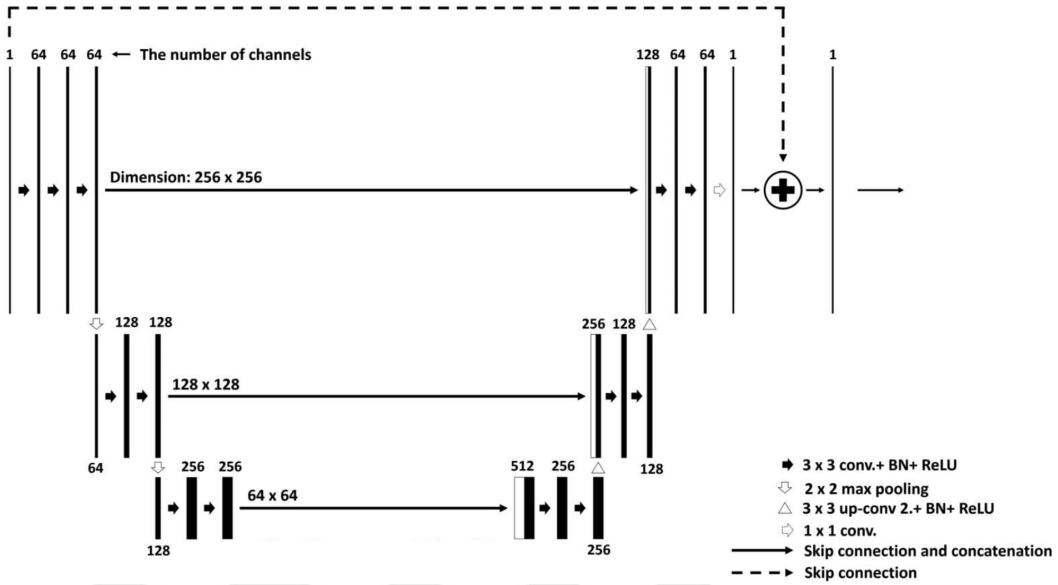


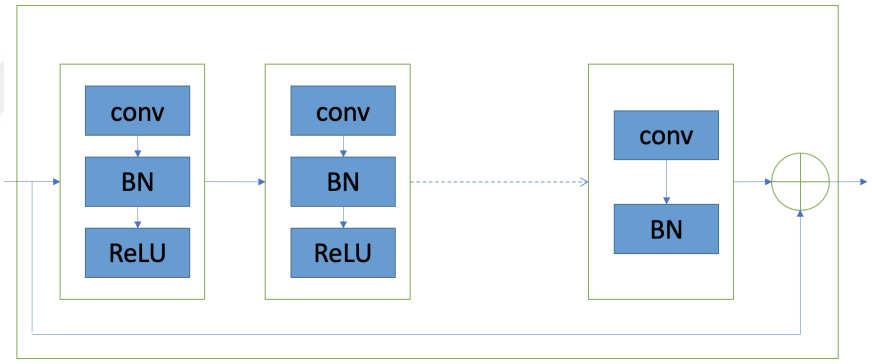
Figure 2.3: The developed end-to-end learned reconstruction with unrolling.

number of trainable parameters and to decouple them with the number of unrolling. We investigate both weights shared and unshared scenarios.

To also investigate the impact of the CNN architecture, we use two different architectures for both methods: I) U-Net adopted from [48] with 3 levels, and II) another CNN architecture adopted from [39] with 5 layers. These network architectures are shown in Fig. 2.4. The Network-I (U-Net), shown in (a), is the modified version of the original U-Net [49] with additional batch normalization layers and skip connections between the input and the output. This network has symmetric downsampling and upsampling convolutional layers, where in between there are skip connections. In the Network-II, shown in (b), each layer contains 64 convolution filters, batch normalization, and non-linear rectified linear unit (ReLU, $f(x) = \max(0, x)$) activation functions. There is no ReLU unit only in the final layer, i.e., N^{th} layer so that the negative part of the learned noise is not lost. Both networks employ a standard multichannel filtering approach to increase the expression power of the network, where filter sizes are 3×3. Moreover, residual learning strategy is used in each case to avoid the vanishing gradient problem [50] during training. That is, the output of the final layer is added with the input to get the output of the denoiser. We have abbreviated our methods using (I) and (II) to identify the used network architecture. For instance, Unrolled-I is the unrolling-based method developed with U-Net.



(a)



(b)

Figure 2.4: CNN architectures. Network-I (U-Net) (a) (figure adapted from [51]), Network-II (b) (figure adapted from [39]). conv, convolutional filter; BN, batch normalization; ReLU, rectified linear unit; up-conv 2., transposed convolutional layer with a stride of two.

2.5 Numerical Results

In this section, we demonstrate the performance of the developed methods and provide a comparative analysis. We also demonstrate the performance of the developed methods under different SNRs and model mismatch cases. For this, we consider a solar imaging application with three nearby solar emission lines in the EUV regime, with central wavelengths of $\lambda_1 = 33.28$ nm, $\lambda_2 = 33.42$ nm and $\lambda_3 = 33.54$ nm (i.e., $P = 3$) [52]. Our goal is to obtain the spectral images for these three nearby emission lines with diffraction-limited spatial resolution. Such high spatial and spectral resolutions are not possible to achieve with the conventional spectral imagers involving wavelength filters [7,8,53,54]. Obtaining these spectral images will provide information about the important physical parameters of the Sun's extended atmosphere, and hence will enable the investigation of the dynamic plasma behavior [52,55].

For the diffractive lens, a sample photon sieve (PS) design proposed in [8] for the considered application is adapted, where the outer diameter was chosen as $D = 25$ mm, and the diameter of the smallest hole as $\Delta = 5\,\mu$m. The resulting PS has first-order focal lengths of $f_1 = 3.756$ m, $f_2 = 3.740$ m, and $f_3 = 3.727$ m at the corresponding wavelengths (i.e. $f = D\Delta/\lambda$) [8]. In our case, measurements are taken by changing the focusing behavior of this PS. That is, for each measurement, we fix the measurement plane at the distance $f_2 = 3.740$ m and obtain the measurements by changing the PS design at each shot. Same design parameters are used to focus λ_2 and only the diameter D is changed to focus λ_1 or λ_3 onto the distance f_2. Because the measurement plane is fixed to $f_2 = 3.740$ m, the outer diameter is decreased by 104.6 μm for λ_1 and is increased by 89.9 μm for λ_3. These modifications on the PS design are well within the limits of modern lithography techniques [56].

A sample measurement simulated using Eq. (2.5) for an SNR of 25 dB (where SNR is defined as $10\log_{10}\frac{\sigma_t^2}{\sigma_w^2}$) is shown in Fig. 2.5. This figure illustrates the three measurements acquired with the different PS designs described above and also the contributions from each spectral image. To also demonstrate the wavelength-dependent behavior of the imaging system, the acting PSFs of the diffractive lens at the three wavelengths are illustrated in Fig. 2.5, (e), (j), (o). These are for the measurement taken with the PS design focusing the solar emission line at the central wavelength of

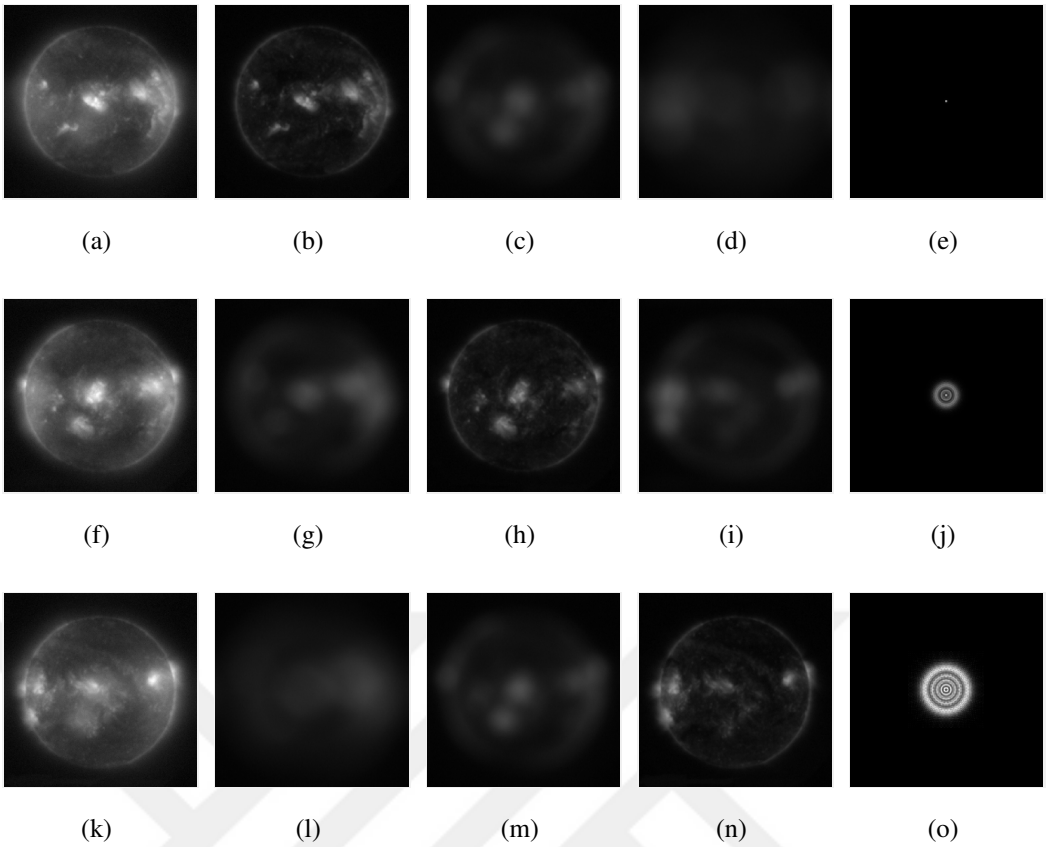


Figure 2.5: Measured intensities for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (f), and λ_3 (k); the contribution of the 1st, 2nd, and 3rd spectral images to the measurement with the PS design focusing wavelength λ_1 (b)-(d), λ_2 (g)-(i), λ_3 (l)-(n); sampled and zoomed PSFs of the system with the PS design focusing solar emission line at the central wavelength λ_1 for the 1st, 2nd, and 3rd spectral images, (e), (j), and (o), respectively.

λ_1. These PSFs illustrate that each spectral image is exposed to a different amount of blur in each measurement. As clearly seen from the shown contributions and PSFs, the first spectral component is focused and the other two spectral components are defocused in the first measurement. We have similar behavior for the measurements taken with the other two PS design. Hence the measurements involve not only the superposition of all spectral images but also substantial and varying amount of blur and noise. We obtain 3 such measurements.

We train our model using solar EUV images from the NASA SDO database [53]. We

use SDO images from the 1^{st}, 5^{th}, 10^{th}, 15^{th}, 25^{th}, and 30^{th} days of every month for years between 2012 and 2020, to have training data diversity. After removing the images with artifacts (such as out-of-focus), we randomly pick 6 images from each day. These images are then converted to grayscale, resized to 256×256, and normalized between $[0, 1]$. Finally, we combine randomly chosen three images to form 725 training datasets (for $P = 3$ case). In a similar way, we have generated separate 105 test datasets.

For plug-and-play methods, we first simulate noisy spectral images at the SNR of 25 dB. Then, we train our denoisers using these noisy and clean images. For unrolling-based methods, we first simulate the measurements $\mathbf{y}$ at the SNR of 25 dB using the forward model in Eq. (2.5). Then, we train our networks using ground truth spectral images and these measurements. To avoid memory issues, we choose L as 10 and 20 for the Unrolled-I and Unrolled-II methods, respectively. All networks are trained by minimizing the l_2 loss between the ground truth and estimated images using Adam solver [57]. Codes are implemented with Tensorflow on Nvidia GeForce RTX 2080 Ti GPU. Training takes approximately 13, 30, 2 and 1 hours for Unrolled-I, Unrolled-II, Plug&Play-I, and Plug&Play-II, respectively.

2.5.1 Performance Comparison of Different Methods

In this section, we analyze and compare the performance of the developed methods for input scenes with different characteristics. The spectral datasets, numbered as 1 to 5, in Table 2.1 have increasing levels of abrupt changes in space and spectrum. For comparison with analytical priors, we also obtain the reconstructions with total variation (TV) regularization. TV has significant popularity as it provides superior results for 2D images with significant structure or piecewise-constant characteristics [45].

The average PSNR and SSIM values between the diffraction-limited ground truth images and reconstructed spectral images are listed in Table 2.1. For unrolled methods, weights between different iterations are shared. As seen, unrolled methods outperform plug-and-play methods, as well as TV-based reconstruction. Moreover, Unrolled-I incorporating U-Net has the best generalization capability and provides the highest SSIM values for all cases. On the other hand, the performance of TV

reconstruction decreases with the increased structure in the images (i.e., from Dataset 1 to Dataset 5). The results also illustrate the effect of the training data. Our training set contains images more similar to Dataset 1 & 2 than Dataset 3, which seems to affect the performance of all learning-based methods. Moreover, TV reconstruction and plug-and-play methods do not perform well for Dataset 4 & 5, which are more challenging to reconstruct due to increased structure and larger changes in space and spectrum.

Table 2.1: Average PSNR (dB) and SSIM values for TV and deep learning-based reconstructions at 25 dB input SNR for different test datasets (with best values shown in bold).

Method	TV Recon.	Unrolled-I	Unrolled-II	Plug&Play-I	Plug&Play-II
Sample Dataset 1	41.94/**0.97**	**42.24/0.97**	40.86/0.96	40.44/0.96	40.67/0.96
Sample Dataset 2	37.12/0.94	**40.80/0.96**	39.02/0.94	34.44/0.87	35.27/0.91
Sample Dataset 3	35.20/0.89	**35.58/0.92**	32.13/0.90	35.21/**0.92**	33.13/0.89
Sample Dataset 4	32.98/0.92	35.37/**0.95**	**37.29**/0.94	28.30/0.75	27.11/0.77
Sample Dataset 5	29.00/0.90	34.60/**0.94**	**35.49**/0.92	25.59/0.67	25.03/0.69
Average	35.25/0.92	**37.72/0.95**	36.96/0.93	32.80/0.83	32.24/0.84

To visually evaluate the results for images with different amount of structure, we provide the reconstructed spectral images for the Datasets 2 and 5 in Fig. 2.6 and 2.7, respectively. For comparison, the diffraction-limited versions of the original images are also given. For each case, we provide false-color representations. False-color representations display images in color, which are recorded in the visible or non-visible range of the electromagnetic spectrum. We obtain these false-color representations by assigning first, second, and third spectral images to red, green, and blue wavelengths, respectively. As seen, unrolling-based methods can reconstruct the fine details of the image, whereas TV reconstruction provides blurry images and plug-and-play methods do not perform sufficient denoising with a single denoiser.

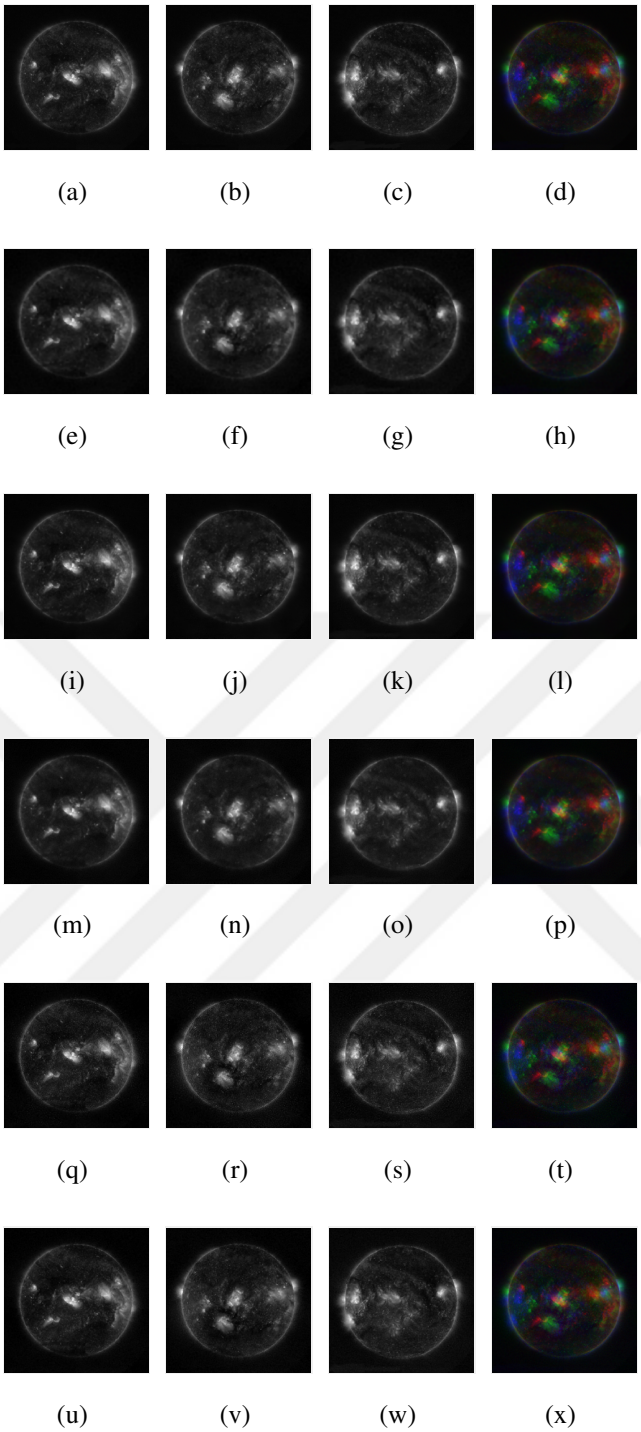


Figure 2.6: Top to bottom: diffraction-limited images, TV, Unrolled-I, Unrolled-II, Plug&Play-I, and Plug&Play-II reconstructions for Dataset 2 (with false-color representations on the right).

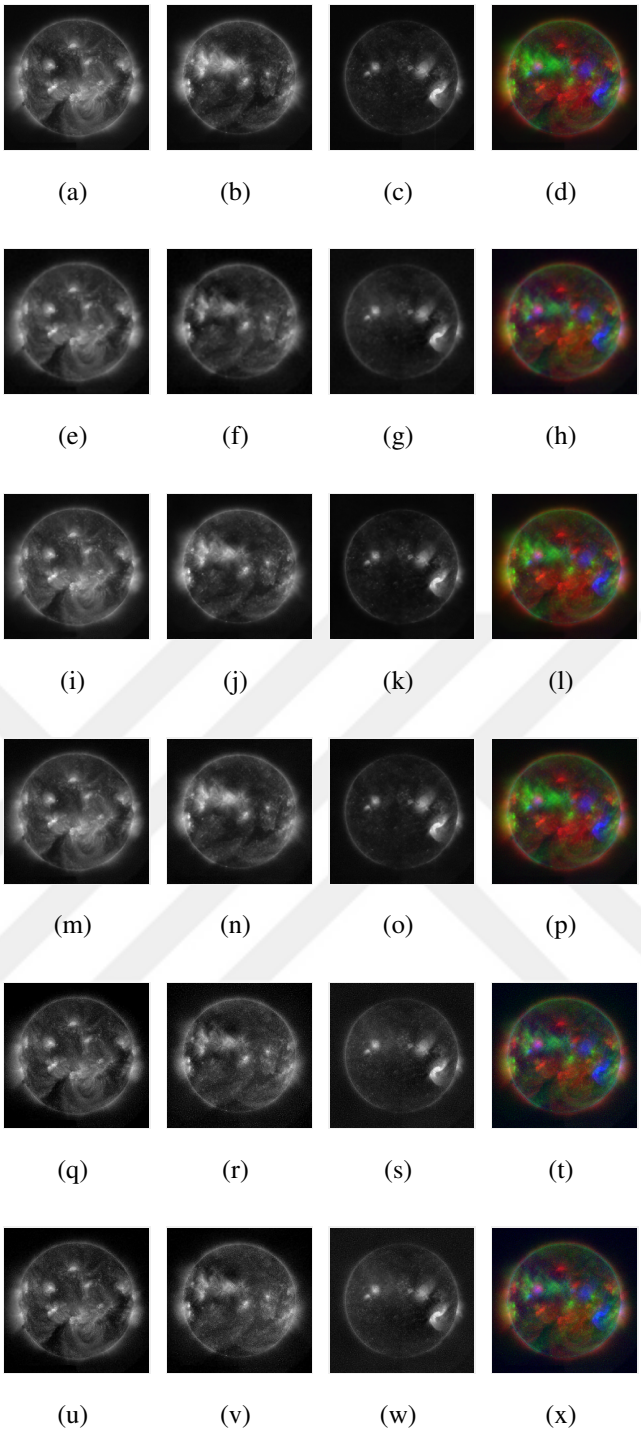
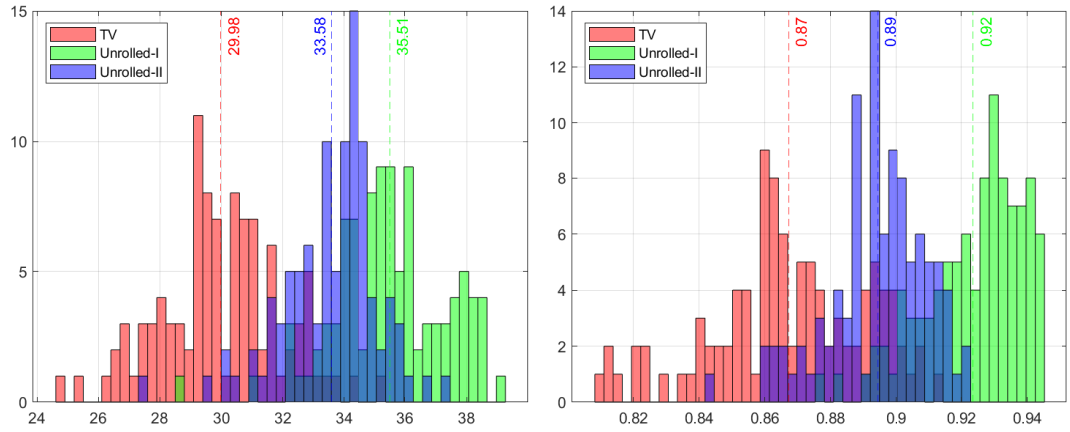


Figure 2.7: Top to bottom: diffraction-limited images, TV, Unrolled-I, Unrolled-II, Plug&Play-I, and Plug&Play-II reconstructions for Dataset 5 (with false-color representations on the right).



(a) (b)

Figure 2.8: PSNR (a) and SSIM (b) histograms for TV, Unrolled-I, and Unrolled-II reconstructions. Red, green, and blue vertical lines show the mean values, respectively.

Since unrolled methods performed better than plug-and-play methods for the tested spectral images, we analyze their performance with a larger test data that contains randomly selected 100 spectral datasets. Fig. 2.8 shows the histograms of the PSNR/SSIM values for these 100 reconstructions. As seen, Unrolled-I provides the best performance in this large test dataset as well with a mean PSNR of 35.51 dB and SSIM of 0.92.

To understand the effect of weight sharing, we trained another network at 25 dB SNR, which has the same structure as Unrolled-I, excepts weights between different iterations are not shared. Still, the penalty parameter ν between different iterations is shared. When we do not share the weights, we have around 0.41 dB and 0.01 improvements in the average PSNR and SSIM values compared to the weight sharing. As the input propagates through the network, the reconstruction artifacts decrease. Using different weights will allow each denoiser to learn to remove different levels of artifacts. Hence reconstruction performance improves. When there is limited memory and/or training samples, one can use the weight sharing strategy since it still provides a good reconstruction quality. However, when there are no such limitations, one can train the network without weight sharing since it improves performance.

Number of trainable parameters for learned reconstruction methods are given in Table 2.2. Here, WS represents the network with weight sharing, and NS represents the network without weight sharing. As seen, sharing the weights decreases the number of trainable parameters significantly. Also, Network-I has more trainable parameters than Network-II, which is expected since U-Net has more convolutional layers. For better comparison, one can decrease the levels of the U-Net, so that the number of trainable parameters for Unrolled-I and Unrolled-II methods becomes similar.

Table 2.2: Number of Trainable Parameters for Different Methods.

Method	Unrolled-I (NS)	Unrolled-I (WS)	Unrolled-II (WS)	Plug&Play-I	Plug&Play-II
Trainable Parameters	18,616,971	1,861,698	112,259	1,861,698	112,259

For many inverse problems, l_1-norm based training loss performs better than l_2 loss. For this reason, we also train Unrolled-I by minimizing the l_1 loss between the ground truth and reconstructed images and analyze the performance for the larger test data. We obtain a mean PSNR of 35.88 dB and SSIM of 0.93 in this case. Compared to l_2 loss, we have a 0.19 dB improvement in PSNR and no improvement in SSIM. This improvement is not significant considering the increased training time with l_1 loss. Hence, in the rest of this chapter, we continue to work with l_2 loss.

We also developed another plug-and-play regularization method in [58], using pre-trained CNN denoisers with natural images instead of denoisers trained with our noisy and clean spectral images. Naturally, the improvement in the reconstruction performance compared to TV was not as significant as the methods developed in this chapter. As a result, we do not present the results of this method in the thesis.

2.5.2 SNR Analysis

In this section, we further analyze the performance of Unrolled-II and TV reconstruction for different levels of SNRs. For this, we train three additional networks for SNRs of 15 dB, 20 dB, and 30 dB. That is, we simulate the measurements $\mathbf{y}$ at these SNRs using the forward model in Eq. (2.5), and then train a different network for each SNR, as before.

Table 2.3: PSNR (dB) and SSIM values for TV and learned reconstruction methods at different input SNRs and for different spectral datasets (with best values shown in bold).

SNR	15 dB		20 dB		25 dB		30 dB	
Method	TV Recon.	Unrolled-II	TV Recon.	Unrolled-II	TV Recon.	Unrolled-II	TV Recon.	Unrolled-II
Average over 105 images	27.56/0.78	**29.70/0.79**	28.61/0.83	**31.57/0.84**	29.73/0.87	**33.55/0.89**	30.80/0.90	**35.77/0.92**
Sample Dataset 1	**38.56/0.93**	37.12/0.91	**39.99/0.95**	37.70/0.92	**41.82/0.97**	40.85/0.96	**43.54/0.98**	43.06/0.97
Sample Dataset 2	34.05/**0.89**	**35.54**/0.87	35.49/**0.92**	**36.40**/0.89	37.09/**0.94**	**39.04/0.94**	38.97/**0.96**	**41.32/0.96**
Sample Dataset 3	**32.38/0.83**	31.20/0.82	**33.80/0.86**	32.62/**0.86**	**35.16**/0.89	32.26/**0.90**	**36.80/0.93**	35.88/**0.93**
Sample Dataset 4	29.99/**0.86**	**33.04/0.86**	30.32/0.88	**35.22/0.90**	33.06/0.92	**37.25/0.94**	34.60/0.94	**35.56/0.95**
Sample Dataset 5	26.44/0.83	**29.09/0.84**	27.48/0.86	**31.25/0.88**	28.83/0.90	**35.72/0.92**	29.98/0.93	**36.72/0.94**

For each test dataset, we compute the average PSNR and SSIM values between the diffraction-limited ground truth images and the reconstructed spectral images. Note that the comparison is again performed with the diffraction-limited versions of the original images to check the imaging system's capability for achieving diffraction-limited spatial resolution. Table 2.3 presents the resulting average PSNR and SSIM values. The first row shows the average performance over the entire test dataset as SNR changes from 15 dB to 30 dB. To also illustrate the performance with increased structure in images, the same 5 spectral datasets in Section 2.5.1 are also considered from this large test dataset and their PSNR/SSIM values are also separately given.

As seen from the first row in Table 2.3, Unrolled-II provides a better performance than TV in this large dataset for all SNRs with the average PSNR changing from 29.7 dB to 35.8 dB and SSIM changing from 0.79 to 0.92 as SNR increases from 15 to 30 dB. The obtained PSNR and SSIM values between the reconstructions and diffraction-limited original images illustrate the imaging system's diffraction-limited spatial resolution capability over a wide range of SNRs. As expected, the learned reconstruction method provides significantly better performance than the TV reconstruction on the average both in terms of PSNR and SSIM. This illustrates the better generalization capability of the learned reconstruction method for a target application. Moreover, as expected, the performance improvement achieved with the learned reconstruction method is more significant for the datasets with significant structure (for example, for Datasets 4 and 5).

2.5.3 Model Mismatch Analysis

We now present model mismatch analysis to understand the effect of discretization along spectral dimension. For this, we consider Unrolled-II method at 25 dB SNR. In the previous sections, we considered an application where the spectrum consists of three isolated nearby emission lines in the EUV regime [40], with central wavelengths of $\lambda_1 = 33.28$ nm, $\lambda_2 = 33.42$ nm and $\lambda_3 = 33.54$ nm. That is, there is no other spectral component than these emission lines. The expected spectral resolution of the considered imaging system is $4\Delta\lambda/D \approx 0.03$ nm near a wavelength of 33 nm, as given by the spectral bandwidth of the diffractive lens [59][Chap.9]. This bandwidth is slightly larger than the typical width of the considered emission lines and hence the entire spectral line will be in good focus when its central wavelength is focused onto the detector plane. As a result, we did not discretize the spectrum with a finer sampling interval and include more terms other than the central wavelengths to the summation in Eq. (2.1).

However, model mismatch error can arise when the spectral sampling interval is not chosen to be smaller than the spectral bandwidth of the designed diffractive lens. To illustrate this, we have considered Eq. (2.1) with a continuous spectrum consisting of three Gaussian spectral lines shown in Fig. 2.9, each with a width slightly larger than the spectral bandwidth. For each spectral line, we have used one solar image from Dataset 2. We have discretized this continuous spectrum with a finer sampling interval of 0.01 nm and included the spectral contribution from each sampled point to the forward model in Eq. (2.1), which results in more than 3 terms.

Fig. 2.10 illustrates the simulated measurements together with the largest contributions from 11 sampled spectral points. Although we have contributions from many points due to finer spectral sampling, we only show contributions from 11 spectral bands since the contributions outside these wavelengths are much less. These 11 spectral bands correspond to the focused central wavelength, its 4 neighbor wavelengths, unfocused central wavelengths, and their 2 neighbor wavelengths. That is, for the first measurement, we show contributions from 33.26, 33.27, 33.28, 33.29, 33.30, 33.41, 33.42, 33.43, 33.53, 33.54, and 33.55 nm. Using these three simulated measurements, we then recover the spectral images at the three central wavelengths

(shown with red dashed lines in Fig. 2.9).

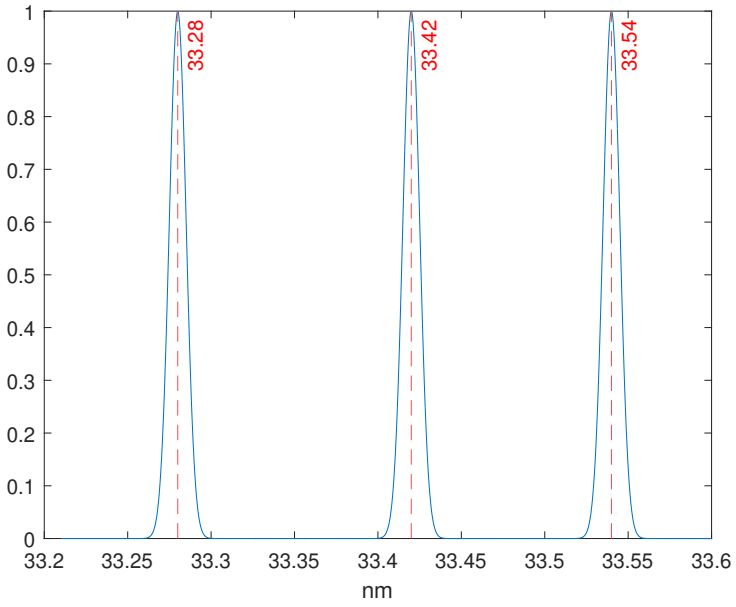
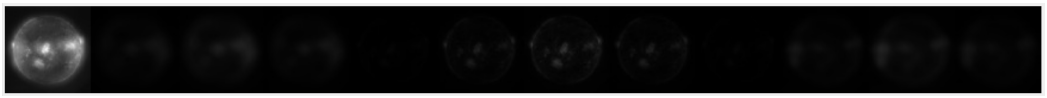


Figure 2.9: Continuous spectrum consisting of three Gaussian spectral lines with central wavelengths 33.28, 33.42, and 33.54 nm.



(a)



(b)



(c)

Figure 2.10: Measured intensities for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (b), and λ_3 (c); with the largest contributions from 11 sampled spectral points.

The recovered images are shown in Fig. 2.11. As seen in this figure, the reconstructions are still very successful. Nevertheless, the performance was slightly decreased (36.57 dB PSNR and 0.92 SSIM for Dataset 2 at 25 dB SNR). To avoid such performance loss, one can consider to revise the design of the diffractive lens to decrease its spectral bandwidth. Other options are to use finer sampling for the spectral discretization (i.e., more terms in Eq. (2.1)) and take more measurements, or train the network with the measurements obtained through a more realistic simulation using the continuous spectrum.

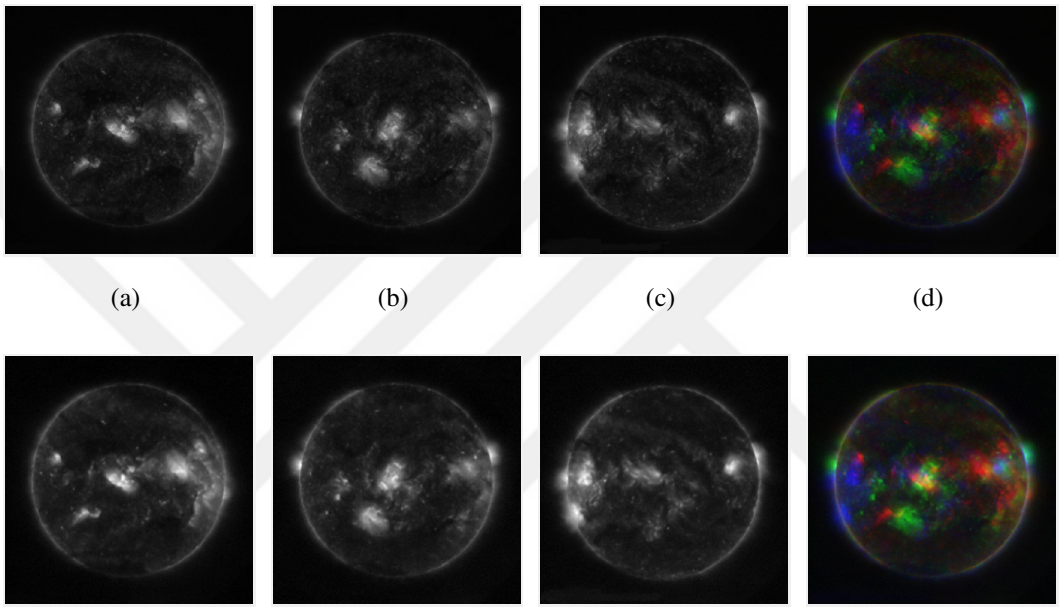


Figure 2.11: Diffraction-limited original images (top), reconstructed spectral images from continuous spectrum measurements with learned-reconstruction method (bottom) for dataset 2 at 25 dB SNR. For each case, false-color representations are also provided on the right.

To illustrate the performance with larger discretization errors, we also increase the width of each Gaussian spectral line in the continuous spectrum by a factor of 2. We will refer this case as "Case II" and to the earlier one in Fig. 2.10 as "Case I". Table 2.4 presents the reconstruction performance of the Unrolled-II method for both Case I and Case II under different measurement SNRs. To quantify the performance loss, we also include the results for the purely discrete treatment as in Section 2.5.1, where we only include 3 spectral terms in Eq. (2.1) corresponding to the central wavelengths.

We perform 5 Monte Carlo runs and compute the average PSNR and SSIM values for each case. As seen from Table 2.4, performance loss is higher for Case II for all SNRs due to larger discretization errors. Also for both Case I and Case II, performance loss increases as the SNR decreases. Nevertheless, the reconstructions are still faithful even for Case II with the average PSNR changing from 30.37 dB to 35.70 dB and SSIM changing from 0.84 to 0.93 as SNR increases from 15 to 30 dB. This suggests the robustness of the imaging system and the reconstruction method to even severe model mismatch errors.

Table 2.4: Average PSNR (dB) and SSIM values for Unrolled-II at different input SNRs for different spectrums.

Spectrum Case	15 dB	20 dB	25 dB	30 dB
Case I	32.81/0.82	34.16/0.88	36.57/0.92	39.04/0.94
Case II	30.37/0.84	32.90/0.89	35.09/0.92	35.70/0.93
Discrete Treatment	34.05/0.89	35.49/0.92	37.09/0.94	38.97/0.96

2.6 Conclusion

In this chapter, we developed different deep learning-based reconstruction methods for multi-spectral imaging with diffractive lenses. The key idea in these approaches is to replace a fixed analytical prior with a data-adaptive deep prior. Unlike deep learning-based direct inversion approaches, which learn the direct mapping from measurements to the reconstructions solely using the training data, we additionally incorporate the physical model into the reconstruction phase. To illustrate the reconstruction performance of the developed methods, we considered a sample application in astrophysical imaging. The presented results demonstrate that the unrolled methods can achieve state-of-the-art reconstructions with diffraction-limited spatial and spectral resolutions. Unrolled methods can also perform well for different levels of SNRs and are robust to model mismatches arising from the discretization along spectral dimension.

The results also demonstrate that unrolled methods consistently outperform other learned and TV reconstruction methods. Unrolled approaches use the forward model

during training. Hence, CNN learns not only denoising but also artifact cleaning. Therefore, they perform better than plug-and-play methods. In plug-and-play approaches, one should optimize hyper-parameter ν and may need to train multiple networks as reconstruction SNR increases after each iteration. Using different denoiser CNNs with proper ordering is challenging. A single denoiser CNN can also be trained with noisy images using multiple SNRs to obtain a more robust denoiser. However, such an approach does not bring any improvement to our application. Although training is longer for unrolled methods, testing is much faster, desirable for practical scenarios. Unrolled methods run for the number of unrolling (e.g., 10, 20) iterations. In contrast, the plug-and-play and TV reconstruction methods run in the order of hundred iterations. In our results, Unrolled-I outperformed Unrolled-II, which is due to the U-Net's ability to process the image at different spatial sizes and its multilevel decomposition.

The developed approaches in this chapter have promising reconstruction performance. The performance of these methods can be further improved by using different networks. We have shown significant performance improvements only by changing the denoiser network. Therefore other networks, such as DenseNets, Feature Pyramid Networks, etc., may perform better than the U-Net in [48]. Using another denoiser network or designing a new denoiser network to fit our multi-spectral imaging application better will be a topic for future study. Although changing training loss to l_1 can bring performance improvements for other inverse problems, for our problem, we did not observe significant improvement in the preliminary trials.

CHAPTER 3

DEEP LEARNING-BASED UNROLLED RECONSTRUCTION FOR IMAGING PROBLEMS INVOLVING TRUNCATED CONVOLUTIONS

3.1 Introduction

Imaging has been a prominent field with widespread applications in physical and life sciences [1, 2]. Images acquired can be either two-dimensional (2D) or higher dimensional (multidimensional). In conventional 2D imaging, the aim is to measure the distribution of light over two-dimensional space. On the other hand, in multidimensional imaging, the goal is to form the image of a scene in more than two dimensions, to acquire much richer information about the chemical, physical, and biological properties of the targeted scene [1–5]. The additional dimensions can contain the spatial, spectral, and/or temporal distribution of the light (or the electromagnetic field). Therefore, the images are hypercubes involving the three spatial coordinates (x, y, z), wavelength (λ), and/or time (t).

Capturing the multidimensional image data with the conventional 2D detectors is challenging. Traditional imaging techniques form the high-dimensional data from a series of 2D measurements. For example, in spectral imaging, three-dimensional spectral datacube is obtained either by scanning the scene spectrally through a series of wavelength filters or by scanning the scene spatially through a spectrometer with a long slit. Such scanning-based methods suffer from low signal-to-noise ratio (SNR), long acquisition time, and temporal artifacts for dynamic scenes [6–8]. Moreover, attainable spatial, spectral, and temporal resolutions are inherently limited by the availability and cost of the optical elements involved. Computational imaging techniques overcome these limitations by passing some of the burden to a reconstruc-

tion algorithm [2,4,5,7,9–13]. These approaches reconstruct the desired image data from superimposed measurements by solving an inverse problem.

In various multidimensional computational imaging systems, the relationship between the measurements (sensor data) and the unknown image data can be characterized by the sum of multiple convolutions. These imaging systems involve an image reconstruction task from superimposed convolutions. Such image reconstruction problems are encountered in various computational imaging modalities such as computational photography, wide-field astronomical imaging, three-dimensional (volumetric) microscopy, spectral imaging, ultrafast imaging, radio interferometric imaging, magnetic resonance imaging, and ultrasound imaging [5,7,9,14–25].

Here, we turn our attention to the limitations of the conventional image formation model used in Chapter 2. For this, we focus on general multidimensional imaging systems whose measurements involve convolution and superposition, and formulate a realistic truncated image formation model to overcome these limitations. The formulated model involves summations of truncated linear convolutions. Hence we take into account the fact that the detectors have finite support and can only acquire the system's response in a limited region. With this, we do not make any unrealistic assumptions about the values (i.e., end conditions) of the unknown images outside the detected region. In fact, the truncated model formulates the measurements regardless of the end conditions, resulting in a better representation of the physical reality.

In this chapter, we develop a deep learning-based unrolled reconstruction method for inverse problems involving a truncated image formation model. Unlike the inverse problem in Chapter 2, the sub-problem containing the data fidelity term does not have a fast-to-compute closed-form solution since it requires forming, storing, and inverting huge matrices. As a result, we solve it using an efficient conjugate gradient implementation involving FFTs. However, using the conjugate gradient method increases the training time. Therefore, we also develop an alternative solution approach using the HQS method with additional variable splittings. In the alternative approach, each sub-problem has a closed-form solution that can be computed efficiently without forming, storing, and inverting any huge matrix. As previously, the denoising sub-problem is solved using a deep CNN denoiser. The performance of the

developed deep learning-based reconstruction method is illustrated for single-frame deconvolution and diffractive multi-spectral imaging applications. Results demonstrate that the developed method outperforms the method with analytical TV prior. Through a model mismatch analysis, we also illustrate the significance of modelling truncation in the image formation model.

3.2 Forward Model

In this section, we formulate the truncated image formation model. As mentioned, in many imaging modalities, measurements can be expressed by the sum of convolutions of the input image with the point spread function (PSF) of the imaging system:

$$y_r[m,n] = \sum_{p=1}^{P} x_p[m,n] * g_{r,p}[m,n] + w_r[m,n]. \tag{3.1}$$

This model allows multiple inputs and multiple outputs. However, it has some drawbacks. Firstly, this model is not generally compatible with real-world imaging systems. Let the PSF $g_{r,p}[m,n]$ has $M_1 \times M_2$ support, i.e., $g_{r,p}[m,n] = 0$ for $m \notin [0, M_1 - 1], n \notin [0, M_2 - 1]$. Let the input images have $N_1 \times N_2$ support, i.e., $x_p[m,n] = 0$ for $m \notin [0, N_1 - 1], n \notin [0, N_2 - 1]$. Then, the measurements $y_r[m,n]$ will have $(M_1 + N_1 - 1) \times (M_2 + N_2 - 1)$ support. That is, the sizes of the measurements are greater than that of input images. However, this may not be realistic since the detectors have finite support and can only acquire the system's response in a limited region. Another drawback of this forward model is that some entries of the measurements depend on the end conditions of the input images. Choosing the proper end condition is challenging. Moreover, different end conditions, such as zero, extended, replicated, mirror, or periodic, yield different forward model matrices [60]. Hence, if the chosen end condition is not compatible with the real-world imaging system, there will be discrepancies between the experimental measurements and the forward model, which can worsen the reconstruction performance.

To overcome these drawbacks, we formulate the following truncated image formation model:

$$y_r[m,n] = \mathcal{T}\left(\sum_{p=1}^{P} x_p[m,n] * g_{r,p}[m,n]\right) + w_r[m,n]. \tag{3.2}$$

Here, $y_r[m,n]$ is the r^{th} measurement of the system, where r ranges from 1 to R. Two-dimensional slices of the unknown image are represented with $x_p[m,n]$, where p ranges from 1 to P. Therefore, each measurement is a superposition of differently blurred image slices. $g_{r,p}[m,n]$ represents the PSF of the imaging system acting to the p^{th} slice in the r^{th} measurement. The acting PSF on each image slice is allowed to be different. The additive white Gaussian noise in the r^{th} measurement is denoted by $w_r[m,n]$. $\mathcal{T}$ denotes the truncation operation. This operation truncates the center part of the convolved signal such that the sizes of the measurements and the image slices are the same, i.e., both have $N_1 \times N_2$ support.

Using lexicographic ordering and the linearity of the convolution operation, the truncated image formation model in Eq. (3.2) can be turned into the following matrix-vector product:

$$\mathbf{y}_r = \mathbf{T}\left(\sum_{p=1}^{P} \mathbf{G}_{r,p}\mathbf{x}_p\right) + \mathbf{w}_r. \tag{3.3}$$

Here, $\mathbf{y}_r \in \mathbb{R}^{N_1N_2}$ is the r^{th} measurement vector, $\mathbf{x}_p \in \mathbb{R}^{N_1N_2}$ is the p^{th} image slice vector, $\mathbf{w}_r \in \mathbb{R}^{N_1N_2}$ is the r^{th} noise vector, $\mathbf{T} \in \mathbb{R}^{N_1N_2 \times (M_1+N_1-1)(M_2+N_2-1)}$ is the truncation matrix, and $\mathbf{G}_{r,p} \in \mathbb{R}^{(M_1+N_1-1)(M_2+N_2-1)\times N_1N_2}$ is the convolution matrix corresponding to the linear convolution operation with the PSF $g_{r,p}[m,n]$.

Now, we illustrate that various two or multidimensional imaging problems can be represented as special cases of the truncated image formation model in Eq. (3.2) and (3.3).

3.2.1 Single-Frame Deconvolution

Single-frame deconvolution is a commonly encountered inverse problem in imaging. The input scene is blurred by the PSF of the imaging system. Therefore, the aim is to reconstruct the input scene from its blurred and noisy version. In this case, we have only one input image and take only one measurement. Therefore, both R and P are 1. Setting R and P to 1 in Eq. (3.2), we arrive at the following image formation model:

$$y[m,n] = \mathcal{T}\left(x[m,n] * g[m,n]\right) + w[m,n], \tag{3.4}$$

which can be turned into the following matrix-vector product:

$$\mathbf{y} = \mathbf{TGx} + \mathbf{w}. \tag{3.5}$$

Here, $\mathbf{y} \in \mathbb{R}^{N_1 N_2}$ is the measurement vector, $\mathbf{x} \in \mathbb{R}^{N_1 N_2}$ is the image vector, and $\mathbf{w} \in \mathbb{R}^{N_1 N_2}$ is the noise vector. The convolution matrix $\mathbf{G} \in \mathbb{R}^{(M_1+N_1-1)(M_2+N_2-1)\times N_1 N_2}$ represents the convolution with the PSF $g[m, n]$. Finally, $\mathbf{T} \in \mathbb{R}^{N_1 N_2 \times (M_1+N_1-1)(M_2+N_2-1)}$ is the truncation matrix. We drop the indices since both R and P are 1.

3.2.2 Snapshot Case

In the snapshot case, the aim is to reconstruct a multidimensional image data, i.e., multiple image slices from a single measurement. Snapshot imaging reduces the acquisition time, which is critical for real-time applications and artifact-free imaging of dynamic scenes. Since we have only one measurement, R is set to 1. There are P image slices. Therefore, Eq. (3.2) becomes:

$$y[m, n] = \mathcal{T}\left(\sum_{p=1}^{P} x_p[m, n] * g_{1,p}[m, n]\right) + w[m, n], \tag{3.6}$$

which can be turned into the following matrix-vector product:

$$\mathbf{y} = \mathbf{T}\left(\sum_{p=1}^{P} \mathbf{G}_{1,p}\mathbf{x}_p\right) + \mathbf{w}. \tag{3.7}$$

By vertically concatenating the image slice vectors $\mathbf{x}_p$'s, we arrive at the following final equation:

$$\begin{gathered}\mathbf{y} = \mathbf{TGx} + \mathbf{w}, \\ \mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{G}_{1,2} & \dots & \mathbf{G}_{1,P} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{bmatrix}.\end{gathered} \tag{3.8}$$

This is an underdetermined problem. The vector $\mathbf{x} \in \mathbb{R}^{PN_1 N_2}$ is the vertically concatenated image vector. The matrix $\mathbf{G} \in \mathbb{R}^{(M_1+N_1-1)(M_2+N_2-1)\times PN_1 N_2}$ contains all linear convolution matrices. Similar to the single-frame deconvolution, $\mathbf{y} \in \mathbb{R}^{N_1 N_2}$, $\mathbf{w} \in \mathbb{R}^{N_1 N_2}$, and $\mathbf{T} \in \mathbb{R}^{N_1 N_2 \times (M_1+N_1-1)(M_2+N_2-1)}$ are respectively the measurement vector, the noise vector, and the truncation matrix.

3.2.3 Multi-Frame Deconvolution

In multi-frame deconvolution, we aim to reconstruct a single image from multiple blurred measurements. Therefore, it is an overdetermined problem. This is particularly useful in reconstructing an image, for which multiple measurements from different angles, and hence with different PSFs, are available. In this case, we have only one image slice, i.e., $P = 1$. We take R measurements. Hence, we arrive at the following image formation model:

$$y_r[m,n] = \mathcal{T}\left(x[m,n] * g_{r,1}[m,n]\right) + w_r[m,n], \tag{3.9}$$

which can be turned into the following matrix-vector product:

$$\mathbf{y}_r = \mathbf{T}\left(\mathbf{G}_{r,1}\mathbf{x}\right) + \mathbf{w}_r. \tag{3.10}$$

By vertically concatenating the measurement vectors $\mathbf{y}_r$'s and noise vectors $\mathbf{w}_r$'s, we arrive at the following final equation:

$$\mathbf{y} = \tilde{\mathbf{T}}\mathbf{G}\mathbf{x} + \mathbf{w},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} \\ \mathbf{G}_{2,1} \\ \vdots \\ \mathbf{G}_{R,1} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_R \end{bmatrix}, \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_R \end{bmatrix}, \tag{3.11}$$

$$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{T} & \dots & \mathbf{0} \\ \vdots & & \vdots \\ \mathbf{0} & \dots & \mathbf{T} \end{bmatrix}.$$

Here, the vector $\mathbf{y} \in \mathbb{R}^{RN_1N_2}$ is the vertically concatenated measurement vector. Similarly, $\mathbf{w} \in \mathbb{R}^{RN_1N_2}$ is the overall noise vector. The matrix $\mathbf{G} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1)\times N_1N_2}$ contains all linear convolution matrices. Lastly, $\tilde{\mathbf{T}} \in \mathbb{R}^{RN_1N_2 \times R(M_1+N_1-1)(M_2+N_2-1)} = \mathbf{I}_R \otimes \mathbf{T}$ is the overall truncation matrix with $\otimes$ denoting the Kronecker product. Similar to the single-frame deconvolution, $\mathbf{x} \in \mathbb{R}^{N_1N_2}$ is the image vector.

3.2.4 Multiple Measurement Multiple Image Case

For the multiple measurement multiple image case, our image formation model and the corresponding matrix-vector product form are respectively given in Eq. (3.2) and (3.3). Similar to previous cases, we arrive at the following final equation after vertically concatenating measurement vectors $\mathbf{y}_r$'s, image slice vectors $\mathbf{x}_p$'s, and noise vectors $\mathbf{w}_r$'s:

$$\begin{aligned}
\mathbf{y} &= \tilde{\mathbf{T}}\mathbf{G}\mathbf{x} + \mathbf{w}, \\
\mathbf{G} &= \begin{bmatrix} \mathbf{G}_{1,1} & \dots & \mathbf{G}_{1,P} \\ \vdots & & \vdots \\ \mathbf{G}_{R,1} & \dots & \mathbf{G}_{R,P} \end{bmatrix}, \\
\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{bmatrix}, \mathbf{y} &= \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_R \end{bmatrix}, \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_R \end{bmatrix}, \\
\tilde{\mathbf{T}} &= \begin{bmatrix} \mathbf{T} & \dots & \mathbf{0} \\ \vdots & & \vdots \\ \mathbf{0} & \dots & \mathbf{T} \end{bmatrix}.
\end{aligned} \tag{3.12}$$

Here, the vector $\mathbf{y} \in \mathbb{R}^{RN_1N_2}$ is the vertically concatenated measurement vector and $\mathbf{x} \in \mathbb{R}^{PN_1N_2}$ is the vertically concatenated image vector. Similarly, $\mathbf{w} \in \mathbb{R}^{RN_1N_2}$ is the overall noise vector. The matrix $\mathbf{G} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1)\times PN_1N_2}$ contains all linear convolution matrices. Again, $\tilde{\mathbf{T}} \in \mathbb{R}^{RN_1N_2\times R(M_1+N_1-1)(M_2+N_2-1)} = \mathbf{I}_R \otimes \mathbf{T}$ is the overall truncation matrix.

As illustrated by different examples, the truncated image formation model can express various two or multidimensional imaging modalities, and therefore provides a unified framework for inverse problems in imaging.

3.3 Inverse Problem

In the inverse problem, the goal is to reconstruct the unknown images, $\mathbf{x}$, from noisy, superimposed, and blurred measurements, $\mathbf{y}$. Considering Eq. (3.5), (3.8), (3.11), and (3.12), we can write the final observation of the truncated image formation model as:

$$\mathbf{y} = \tilde{\mathbf{T}}\mathbf{G}\mathbf{x} + \mathbf{w}. \tag{3.13}$$

All the previous cases are in the form of Eq. (3.13). Only the exact expressions of $\mathbf{y}, \tilde{\mathbf{T}}, \mathbf{G}, \mathbf{x}$, and $\mathbf{w}$ change. The problem in Eq. (3.13) is inherently ill-posed. As the PSFs, $g_{r,p}[m, n]$ become more similar, the conditioning of the problem gets worse due to increased dependency of the columns or rows of $\mathbf{G}$. Moreover, if the number of measurements is less than the number of image slices, i.e., $R < P$, there is also no unique solution. Hence, we introduce regularization to the inverse problem and solve it through the minimization of the following cost function:

$$\min_{\mathbf{x}} \; ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{G}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{x}). \tag{3.14}$$

Here the first term controls the fidelity of the forward model with the measurements, and the second term controls the fidelity of the solution with our prior knowledge. The parameter λ determines the trade-off between the data fidelity and prior terms.

There are popular and powerful choices for the regularizer $\Phi(.)$ [27, 39, 61, 62]. Because we want our approach to be applicable for scenes that can possibly lack correlations in the third dimension, we can exploit either 2D priors for each image slice or a 3D prior for the entire image cube. Among handcrafted analytical priors, total-variation (isotropic or anisotropic) has significant popularity as it provides superior results for images with significant structure or piecewise-constant 2D characteristics [45]. As the analytical priors have recently been shown to be outperformed by deep priors [26, 27, 39], there is also significant interest in learning the prior through a deep network that is trained offline or end-to-end with an unrolling approach. Here, we develop an unrolling approach with deep prior to solve this inverse problem.

3.4 Developed Reconstruction Method

Here, we develop a deep learning-based unrolled reconstruction method to solve the inverse problems associated with the truncated image formation model. Develop reconstruction method applies to all previously described cases.

We can re-formulate Eq. (3.14) as a constrained optimization problem by introducing the auxiliary variable $\mathbf{z}$:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{G}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}. \tag{3.15}$$

By using the HQS method, this constrained optimization problem can be turned into a non-constrained one as follows:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{G}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}||_2^2, \tag{3.16}$$

where ν is a penalty parameter. We can solve this problem by performing minimization over $\mathbf{x}$ and $\mathbf{z}$ alternatingly as follows:

$$\mathbf{x}^{l+1} = \arg\min_{\mathbf{x}} ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{G}\mathbf{x}||_2^2 + \nu||\mathbf{z}^l - \mathbf{x}||_2^2, \tag{3.17a}$$

$$\mathbf{z}^l = \arg\min_{\mathbf{z}} \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}^l||_2^2. \tag{3.17b}$$

The image update step in Eq. (3.17a) requires to solve a least-squares problem, whose normal equation is given by:

$$\left(\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G} + \nu\mathbf{I}\right)\mathbf{x}^{l+1} = \mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y} + \nu\mathbf{z}^l. \tag{3.18}$$

Direct computation of Eq. (3.18) requires forming, storing and inverting huge matrices, which have high computation costs. To have an efficient algorithm that exploits FFT, we do not form any matrices and instead solve this linear system of equations using the conjugate gradient method. Note that the conjugate gradient method requires multiplications with $\tilde{\mathbf{T}}\mathbf{G}$ and $\mathbf{G}^H\tilde{\mathbf{T}}^H$ as given in Algorithm 1. We compute these multiplications without forming or storing any of the matrices. Consider the multiplication of a vector, say $\mathbf{x}$, with the matrix $\tilde{\mathbf{T}}\mathbf{G}$, i.e., $\tilde{\mathbf{T}}\mathbf{G}\mathbf{x}$. We calculate this multiplication efficiently as:

$$\mathcal{S}(x) = \sum_{p=1}^{P} \mathcal{T}\mathcal{F}^{-1}\left(\mathcal{F}\left(g_{r,p}[m,n]\right) \odot \mathcal{F}\left(x_p[m,n]\right)\right). \tag{3.19}$$

Algorithm 1 Conjugate Gradient Method

Input: $\mathbf{y}$: superimposed and blurred measurements, $\mathbf{z}^l$: denoised image, E: conjugate gradient iteration number

Output: $\mathbf{x}^{l+1}$: reconstructed images

$\mathbf{x}^{l+1} \leftarrow \mathbf{z}^l$
$\mathbf{b} \leftarrow \mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y} + \nu\mathbf{z}^l$ {compute using Eq. (3.20)}
$\mathbf{r} \leftarrow \mathbf{b} - \left(\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G} + \nu\mathbf{I}\right)\mathbf{x}^{l+1}$ {compute using Eq. (3.19) and (3.20)}
$\mathbf{p} \leftarrow \mathbf{r}$
$\epsilon_1 = \langle \mathbf{r}, \mathbf{r} \rangle$
$i \leftarrow 0$
for $i = 0$ to $E - 1$ **do**
  $\mathbf{q} \leftarrow \left(\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G} + \nu\mathbf{I}\right)\mathbf{p}$ {compute using Eq. (3.19) and (3.20)}
  $\alpha = \epsilon_1 / \langle \mathbf{p}, \mathbf{q} \rangle$
  $\mathbf{x}^{l+1} \leftarrow \mathbf{x}^{l+1} + \alpha\mathbf{p}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{q}$
  $\epsilon_2 \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$
  $\beta \leftarrow \epsilon_2 / \epsilon_1$
  $\mathbf{p} \leftarrow \mathbf{r} + \beta\mathbf{p}$
  $\epsilon_1 \leftarrow \epsilon_2$
end for
return $\mathbf{x}^{l+1}$

In Eq. (3.19), $\mathcal{S}$ is the system operator for the r^{th} measurement which represents the forward model in Eq. (3.2). We implement this operator in the frequency domain by first taking the 2D FFTs of each image slice x_p and PSF $g_{r,p}$, element-wise multiplying them, taking the 2D inverse FFT, truncating the result, and then summing over all image slices. Element-wise multiplication, i.e., Hadamard product, is denoted by $\odot$. Operators $\mathcal{F}$ and $\mathcal{F}^{-1}$ take 2D FFT and its inverse, which are of $(M_1 + N_1 - 1) \times (M_2 + N_2 - 1)$-point so that we get the result of the linear convolution. Since FFT size is larger than PSF, image, and measurement sizes, it requires zero padding of the signals. For the PSFs, we zero pad by adding $\lceil (N_1 - 1)/2 \rceil$ zero pixels before the leftmost pixels, $\lfloor (N_1 - 1)/2 \rfloor$ zero pixels after the rightmost pixels, $\lceil (N_2 - 1)/2 \rceil$ zero pixels before the upmost pixels, and $\lfloor (N_2 - 1)/2 \rfloor$ zero pixels

after the lowermost pixels. $\lceil(.)\rceil$ and $\lfloor(.)\rfloor$ are respectively the ceiling and flooring operations. For the images and measurements, we zero pad by adding $\lceil(M_1-1)/2\rceil$ zero pixels before the leftmost pixels, $\lfloor(M_1-1)/2\rfloor$ zero pixels after the rightmost pixels, $\lceil(M_2-1)/2\rceil$ zero pixels before the upmost pixels, and $\lfloor(M_2-1)/2\rfloor$ zero pixels after the lowermost pixels. So in both cases, non-zero actual pixels lie at the center. The operator $\mathcal{T}$ is the truncation operation. It truncates the $(M_1+N_1-1)\times(M_2+N_2-1)$ image into $N_1\times N_2$ image by deleting the leftmost $\lceil(M_1-1)/2\rceil$ pixels, rightmost $\lfloor(M_1-1)/2\rfloor$ pixels, upmost $\lceil(M_2-1)/2\rceil$ pixels, and lowermost $\lfloor(M_2-1)/2\rfloor$ pixels. That is, it truncates the center $N_1\times N_2$ pixels.

Now consider the multiplication of a vector, say $\mathbf{y}$, with the matrix $\mathbf{G}^H\tilde{\mathbf{T}}^H$, i.e., $\mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y}$. We calculate this multiplication efficiently as:

$$\mathcal{S}^*(y)=\sum_{r=1}^{R}\mathcal{T}\mathcal{F}^{-1}\left(\mathcal{F}\left(g_{r,p}^{\dagger}[m,n]\right)\odot\mathcal{F}\left(y_r[m,n]\right)\right). \tag{3.20}$$

Here $\mathcal{S}^*$ is the adjoint of the system operator for the p^{th} spectral band with $g_{r,p}^{\dagger}[m,n]=g_{r,p}[M_1-1-m,M_2-1-n]$ being flipped version of the PSF $g_{r,p}$. Similar to system operator, we compute the adjoint of the system operator in the frequency domain by first taking 2D FFTs, element-wise multiplying them, taking the 2D inverse FFT, truncating the inverse FFT, and then summing over all measurements. Hence, all matrix-vector multiplications can be computed using operators $\mathcal{S}$ and $\mathcal{S}^*$. Also, in Algorithm 1, additions are element-wise, and inner products are computed using element-wise multiplications followed by element-wise additions. Hence, all steps of the conjugate gradient method are computed fast and efficiently without forming or storing any of the matrices. We give the updates in matrix-vector forms just for convenience. The conjugate gradient method runs for E iterations.

The problem in Eq. (3.17b) corresponds to a denoising problem and can be solved using deep denoisers. In Chapter 2, unrolled reconstruction incorporating the U-Net [48] performed better than the unrolled reconstruction using the other CNN architecture [39]. Therefore, here we solve the denoising problem using the U-Net. Again we set the number of levels to 3. Hence, the U-Net has the same structure shown in Fig. 2.4 (a).

After fixing the iteration number L, we unroll the algorithm described by Eq. (3.17a)

and (3.17b) into an end-to-end trainable network [39], and learn the parameters in this structure. Images are then reconstructed by feeding this network with the measurements. The iterative algorithm and proposed unrolled reconstruction network have similar structure as Fig. 2.3, except the weights are not shared. The penalty parameter is shared. The number of trainable parameters is 18,616,971.

3.5 Numerical Results

In this section, we demonstrate the performance of the deep learning-based unrolled reconstruction method for the multiple measurement multiple image case and single-frame deconvolution. We also compare its performance with analytical TV reconstruction.

3.5.1 Performance Analysis for the Multiple Measurement Multiple Image Case

In this case, we have multiple inputs and multiple outputs. For this case, we focus on the multi-spectral imaging problem, which is introduced and studied in Chapter 2. The imaging system is shown in Fig. 2.1. Detailed analysis of the imaging system can be found in Chapter 2 and [36].

Same as Chapter 2, we consider the solar imaging of three nearby emission lines in the EUV regime [40], with central wavelengths of $\lambda_1 = 33.28$ nm, $\lambda_2 = 33.42$ nm, and $\lambda_3 = 33.54$ nm. For the diffractive lens, a sample photon sieve (PS) design proposed in [8] for the considered application is adapted, where the outer diameter was chosen as $D = 25$ mm, and the diameter of the smallest hole as $\Delta = 5\,\mu$m. The resulting PS has first-order focal lengths of $f_1 = 3.756$ m, $f_2 = 3.740$ m, and $f_3 = 3.727$ m at the corresponding wavelengths (i.e. $f = D\Delta/\lambda$) [8]. In our case, measurements are taken by changing the focusing behavior of this PS. That is, for each measurement, we fix the measurement plane at the distance $f_2 = 3.740$ m and obtain the measurements by changing the PS design at each shot. Same design parameters are used to focus λ_2 and only the diameter D is changed to focus λ_1 or λ_3 onto the distance f_2. Because the measurement plane is fixed to $f_2 = 3.740$ m, the outer diameter is decreased by 104.6 μm for λ_1 and is increased by 89.9 μm for λ_3. These modifications on the PS

design are well within the limits of modern lithography techniques [56].

A sample measurement simulated using Eq. (3.2) for an SNR of 25 dB is shown in Fig. 3.1. This figure illustrates the three measurements acquired with the different PS designs described above and also the contributions from each spectral image. To illustrate the wavelength-dependent behavior of the imaging system, the acting PSFs of the system to the three spectral images are also shown in Fig. 3.1 for each measurement. These PSFs illustrate that each spectral image is exposed to a different amount of blur in each measurement. As clearly seen from the shown contributions and PSFs, the first spectral component is focused and the other two spectral components are defocused in the first measurement. We have similar behavior for the measurements

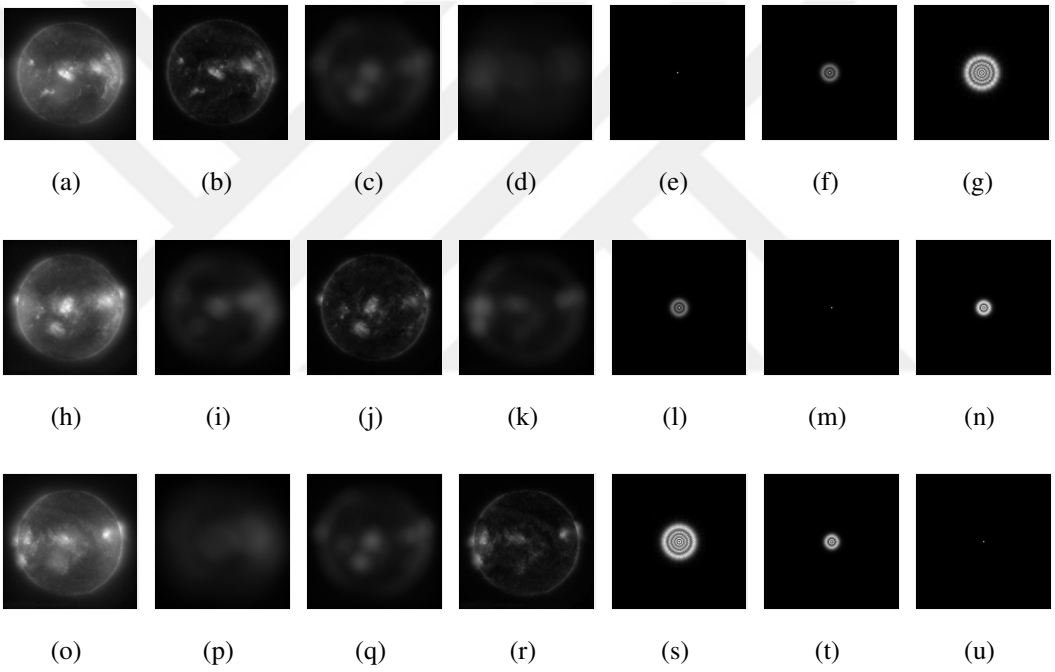


Figure 3.1: Measured intensities using truncated image formation model for SNR=25 dB with the PS design focusing solar emission line at the central wavelength λ_1 (a), λ_2 (h), and λ_3 (o); the contribution of the 1st, 2nd, and 3rd spectral images to the measurement with the PS design focusing solar emission line at the central wavelength λ_1 (b)-(d), λ_2 (i)-(k), λ_3 (p)-(r); sampled and zoomed PSFs of the system for the 1st, 2nd, and 3rd spectral images with the PS design focusing solar emission line at the central wavelength λ_1 (e)-(g), λ_2 (l)-(n), λ_3 (s)-(u) , respectively.

taken with the other two PS design. Hence the measurements involve not only the superposition of all spectral images but also substantial and varying amount of blur and noise. We obtain 3 such measurements. Measurements and contributions from each spectral image in Fig. 3.1 are different from those in Fig. 2.5 since the image formation models are different. However, PSFs are the same since the design parameters of the PS, and the measurement settings, such as wavelengths, the technique to achieve measurement diversity, are the same.

We train our network using the same solar dataset as Chapter 2, i.e., solar EUV images from the NASA SDO database [53]. We use SDO images from the 1^{st}, 5^{th}, 10^{th}, 15^{th}, 25^{th}, and 30^{th} days of every month for years between 2012 and 2020, to have training data diversity. After removing the images with artifacts (such as out-of-focus), we randomly pick 6 images from each day. These images are then converted to grayscale, resized to 256×256, and normalized between $[0, 1]$. Finally, we combine randomly chosen three images to form 725 training data (for $P = 3$ case). In a similar way, we have generated separate 105 test data. For the training, we first simulate the measurements $\mathbf{y}$ at the SNR of 25 dB using the forward model in Eq. (3.2). Then, we train our networks using ground truth spectral images and these measurements. The network is trained by minimizing the l_2 loss between the ground truth and reconstructed images using Adam solver [57]. To avoid memory issues, we choose L as 10. The conjugate gradient method runs for 5 iterations. Codes are implemented with Tensorflow on Nvidia GeForce RTX 2080 Ti GPU. Training takes approximately 22 hours.

First, we analyze the performance of the developed and TV methods for input scenes with different characteristics. The spectral test data numbered as 1 to 5 in Table 3.1 have increasing levels of abrupt changes in space and spectrum. We perform 5 Monte Carlo runs and compute the average PSNR and SSIM values between the diffraction-limited ground truth images and the reconstructed spectral images for each test data. As seen, the developed method has better generalization capability. It can provide 2 dB improvement in average PSNR and 0.02 improvement in average SSIM. The results also illustrate the effect of the training data. Our training set contains images more similar to test data 1 & 2 than test data 3, which seems to affect the performance of the developed learning-based method. Moreover, TV reconstruction does not per-

Table 3.1: Average PSNR (dB), SSIM and runtime (second) values for TV and learned reconstructions for different spectral test data (with best values shown in bold).

Method	TV Recon.	Learned Recon.
Test Data 1	41.63/0.96/559	**42.98/0.97/0.4**
Test Data 2	37.44/0.95/902	**40.72/0.96/0.4**
Test Data 3	35.29/0.90/512	**35.66/0.92/0.4**
Test Data 4	34.76/0.93/1025	**34.91/0.95/0.4**
Test Data 5	29.88/0.90/876	**34.73/0.95/0.4**
Average	35.80/0.93/775	**37.80/0.95/0.4**

form well for test data 5, which is more challenging to reconstruct due to increased structure and larger changes in space and spectrum. We also give runtime for each test data in Table 3.1. TV reconstruction runs until the ratio of the Frobenius norm of the change in the image to the Frobenius norm of the image becomes less than 0.00001. As seen, the learned method reconstructs images only in 0.4 seconds and is around 1937-fold faster than TV reconstruction. Hence learned reconstruction not only outperforms TV reconstruction in terms of reconstruction quality but is also significantly faster than TV reconstruction. This is desirable for real-time applications.

To visually evaluate the results for images with different amount of structure, we provide the reconstructed spectral images for test data 2 and 5 in Fig. 3.2 and 3.3, respectively. For comparison, the diffraction-limited versions of the original images are also given. Visual inspection shows that important characteristics, such as solar flares, are successfully recovered. The learned reconstruction method is more successful in this respect than the TV reconstruction, which introduces some blur for the rapidly varying spatial structures.

We also analyze the reconstruction performance of the learned reconstruction with a larger test data that contains randomly selected 100 spectral test data. Fig. 3.4 depicts histograms PSNR/SSIM values for these 100 reconstructions.As seen, learned reconstruction performs significantly better than TV reconstruction without sacrificing generalization capability with a mean PSNR of 36.03 dB and SSIM of 0.93.

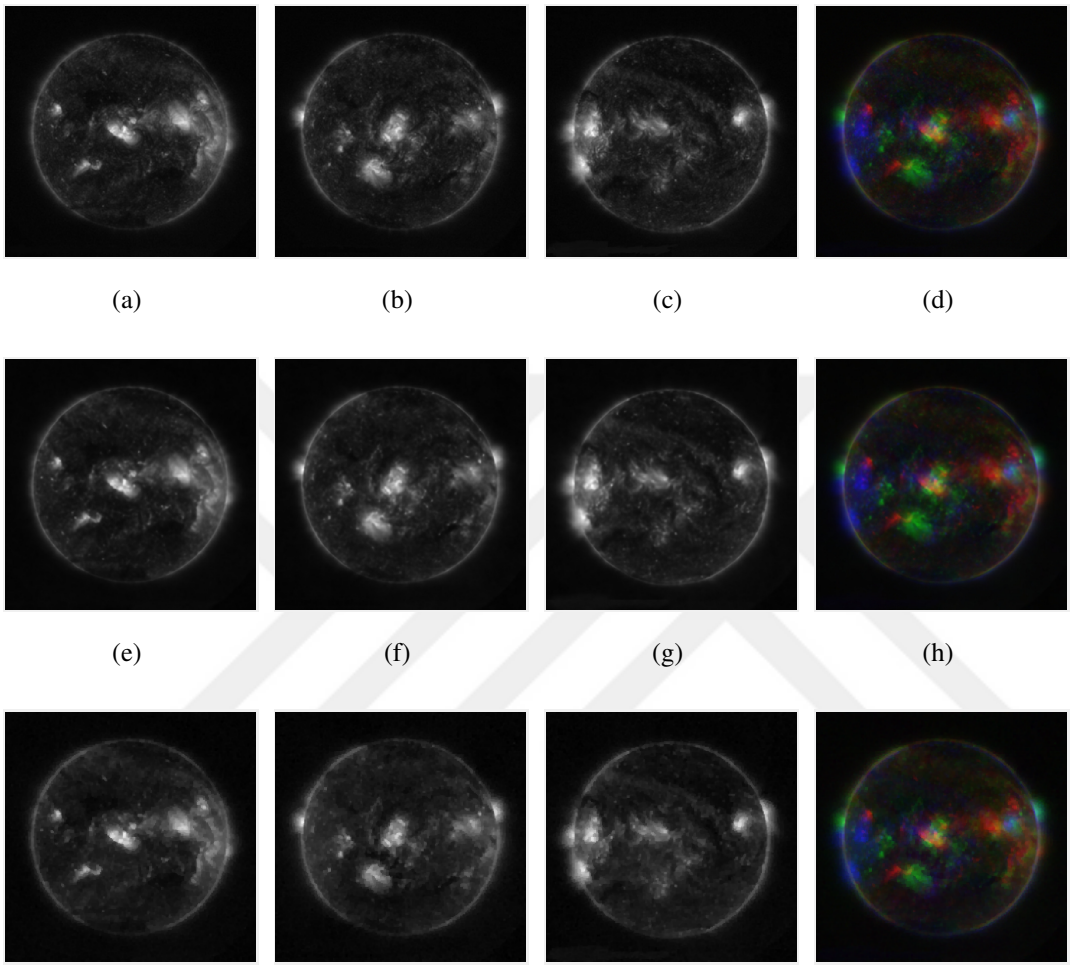


Figure 3.2: Top to bottom: diffraction-limited images, deep learning-based unrolled reconstruction, TV reconstruction for test data 2 (with false-color representations on the right).

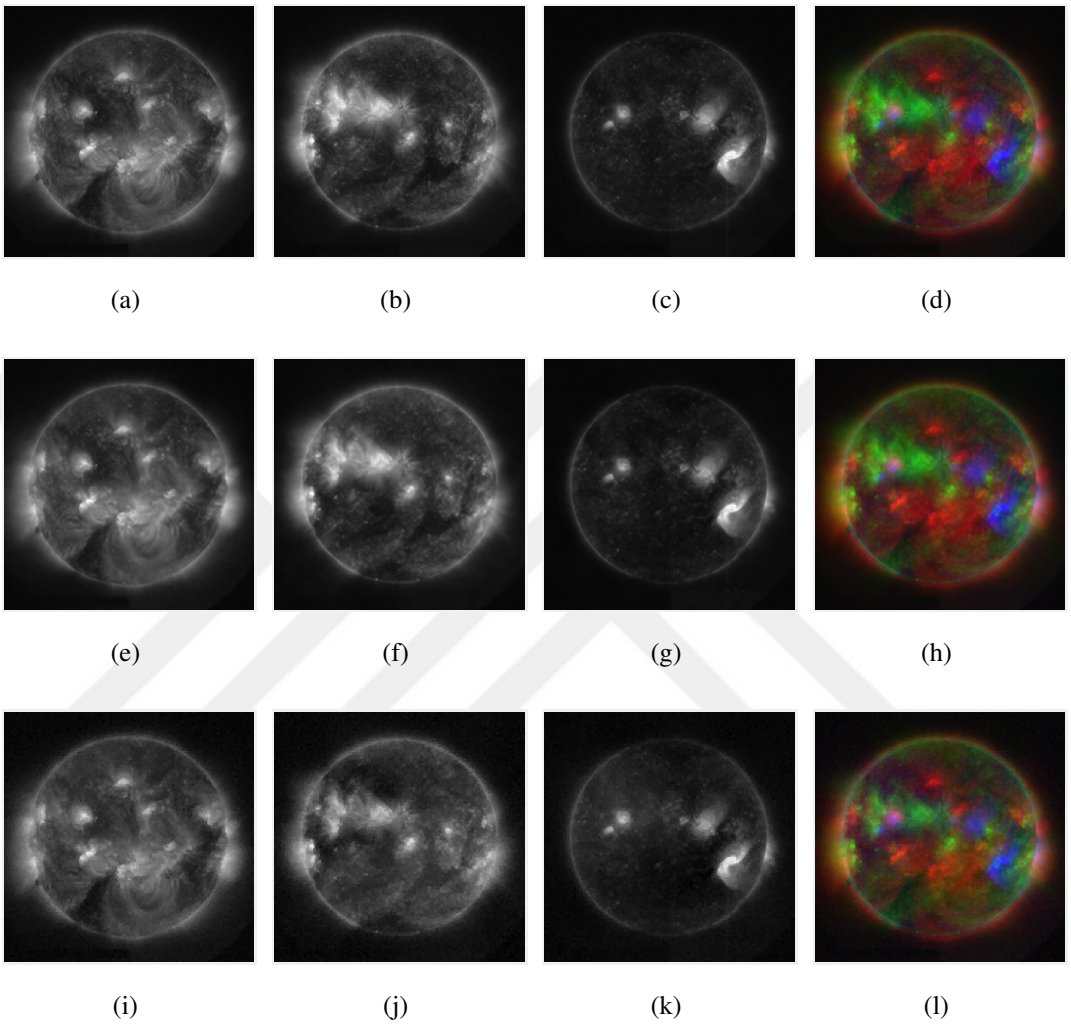
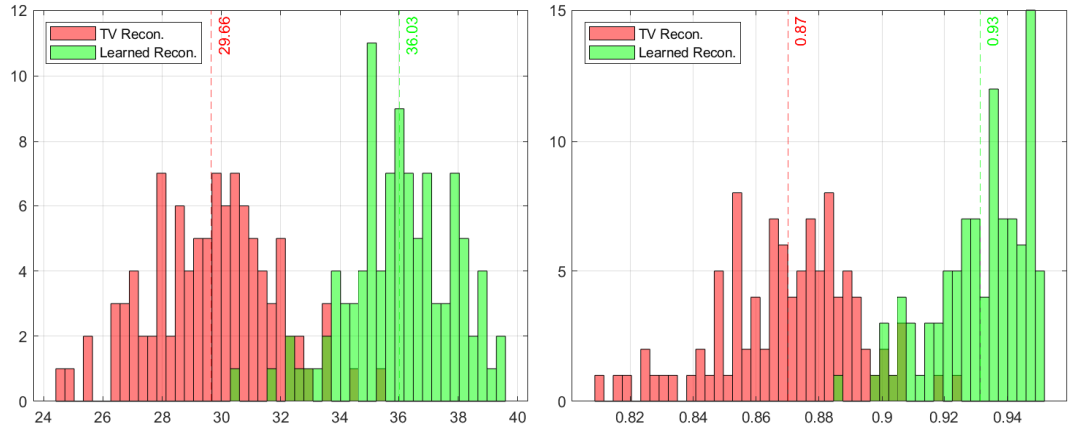


Figure 3.3: Top to bottom: diffraction-limited images, deep learning-based unrolled reconstruction, TV reconstruction for test data 5 (with false-color representations on the right).



(a) (b)

Figure 3.4: PSNR (a) and SSIM (b) histograms of TV and learned reconstructions. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively.

3.5.2 Model Mismatch Analysis for the Multiple Measurement Multiple Image Case

Here, we investigate the possible performance reductions due to forward model mismatch. Note that this analysis is different than that of in Chapter 2. In Chapter 2, the cause of the mismatch is the discrepancies between the discretization along spectral dimension for training and testing images. Here, on the other hand, the mismatch is due to the forward model discrepancies between training and testing images. This analysis provides us insights into how the performance will decrease if we use the image formation model in Chapter 2 instead of the truncated image formation model. To this end, we use the Unrolled-I network trained for the SNR of 25 dB in Chapter 2 and analyze its performance on the test data simulated using the truncated image formation model in Eq. (3.2). Remember that detectors have finite support and can only acquire the system's response in a limited region. In Chapter 2, support of the spectral images is assumed to be less than that of measurements, such that detectors can capture the convolution results fully. This approach effectively assumes zero end conditions for the spectral images and generally is not compatible with real-world imaging systems.

For the analysis, we perform 5 Monte Carlo runs and compute the average PSNR and SSIM values for the test data 1 to 5. To visually evaluate the results, we provide the reconstructed spectral images for test data 4 in Fig. 3.5. This figure shows that when there exists a model mismatch due to omitting the effect of truncation, the reconstruction performance is significantly decreased, i.e., the unrolled reconstruction cannot capture spectral changes and reconstruct fine details accurately, which can be seen better in false-color representations. Furthermore, as seen from Table 3.2, we have around 4.5 dB decrease in the average PSNR and 0.05 decrease in the average

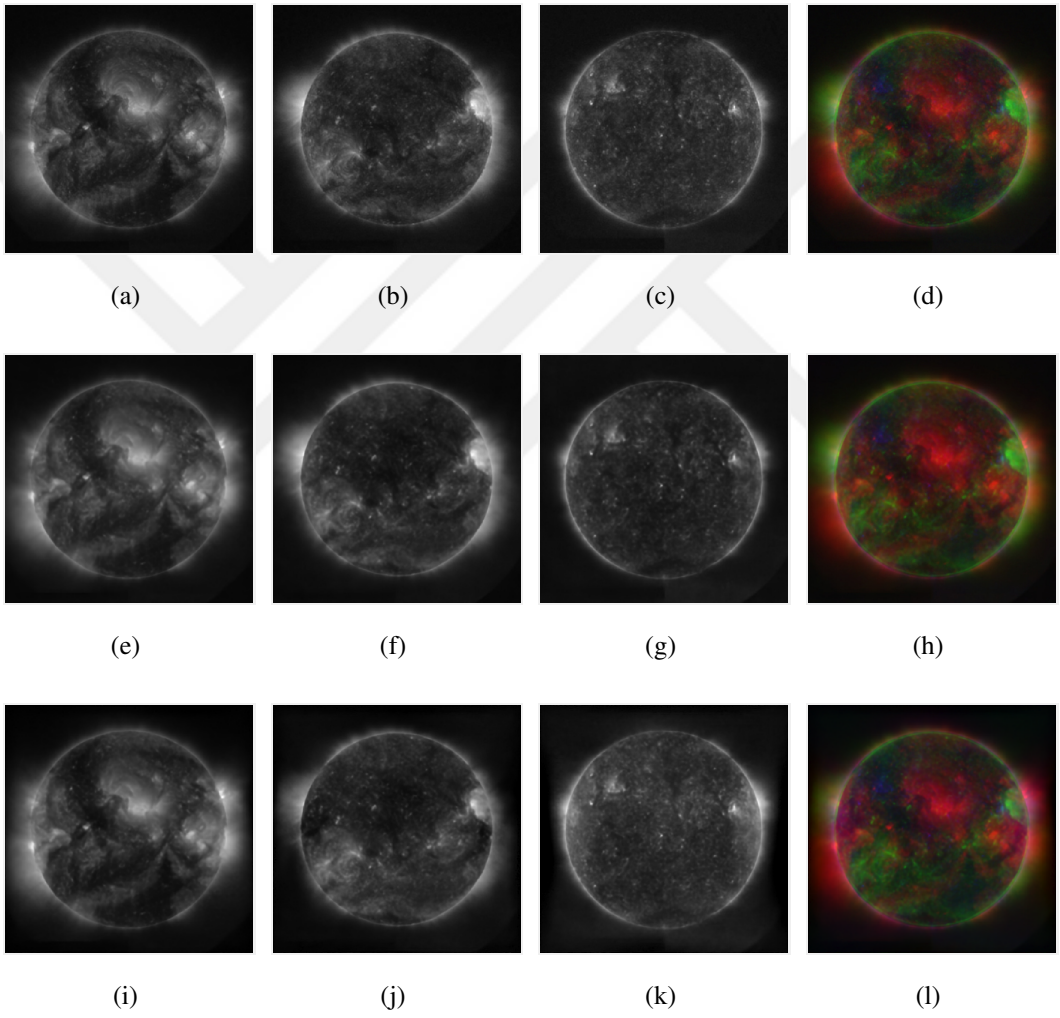


Figure 3.5: Top to bottom: diffraction-limited images, reconstructions without model mismatch, reconstructions with model mismatch for test data 4 (with false-color representations on the right).

Table 3.2: Average PSNR (dB) and SSIM values for unrolled reconstruction method with and without model mismatch for different spectral test data.

Setting	With Mismatch	Without Mismatch
Test Data 1	39.36/0.94	42.98/0.97
Test Data 2	39.64/0.93	40.72/0.96
Test Data 3	27.23/0.86	35.66/0.92
Test Data 4	30.40/0.90	34.91/0.95
Test Data 5	30.09/0.89	34.73/0.95
Average	33.34/0.90	37.80/0.95

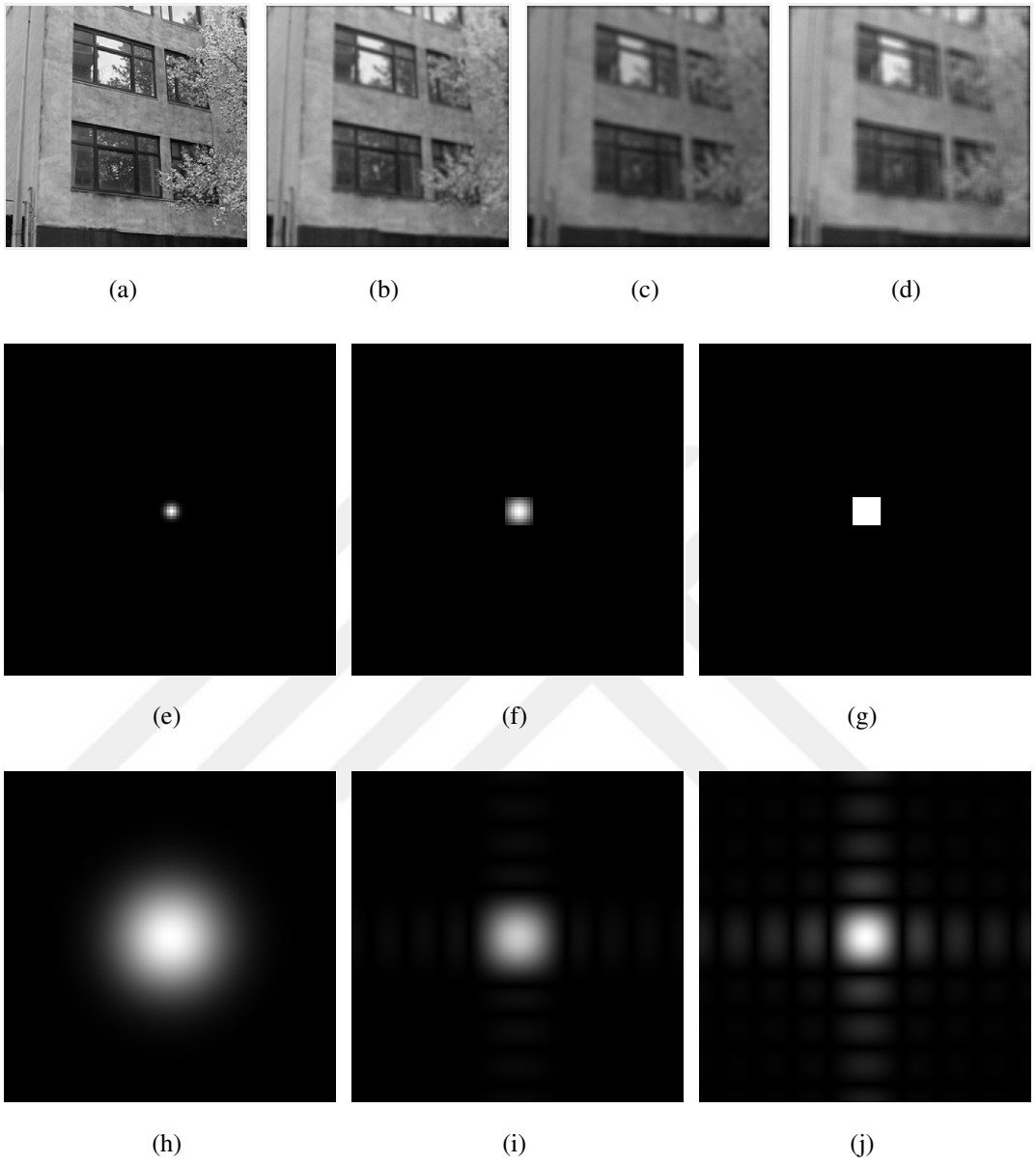
SSIM values when there is such model mismatch. The performance decrease is most significant for test data 3 since our training set contains images less similar to test data 3, affecting its learning ability to reconstruct such spectral images.

3.5.3 Performance Analysis for Single-Frame Deconvolution

In the single-frame deconvolution, we have one input image and one measurement. The measurement is the blurred and noisy version of the input image, as described in Eq. (3.4).

We train the networks using 725 natural images from the Berkeley segmentation dataset (BSD) [63], validation set of the ImageNet database [64], and Waterloo exploration database [65]. The images are of size 256×256 and have pixel values between 0 to 1. For the developed deep learning-based unrolled reconstruction methods, we first simulate the measurements $\mathbf{y}$ at the SNR of 25 dB using the forward model in Eq. (3.4). For the PSFs, we use a Gaussian blur with a variance of 2, a Gaussian blur with a variance of 8, and a uniform blur. The blurs have 9×9 support. Sample measurements, sampled versions of the blurs, and magnitudes of the Fourier transforms of the blurs are shown in Fig. 3.6. We then train our networks by minimizing the l_1 loss between the ground truth and reconstructed images using Adam solver [57]. To be compatible with EUV solar imaging case, we choose L as 10. Codes are implemented with Tensorflow on Nvidia GeForce RTX 2080 Ti GPU. Training takes

approximately 9 hours. Once the networks are trained, reconstructions are obtained simply by feeding the networks with the available measurements.



(a) (b) (c) (d)

(e) (f) (g)

(h) (i) (j)

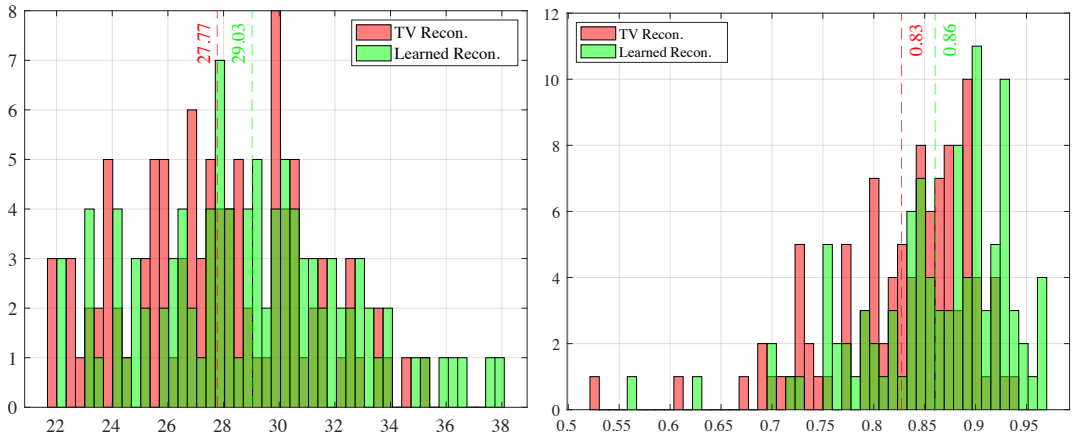
Figure 3.6: Sample image (a), sample measurements at 25 dB SNR for the Gaussian blur with a variance of 2 (b), Gaussian blur with a variance of 8 (c), and uniform blur (d). All blurs have 9×9 support. Sampled and zoomed versions of the Gaussian blur with a variance of 2 (e), Gaussian blur with a variance of 8 (f), and uniform blur (g); magnitudes of the Fourier transforms of the Gaussian blur with a variance of 2 (h), Gaussian blur with a variance of 8 (i), and uniform blur (j).

Table 3.3 presents the average PSNR and SSIM values over 100 test images. As seen, for all blurs, developed learned reconstruction performs better than TV reconstruction. Fig. 3.6 shows that when the variance of the Gaussian blur increases, images become more blurry. Therefore, the inverse problem becomes more challenging. Hence, both learned and analytical reconstruction performances decrease as the variance of Gaussian blur increases. Average reconstruction PSNR and SSIM values of the learned reconstruction decrease around 3 dB and 0.1 when the Gaussian blur variance increases from 2 to 8. Similarly, the average reconstruction PSNR and SSIM values of the TV reconstruction decrease around 3 dB and 0.08, respectively.

Table 3.3: Average PSNR (dB) and SSIM values for TV and learned reconstructions and for different blurs (with best values shown in bold).

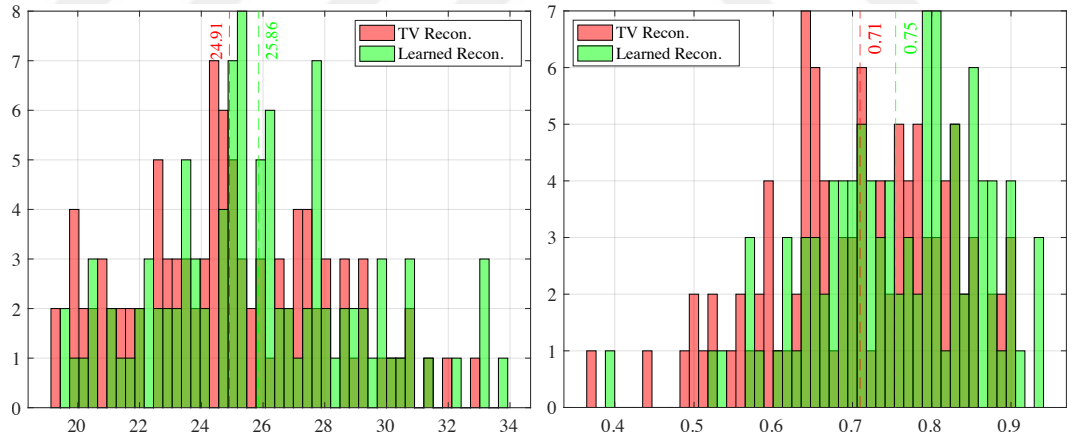
Blur	TV Recon.	Learned Recon.
Gaussian blur with a variance of 2	27.77/0.83	**29.03/0.86**
Gaussian blur with a variance of 8	24.91/0.71	**25.86/0.76**
Uniform blur	25.65/0.74	**26.63/0.78**

Fig. 3.7, 3.8, and 3.9 show the histograms of the reconstruction PSNR/SSIM values for the Gaussian blur with a variance of 2, Gaussian blur with a variance of 8, and uniform blur, respectively. As seen, for all blurs, the learned reconstruction has better generalization capability. The learned reconstruction can also reconstruct more test images with higher PSNR and SSIM values. For instance, for the Gaussian blur with a variance of 2, the developed method can reconstruct 5 test images with PSNR greater than 36. In contrast, the maximum reconstruction PSNR for the TV reconstruction is around 35 dB. Likewise, for the Gaussian blur with a variance of 2, the developed method can reconstruct 5 test images with SSIM greater than 0.95. In contrast, the maximum reconstruction SSIM for the TV reconstruction is around 0.94. Moreover, the minimum reconstruction PSNR is also higher for the learned reconstruction. Similar results are also observed for the other blurs.



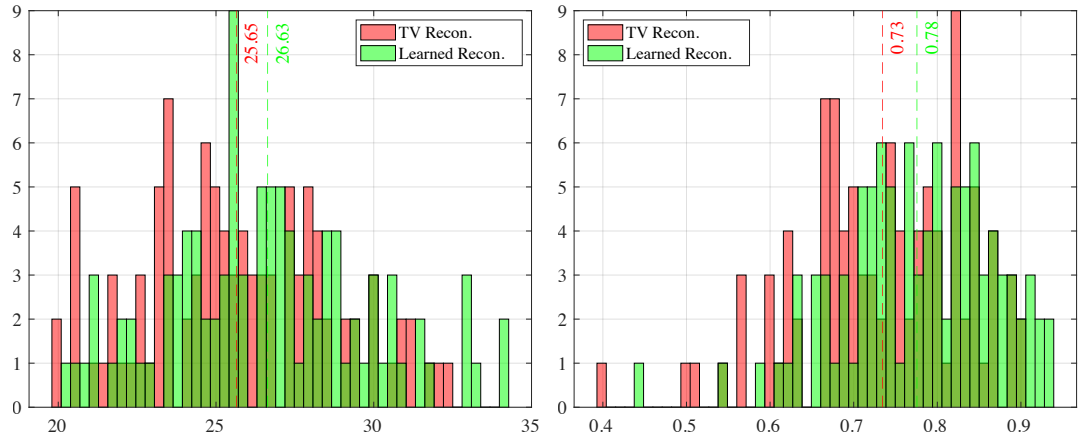
(a) (b)

Figure 3.7: PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for Gaussian blur with a variance of 2. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively.



(a) (b)

Figure 3.8: PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for Gaussian blur with a variance of 8. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively.



(a) (b)

Figure 3.9: PSNR (a) and SSIM (b) histograms of TV and learned reconstructions in single-frame deconvolution for uniform blur. Red and green vertical lines show the mean values for TV and learned reconstructions, respectively.

For visual evaluation, a sample test data and reconstructions are illustrated in Fig. 3.10. This test data is challenging since the texture of the original image is highly complex due to tree branches and windows containing reflections. When we visually evaluate the reconstructions, clearly learned reconstructions are sharper and better than TV reconstructions for all blurs. Especially, tree branches and reflections on windows are better reconstructed. Numerically, there is around 0.4 dB improvement in PSNR and 0.05 improvement in SSIM. Best reconstructions are obtained when the blur is Gaussian with a variance of 2 since the corresponding inverse problem is easier. PSNR and SSIM values for the learned reconstruction are respectively 25.03 dB and 0.80. When the blur is Gaussian with a variance of 8, reconstructions are worse compared to others since the corresponding inverse problem is more challenging, as explained. PSNR and SSIM values for the learned reconstruction are respectively 23.05 dB and 0.67. There is a significant 2 dB decrease in PSNR and 0.13 decrease in the SSIM compared to the Gaussian blur with a variance of 2.



(a)



(b)



(c)



(d)



(e)



(f)



(g)

Figure 3.10: Top to bottom: Sample image (a); TV reconstructions for the Gaussian blur with a variance of 2 (b), Gaussian blur with a variance of 8 (c), uniform blur (d); learned reconstructions for the Gaussian blur with a variance of 2 (e), Gaussian blur with a variance of 8 (f), uniform blur (g). Reconstruction PSNR (dB) and SSIM values from (b) to (g) are respectively $24.67/0.77, 22.60/0.61, 23.35/0.66, 25.03/0.80, 23.05/0.67$, and $23.71/0.71$.

3.6 Discussion

In this section, we discuss an alternative solution approach for the inverse problems associated with the truncated image formation model. The motivation of this alternative solution is to decrease the training time. In Chapter 2, we use the diagonalizable circular convolution matrices. Therefore, the image update step can be calculated using a closed-form solution, and the training takes 13 hours. On the other hand, in Chapter 3, training takes 22 hours. The training time increases due to the truncated formulation model in Eq. (3.13), which requires using a conjugate gradient method for the image update step since we cannot use closed-form solution directly due to huge matrix formation, storing, and inversion costs.

In the alternative solution approach here, each update step can be calculated using a closed-form solution. The main idea is to use diagonalizable circular convolution matrices and multiple auxiliary variables, i.e., applying multiple variable splittings. Let us assume that we have a PSF and image having $M_1 \times M_2$ and $N_1 \times N_2$ supports, respectively. Then, their linear convolution is the same as the $(M_1 + N_1 - 1) \times (M_2 + N_2 - 1)$ point circular convolution. Therefore, we can implement linear convolution by first zero padding PSF and image to appropriate sizes and then applying circular convolution. Using this approach, we replace the linear convolution matrices in Eq. (3.13) with zero padding and circular convolution matrices:

$$\mathbf{y} = \tilde{\mathbf{T}}\tilde{\mathbf{G}}\tilde{\mathbf{S}}\mathbf{x} + \mathbf{w}, \tag{3.21}$$

where $\mathbf{G} = \tilde{\mathbf{G}}\tilde{\mathbf{S}}$. Here, $\tilde{\mathbf{G}} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1) \times P(M_1+N_1-1)(M_2+N_2-1)}$ is a block circulant matrix and contains all circular convolution matrices, representing the circular convolutions with the PSFs. The zero padding matrix is represented with $\tilde{\mathbf{S}} \in \mathbb{R}^{P(M_1+N_1-1)(M_2+N_2-1) \times PN_1N_2}$. The measurements obtained using Eq. (3.13) and (3.21) are the same. We again introduce regularization to the inverse problem and solve it through the minimization of the following cost function:

$$\min_{\mathbf{x}} \; ||\mathbf{y} - \tilde{\mathbf{T}}\tilde{\mathbf{G}}\tilde{\mathbf{S}}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{x}). \tag{3.22}$$

To utilize the diagonalizability of the $\tilde{\mathbf{G}}$ matrix, we need to decouple $\tilde{\mathbf{G}}$, $\tilde{\mathbf{T}}$, and $\tilde{\mathbf{S}}$

matrices. To this end, we apply multiple variable splittings as:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \tilde{\mathbf{T}}\tilde{\mathbf{G}}\tilde{\mathbf{S}}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}, \tag{3.23}$$

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{m}}) = \arg\min_{\mathbf{x},\mathbf{z},\mathbf{m}} ||\mathbf{y} - \tilde{\mathbf{T}}\tilde{\mathbf{G}}\mathbf{m}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}, \tilde{\mathbf{S}}\mathbf{x} = \mathbf{m}, \tag{3.24}$$

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{m}}, \hat{\mathbf{v}}) = \arg\min_{\mathbf{x},\mathbf{z},\mathbf{m},\mathbf{v}} ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{v}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}, \tilde{\mathbf{S}}\mathbf{x} = \mathbf{m}, \tilde{\mathbf{G}}\mathbf{m} = \mathbf{v}. \tag{3.25}$$

Here, the vectors have the following sizes: $\mathbf{y} \in \mathbb{R}^{RN_1N_2}$, $\mathbf{x} \in \mathbb{R}^{PN_1N_2}$, $\mathbf{z} \in \mathbb{R}^{PN_1N_2}$, $\mathbf{m} \in \mathbb{R}^{P(M_1+N_1-1)(M_2+N_2-1)}$, and $\mathbf{v} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1)}$. By using the HQS method, the constrained optimization problem in Eq. (3.25) can be turned into a non-constrained one as follows:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{m}}, \hat{\mathbf{v}}) = \arg\min_{\mathbf{x},\mathbf{z},\mathbf{m},\mathbf{v}} ||\mathbf{y}-\tilde{\mathbf{T}}\mathbf{v}||_2^2+\lambda\Phi(\mathbf{z})+\lambda_1||\mathbf{z}-\mathbf{x}||_2^2+\lambda_2||\mathbf{m}-\tilde{\mathbf{S}}\mathbf{x}||_2^2+\lambda_3||\mathbf{v}-\tilde{\mathbf{G}}\mathbf{m}||_2^2, \tag{3.26}$$

where λ_1, λ_2, and λ_3 are penalty parameters. We can solve this problem by performing minimization over $\mathbf{x}$, $\mathbf{z}$, $\mathbf{m}$, and $\mathbf{v}$ alternatingly as follows:

$$\mathbf{m}^{l+1} = \arg\min_{\mathbf{m}} \lambda_2||\mathbf{m} - \tilde{\mathbf{S}}\mathbf{x}^l||_2^2 + \lambda_3||\mathbf{v}^l - \tilde{\mathbf{G}}\mathbf{m}||_2^2, \tag{3.27a}$$

$$\mathbf{v}^{l+1} = \arg\min_{\mathbf{v}} ||\mathbf{y} - \tilde{\mathbf{T}}\mathbf{v}||_2^2 + \lambda_3||\mathbf{v} - \tilde{\mathbf{G}}\mathbf{m}^{l+1}||_2^2, \tag{3.27b}$$

$$\mathbf{z}^{l+1} = \arg\min_{\mathbf{z}} \lambda\Phi(\mathbf{z}) + \lambda_1||\mathbf{z} - \mathbf{x}^l||_2^2, \tag{3.27c}$$

$$\mathbf{x}^{l+1} = \arg\min_{\mathbf{x}} \lambda_1||\mathbf{z}^{l+1} - \mathbf{x}||_2^2 + \lambda_2||\mathbf{m}^{l+1} - \tilde{\mathbf{S}}\mathbf{x}||_2^2. \tag{3.27d}$$

Since we have a convex optimization problem, the order in which we update the vectors does not affect the solution. The $\mathbf{m}$-update in Eq. (3.27a) has a closed-form solution as:

$$\mathbf{m}^{l+1} = (\lambda_2\mathbf{I} + \lambda_3\tilde{\mathbf{G}}^H\tilde{\mathbf{G}})^{-1}(\lambda_2\tilde{\mathbf{S}}\mathbf{x}^l + \lambda_3\tilde{\mathbf{G}}^H\mathbf{v}^l). \tag{3.28}$$

Here, the circular convolution matrix $\tilde{\mathbf{G}}$ is diagonalizable by Fourier transform, i.e., $\tilde{\mathbf{G}} = \bar{\mathbf{F}}^H\mathbf{\Lambda}\tilde{\mathbf{F}}$ [36] where $\bar{\mathbf{F}} = \mathbf{I}_R \otimes \mathbf{F}_{2D}$ and $\tilde{\mathbf{F}} = \mathbf{I}_P \otimes \mathbf{F}_{2D}$ with $\mathbf{I}_n$ denoting the identity matrix of size $n \times n$ and $\mathbf{F}_{2D}$ denoting the unitary 2D discrete Fourier transform (DFT) matrix. Here $\mathbf{\Lambda}$ is a matrix of $R \times P$ blocks, with each block being

a diagonal matrix whose diagonal consists of the 2D DFT of the corresponding PSF. Inserting this expression to Eq. (3.28), we get:

$$\mathbf{m}^{l+1} = (\lambda_2\mathbf{I} + \lambda_3\tilde{\mathbf{F}}^H\mathbf{\Lambda}^H\bar{\mathbf{F}}\bar{\mathbf{F}}^H\mathbf{\Lambda}\tilde{\mathbf{F}})^{-1}(\lambda_2\tilde{\mathbf{S}}\mathbf{x}^l + \lambda_3\tilde{\mathbf{F}}^H\mathbf{\Lambda}^H\bar{\mathbf{F}}\mathbf{v}^l). \tag{3.29}$$

Since $\mathbf{F}_{2D}$ is the unitary 2D DFT matrix, $\bar{\mathbf{F}}\bar{\mathbf{F}}^H$ equals the identity matrix. Similarly, $\tilde{\mathbf{F}}^H\tilde{\mathbf{F}}$ equals the identity matrix. Therefore, we can rewrite Eq. (3.29) as follows:

$$\mathbf{m}^{l+1} = \left(\tilde{\mathbf{F}}^H(\lambda_2\mathbf{I} + \lambda_3\mathbf{\Lambda}^H\mathbf{\Lambda})\tilde{\mathbf{F}}\right)^{-1}(\lambda_2\tilde{\mathbf{S}}\mathbf{x}^l + \lambda_3\tilde{\mathbf{F}}^H\mathbf{\Lambda}^H\bar{\mathbf{F}}\mathbf{v}^l). \tag{3.30}$$

Using the matrix inversion identity $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$, we get:

$$\mathbf{m}^{l+1} = \tilde{\mathbf{F}}^H(\lambda_2\mathbf{I} + \lambda_3\mathbf{\Lambda}^H\mathbf{\Lambda})^{-1}\tilde{\mathbf{F}}(\lambda_2\tilde{\mathbf{S}}\mathbf{x}^l + \lambda_3\tilde{\mathbf{F}}^H\mathbf{\Lambda}^H\bar{\mathbf{F}}\mathbf{v}^l). \tag{3.31}$$

By distributing $\tilde{\mathbf{F}}$ into the parentheses, we get the following final closed-form solution for the $\mathbf{m}$-update:

$$\mathbf{m}^{l+1} = \tilde{\mathbf{F}}^H(\lambda_2\mathbf{I} + \lambda_3\mathbf{\Lambda}^H\mathbf{\Lambda})^{-1}(\lambda_2\tilde{\mathbf{F}}\tilde{\mathbf{S}}\mathbf{x}^l + \lambda_3\mathbf{\Lambda}^H\bar{\mathbf{F}}\mathbf{v}^l). \tag{3.32}$$

Here the inverse of $\left(\lambda_2\mathbf{I} + \lambda_3\mathbf{\Lambda}^H\mathbf{\Lambda}\right)$ can be computed through a recursive block matrix inversion approach [47]. As previously, multiplication by $\tilde{\mathbf{F}}$ or $\tilde{\mathbf{F}}^H$ corresponds to taking the 2D DFT or inverse DFT of all image slices. Similarly, multiplication by $\bar{\mathbf{F}}$ corresponds to taking the 2D DFT of all measurements. Multiplication with $\mathbf{\Lambda}$ is implemented simply by element-wise multiplication. Multiplication with $\tilde{\mathbf{S}}$ corresponds to zero padding, which is implemented by inserting zeros to the images.

The $\mathbf{v}$-update in Eq. (3.27b) has a closed-form solution as:

$$\mathbf{v}^{l+1} = (\tilde{\mathbf{T}}^H\tilde{\mathbf{T}} + \lambda_3\mathbf{I})^{-1}(\tilde{\mathbf{T}}^H\mathbf{y} + \lambda_3\tilde{\mathbf{G}}\mathbf{m}^{l+1}), \tag{3.33}$$

which can also be written as follows using the expression of $\tilde{\mathbf{G}}$:

$$\mathbf{v}^{l+1} = (\tilde{\mathbf{T}}^H\tilde{\mathbf{T}} + \lambda_3\mathbf{I})^{-1}(\tilde{\mathbf{T}}^H\mathbf{y} + \lambda_3\bar{\mathbf{F}}^H\mathbf{\Lambda}\tilde{\mathbf{F}}\mathbf{m}^{l+1}). \tag{3.34}$$

Here, $\tilde{\mathbf{T}}^H\tilde{\mathbf{T}} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1)\times R(M_1+N_1-1)(M_2+N_2-1)}$ is a diagonal matrix, with its diagonal elements being either 0 or 1. Therefore, $\left(\tilde{\mathbf{T}}^H\tilde{\mathbf{T}} + \lambda_3\mathbf{I}\right)$ is a diagonal matrix, with its diagonal elements being either λ_3 or $\lambda_3 + 1$. Hence, the inverse of the $\left(\tilde{\mathbf{T}}^H\tilde{\mathbf{T}} + \lambda_3\mathbf{I}\right)$ can be calculated easily by element-wise inversion of the diagonal elements. Multiplication with $\tilde{\mathbf{T}}^H$ corresponds to zero padding, which is implemented by inserting zeros to the measurements.

The $\mathbf{z}$-update in Eq. (3.27c) is the same as Eq. (3.17b) and corresponds to a denoising problem. Therefore, again, we obtain its solution using the 2D U-Net in [48].

Finally, $\mathbf{x}$-update, i.e., image update in Eq. (3.27d), has a closed-form solution as:

$$\mathbf{x}^{l+1} = (\lambda_1 \mathbf{I} + \lambda_2 \tilde{\mathbf{S}}^H \tilde{\mathbf{S}})^{-1} (\lambda_1 \mathbf{z}^{l+1} + \lambda_2 \tilde{\mathbf{S}}^H \mathbf{m}^{l+1}). \tag{3.35}$$

When we multiply the image with $\tilde{\mathbf{S}}^H \tilde{\mathbf{S}}$, first, the image is zero padded and then truncated. Therefore, after the multiplication, we get the image itself back. Hence, $\tilde{\mathbf{S}}^H \tilde{\mathbf{S}} \in \mathbb{R}^{(PN_1N_2)\times(PN_1N_2)}$ is equal to the identity matrix. Thus, we can rewrite Eq. (3.35) as follows:

$$\mathbf{x}^{l+1} = ((\lambda_1 + \lambda_2)\, \mathbf{I})^{-1} (\lambda_1 \mathbf{z}^{l+1} + \lambda_2 \tilde{\mathbf{S}}^H \mathbf{m}^{l+1}), \tag{3.36}$$

The inverse of the $(\lambda_1 + \lambda_2)\, \mathbf{I}$ can be calculated easily by element-wise inversion of the diagonal elements. Multiplication with $\tilde{\mathbf{S}}^H$ corresponds to the truncation, which can be implemented in the spatial domain by eliminating some entries.

Overall, Eq. (3.32), (3.34), and (3.36) have closed-form solutions, which can be computed without forming, storing, and inverting huge matrices. Therefore, the inverse problem of the truncated image formation model is solved efficiently without using the conjugate gradient method. After fixing the iteration number L, we unroll the algorithm described by Eq. (3.27a), (3.27b), (3.27c), and (3.27d) into an end-to-end trainable network and learn the parameters in this structure.

In this alternative solution, we have three auxiliary variables and in the solution using the conjugate gradient method, we have one auxiliary variable. Therefore, optimization of the network using the alternative solution is more challenging since it requires optimization with respect to four variables. In training, we observed more iterations per second for the network using the alternative solution than the network using the conjugate gradient method. This indicates faster computation both for the forward model and backpropagation, which can accelerate the training. However, the training loss may not converge, i.e., initially, it is decreasing, but after some epochs, it is increasing and diverging. Decreasing the learning rate can prevent this behavior. However, when the learning rate is decreased, more epochs are needed, which increases the training time. Hence, decreasing the learning rate is not feasible since our motivation is to reduce the training time. Early stopping the training before the

divergence is another option. However, this may lead to a sub-optimum solution. Due to this convergence problem, we do not include results obtained using this network. However, it may have the potential for faster training once the convergence issue is resolved.

3.7 Conclusion

In this chapter, we formulated a truncated image formation model that can represent various two and multidimensional imaging modalities. The formulated image formation model is compatible with real-world imaging systems. Through the mismatch analysis for EUV solar imaging, we showed the performance gains obtained using the truncated image formation model.

We developed a deep learning-based unrolled reconstruction method to solve the inverse problems associated with the truncated image formation model. The developed unrolled method uses the forward model during training. As a result, the network learns not only denoising but also artifact cleaning. As expected, unrolled reconstruction performs significantly better than TV reconstruction and has better generalization capability. Although training time is long for the unrolled reconstruction, testing is much faster than TV reconstruction, as is desirable in practical settings. To illustrate the utility and versatility, we apply the unrolled reconstruction to both 2D and 3D imaging problems. For the 2D single-frame deconvolution, the unrolled reconstruction outperforms TV reconstruction for various Gaussian and uniform blurs. Similarly, for the 3D multi-spectral imaging problem, unrolled reconstruction performs better than TV reconstruction for various spectral data with significantly different spatial features.

We also developed an alternative solution for the inverse problems associated with the truncated image formation model. In the developed deep learning-based unrolled reconstruction, we used the conjugate gradient method for the image update step since the image update step cannot be calculated using a closed-form solution due to huge matrix formations and inversions. However, using the conjugate gradient method increases the training time. The motivation of the alternative solution is to decrease the

training time by avoiding the use of the conjugate gradient method. To this end, the alternative solution applies multiple variable splittings, such that each sub-problem can be solved using a closed-form solution. Our training results show that the network can perform more iterations per second. Therefore, it has the potential for faster training. However, training loss may not converge. Resolving the converge issue will be a focus of our future research.

Finally, the performance of unrolled reconstruction methods can be further improved. In training for single-frame deconvolution, we use 725 training data and 10 unrolling to be compatible with EUV solar imaging. However, we have access to more natural images to be used in training. Moreover, we can further increase the unrolling number since the single-frame deconvolution is a 2D application with less memory requirement than the 3D EUV solar imaging application. If we use more training data and increase the number of unrolling, the performance of the unrolled reconstruction for single frame deconvolution may increase. Performance may also be improved using other optimization methods or more inner iterations, such as using 10 iterations instead of 5 iterations in the conjugate gradient method. However, using more inner iterations will also increase the training time. Lastly, we may improve the reconstruction performance by using an application-specific denoiser network instead of a generic U-Net in [48] and using hybrid or different loss functions during training.

CHAPTER 4

DEEP LEARNING-BASED UNROLLED RECONSTRUCTION FOR COMPRESSIVE SPECTRAL IMAGING WITH DIFFRACTIVE LENSES

4.1 Introduction

As mentioned before, spectral imaging is a useful diagnostics technique with widespread application in remote sensing, astronomy, physics, chemistry, biology, and industry. However, capturing the three-dimensional (3D) spectral data cube using 2D detectors impose inherent challenges. Traditional methods capture this spectral data by scanning the scene either spatially or spectrally, which generally cause long acquisition times, low signal-to-noise ratios, and low spatial or spectral resolutions. Computational spectral imaging aims to overcome these limitations by incorporating computation [6, 10]. In particular, compressive spectral imaging enables the reconstruction of the entire 3D spectral data cube from a few superimposed 2D images by solving an inverse problem. Several compressive spectral imaging modalities enabling reduced number of measurements (i.e., reduced acquisition time) have been proposed such as coded aperture snapshot spectral imaging (CASSI) [11] and its variants [10, 12, 13], and *compressive spectral imaging with diffractive lenses* (CSID) [9].

In this chapter, we develop novel deep learning-based unrolled reconstruction methods for the CSID [9] system. The CSID system, illustrated in Fig. 4.1, has a simple optical configuration with a coded aperture and a diffractive lens. Different than CASSI systems that use gratings/prisms to disperse the optical field and require ad-

Some parts of this chapter have been presented in [37].

ditional collimating/re-imaging optics, CSID system uses a diffractive lens both for focusing and dispersing the optical field. The focusing behavior of a diffractive lens is wavelength-dependent. As a result, the coded measurements obtained with the diffractive lens are the superposition of coded and differently blurred components. Hence, the goal in the inverse problem is to reconstruct the entire spectral data cube from these superimposed, coded and blurred measurements. We develop unrolled reconstruction methods with a data driven 3D prior to solve this problem. For this, we first establish the forward model using both a simplified model, like the one in Chapter 2, and a realistic truncated model involving truncated convolutions, like the one in Chapter 3. Then, we formulate the inverse problem as an optimization problem and split it into two sub-problems using the half quadratic splitting (HQS) method. We efficiently solve the sub-problem containing the data fidelity term using efficient steepest descent and conjugate gradient implementations with fast Fourier transforms (FFTs). Because we need to exploit spectral correlations in addition to spatial correlations, we use a 3D CNN-based prior to solve the denoising sub-problem.

Recently, deep learning-based reconstruction has been successfully applied to various inverse problems in imaging [26,27,39], including compressive spectral imaging [34, 35, 66–69]. These methods rely on direct end-to-end reconstruction [66, 67], plug-and-play regularization [68], and unrolling [34,35]. Our method differs from [66–68] since we rely on unrolling. Different than other unrolled reconstruction methods [34, 35] developed for different compressive architectures than CSID, we here employ a 3D network architecture that performs 3D convolutions and processing instead of their 2D counterparts so that the data driven prior can exploit both spatial and spectral correlations. Furthermore, we explicitly include the effect of truncation for the first time, to the best of our knowledge, into the model to simulate the real-world scenario since the detectors have finite support and can only acquire measurements in a limited region. We also show how to efficiently compute each step in the reconstruction algorithm without forming, storing or, inverting any matrices.

For a sample imaging application, we comparatively evaluate the performance of the developed reconstruction methods, which shows superior results compared to analytical priors. The developed methods are also applicable to similar multidimensional imaging settings that involve the reconstruction of an image cube from its super-

imposed, coded, and blurred measurements. We also perform a model mismatch analysis to illustrate the significance of the truncated model. In our discussions, we use deep learning-based unrolled reconstruction, learned reconstruction, and unrolled reconstruction interchangeably.

4.2 Forward Model

Fig. 4.1 illustrates the CSID system [9]. The optical field is first passed through an imaging lens and a coded aperture for spatial modulation, then passed through a diffractive lens for focusing and dispersion, and finally measured with a monochrome detector. The focusing behavior of the diffractive lens is wavelength-dependent. Hence, each spectral component is exposed to a different blur. As a result, each measurement is a superposition of differently blurred spectral components. For measurement diversity, one can also take multiple measurements using different optical setups such as by fixing the detector distance, i.e., d_i, and changing the diffractive lens design. We present the simplified and truncated models in the following sections.

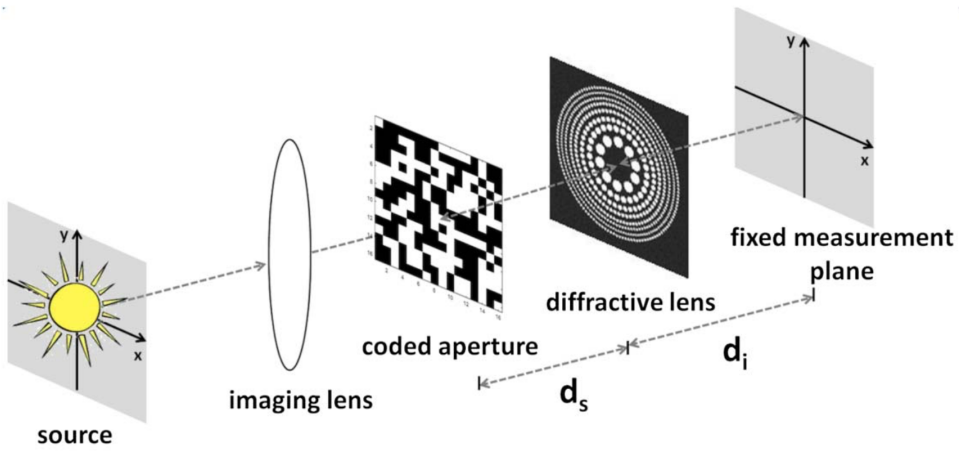


Figure 4.1: CSID System.

4.2.1 Simplified Model with Zero End Conditions

For the simplified model, measurements can be expressed as:

$$y_r[m,n] = \sum_{p=1}^{P} (c_p[m,n]x_p[m,n]) * g_{r,p}[m,n] + w_r[m,n]. \tag{4.1}$$

Here, $y_r[m,n], x_p[m,n], g_{r,p}[m,n], w_r[m,n]$ respectively denote the r^{th} measurement, the p^{th} spectral image, the point spread function (PSF) of the diffractive lens acting to the p^{th} spectral component in the r^{th} measurement, and additive white Gaussian noise in the r^{th} measurement. We have $r = 1, 2, \ldots, R$ and $p = 1, 2, \ldots, P$, i.e., total of R measurements and P spectral images with $R \ll P$. The exact forms of the PSFs $h_{r,p}[m,n]$ can be found elsewhere [36]. Coded mask, denoted by $c_p[m,n]$, has entries 1 or 0, drawn from Bernoulli distribution, and can be same or different for each spectral band. The symbol $*$ represents linear convolution. Measurement diversity can be achieved by changing the focusing behavior of the diffractive lens. In the simplified model, we assume that the support of the spectral images is slightly smaller than the detector range covered by $N_1 \times N_2$ pixels. This assumption corresponds to the zero end condition. Let the PSF $g_{r,p}[m,n]$ has $M_1 \times M_2$ support, i.e., $g_{r,p}[m,n] = 0$ for $m \notin [0, M_1 - 1], n \notin [0, M_2 - 1]$. The support of the spectral images is $(N_1 - M_1 + 1) \times (N_2 - M_2 + 1)$, i.e., $x_p[m,n] = 0$ for $m \notin [0, N_1 - M_1], n \notin [0, N_2 - M_2]$. Then, we can replace the linear convolution in Eq. (4.1) with $N_1 \times N_2$ point circular convolution, which will be useful to develop fast image reconstruction methods.

Since all the operators in Eq. (4.1) are linear, discrete image formation model can be formulated in the following matrix-vector form using appropriate lexicographic

ordering:

$$\mathbf{y} = \mathbf{GCx} + \mathbf{w},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \dots & \mathbf{G}_{1,P} \\ \vdots & & \vdots \\ \mathbf{G}_{R,1} & \dots & \mathbf{G}_{R,P} \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_R \end{bmatrix}, \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_R \end{bmatrix}. \tag{4.2}$$

Here, $\mathbf{G} \in \mathbb{R}^{(RN_1N_2)\times(PN_1N_2)}$ is a block circulant matrix and contains circular convolution matrices $\mathbf{G}_{r,p}$, each representing the circular convolution with the PSF $g_{r,p}[m,n]$. The vector $\mathbf{x} \in \mathbb{R}^{PN_1N_2}$ is the vectorized spectral cube with zero end conditions. Similarly, vectors $\mathbf{y} \in \mathbb{R}^{RN_1N_2}$ and $\mathbf{w} \in \mathbb{R}^{RN_1N_2}$ are respectively the vertically concatenated measurement and noise vectors. The diagonal masking matrix $\mathbf{C} \in \mathbb{R}^{(PN_1N_2)\times(PN_1N_2)}$, representing the effect of the coded aperture, has values either 0 or 1 along its diagonals.

4.2.2 Truncated Model

For the truncated model, measurements can be expressed as:

$$y_r[m,n] = \mathcal{T}\left(\sum_{p=1}^{P} (c_p[m,n]x_p[m,n]) * g_{r,p}[m,n]\right) + w_r[m,n]. \tag{4.3}$$

The truncation operation denoted by $\mathcal{T}$, truncates the center part of the linear convolution result so that the sizes of the recorded measurements and the spectral images to be reconstructed are the same. Hence we take into account the fact that the detectors have finite support and can only acquire the system's response in a limited region. With this, we do not make any unrealistic assumptions about the values (i.e., end conditions) of the unknown spectral images outside the detected region. In fact, our truncated model formulates the measurements regardless of the end conditions, resulting in a better representation of the physical reality. Again, the measurement diversity can be achieved by changing the focusing behavior of the diffractive lens. We

have P spectral components, i.e., p ranges from 1 to P, and we take R measurements, i.e., r ranges from 1 to R.

Let the PSF $g_{r,p}[m,n]$ has $M_1 \times M_2$ support, and the detector covers $N_1 \times N_2$ pixels. Since all the operators in Eq. (4.3) are linear, discrete image formation model can be formulated in the following matrix-vector form using appropriate lexicographic ordering:

$$\mathbf{y} = \tilde{\mathbf{T}}\mathbf{G}\mathbf{C}\mathbf{x} + \mathbf{w},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \dots & \mathbf{G}_{1,P} \\ \vdots & & \vdots \\ \mathbf{G}_{R,1} & \dots & \mathbf{G}_{R,P} \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_P \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_R \end{bmatrix}, \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_R \end{bmatrix}, \tag{4.4}$$

$$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{T} & \dots & \mathbf{0} \\ \vdots & & \vdots \\ \mathbf{0} & \dots & \mathbf{T} \end{bmatrix}.$$

Here, $\mathbf{G} \in \mathbb{R}^{R(M_1+N_1-1)(M_2+N_2-1)\times PN_1N_2}$ contains linear convolution matrices $\mathbf{G}_{r,p}$, each representing the linear convolution with the PSF $g_{r,p}[m,n]$. The vector $\mathbf{x} \in \mathbb{R}^{PN_1N_2}$ is the vectorized spectral cube. Similarly, vectors $\mathbf{y} \in \mathbb{R}^{RN_1N_2}$ and $\mathbf{w} \in \mathbb{R}^{RN_1N_2}$ are respectively the vertically concatenated measurement and noise vectors. Again, the diagonal masking matrix $\mathbf{C} \in \mathbb{R}^{(PN_1N_2)\times(PN_1N_2)}$, representing the effect of the coded aperture, has values either 0 or 1 along its diagonals. Lastly, $\tilde{\mathbf{T}} \in \mathbb{R}^{RN_1N_2\times R(M_1+N_1-1)(M_2+N_2-1)} = \mathbf{I}_R \otimes \mathbf{T}$ is the overall truncation matrix with $\mathbf{I}_n$ denoting the identity matrix of size $n \times n$, $\otimes$ denoting the Kronecker product, and $\mathbf{T}$ denoting the matrix representation of the truncation operator $\mathcal{T}$, for a single measurement.

4.3 Inverse Problem

In the inverse problem, the goal is to recover the unknown spectral images, $\mathbf{x}$, from the measurements, $\mathbf{y}$, of the CSID system. This inverse problem can be considered as a multi-frame deconvolution problem where the measurements are in the form of superimposed blurred images. That is, each measurement is a superposition of focused or defocused versions of different spectral images. Considering Eq. (4.2) and (4.4), we can write the general matrix-vector form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}. \tag{4.5}$$

For the simplified and truncated models, the exact forms of the $\mathbf{A}$ matrices are different. In the following discussion, we continue with the general formulation in Eq. (4.5) and replace it with the exact form whenever necessary. The problem in Eq. (4.5) inherently ill-posed; as the PSFs, $g_{r,p}[m,n]$, of different wavelengths, p, and measurements, r, become more similar (for example, when the difference between different wavelengths decreases), the conditioning of the problem gets worse due to the increased dependency of the columns and rows of $\mathbf{G}$, respectively. Therefore, we solve the inverse problem through the minimization of the following cost function with regularization:

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{x}). \tag{4.6}$$

The first term controls the consistency of the forward model with the measurements, and the second term controls the consistency of the solution with our prior knowledge where $\Phi(\mathbf{x})$ is the spectral image prior. The parameter λ determines the trade-off between the data fidelity and prior terms.

4.4 Developed Reconstruction Methods

Here, we develop deep learning-based unrolled reconstruction methods to solve the inverse problems associated with the simplified and truncated models. For an efficient and effective reconstruction method, we need to decouple the prior and the data-fidelity terms in Eq. (4.6) such that a data-driven deep-prior can be easily incorporated into the reconstruction. For this, we first reformulate Eq. (4.6) by introducing the

auxiliary variable $\mathbf{z} \in \mathbb{R}^{PN_1N_2}$:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) \text{ subject to } \mathbf{x} = \mathbf{z}. \tag{4.7}$$

Using the HQS method, this constrained optimization problem can be turned into a non-constrained one as:

$$(\hat{\mathbf{x}}, \hat{\mathbf{z}}) = \arg\min_{\mathbf{x},\mathbf{z}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}||_2^2, \tag{4.8}$$

where ν is a penalty parameter. We can solve Eq. (4.8) by performing minimization over $\mathbf{x}$ and $\mathbf{z}$ alternatingly as follows:

$$\mathbf{z}^l = \arg\min_{\mathbf{z}} \lambda\Phi(\mathbf{z}) + \nu||\mathbf{z} - \mathbf{x}^l||_2^2, \tag{4.9a}$$

$$\mathbf{x}^{l+1} = \arg\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \nu||\mathbf{z}^l - \mathbf{x}||_2^2, \tag{4.9b}$$

Eq. (4.9a) and (4.9b) respectively correspond to a denoising problem (also the proximal operator of the spectral image prior) and a least squares problem. Therefore, data fidelity and prior terms are decoupled into two sub-problems. For both models, the denoising problem in Eq. (4.9a) is solved using the 3D U-Net architecture in [70] ($\mathbf{D}(.)$) with 4 levels, i.e., $\mathbf{z}^l = \mathbf{D}(\mathbf{x}^l)$. The network architecture is shown in Fig. 4.2. 3D U-Net is chosen since this deep prior implicitly exploits both spatial and spectral correlations through 3D filters and processes the spectral cube at different scales with the multiresolution structure of the network. This network is an extension of the 2D U-Net [48] to 3D by replacing the convolutions, max pooling, ReLU, and interpolation layers with their 3D counterparts.

The least-squares problem in Eq. (4.9b) for the image update step has a closed-form solution:

$$\mathbf{x}^{l+1} = \left(\mathbf{A}^H\mathbf{A} + \nu\mathbf{I}\right)^{-1}\left(\mathbf{A}^H\mathbf{y} + \nu\mathbf{z}^l\right). \tag{4.10}$$

Exact form and solution of the image update step change for simplified and truncated models. Therefore, they examined separately in detail in the following sections.

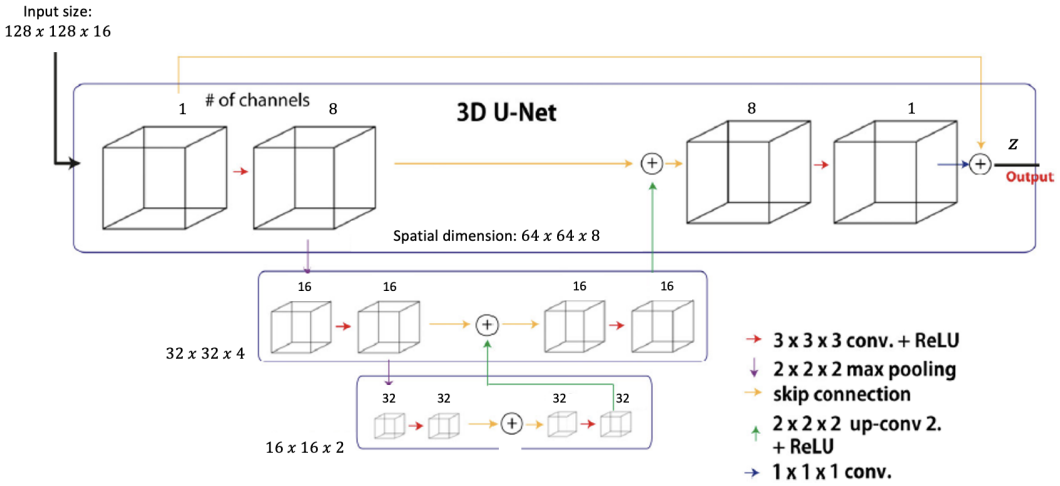


Figure 4.2: 3D U-Net architecture. conv, convolutional filter; ReLU, rectified linear unit; up-conv 2., transposed convolutional layer with a stride of two. This figure is adapted from [70].

4.4.1 Method for the Simplified Model

In the simplified model, we have $\mathbf{A} = \mathbf{GC}$. Inserting this expression into Eq. (4.10) yields:

$$\mathbf{x}^{l+1} = \left(\mathbf{CG}^H\mathbf{GC} + \nu\mathbf{I}\right)^{-1}\left(\mathbf{CG}^H\mathbf{y} + \nu\mathbf{z}^l\right). \tag{4.11}$$

Note that $\mathbf{C}^H = \mathbf{C}$ since $\mathbf{C}$ is a real diagonal matrix. Due to huge matrix sizes and inversion cost, we obtain this solution using the steepest descent (SD) method as:

$$\mathbf{x}^{l+1} = (1 - \epsilon\nu)\,\mathbf{x}^l - \epsilon\mathbf{CG}^H\mathbf{GCx}^l + \epsilon\left(\mathbf{CG}^H\mathbf{y} + \nu\mathbf{z}^l\right). \tag{4.12}$$

Here, ϵ is the step size of the steepest descent method. This iteration can be computed efficiently in the frequency domain using the diagonalizability of the convolution matrices in $\mathbf{G}$; we can exploit $\mathbf{G} = \bar{\mathbf{F}}^H\mathbf{\Lambda}\tilde{\mathbf{F}}$ [9] where $\bar{\mathbf{F}} = \mathbf{I}_R \otimes \mathbf{F}_{2D}$ and $\tilde{\mathbf{F}} = \mathbf{I}_P \otimes \mathbf{F}_{2D}$ with $\mathbf{F}_{2D}$ denoting the unitary 2D discrete Fourier transform (DFT) matrix. Multiplication by $\tilde{\mathbf{F}}$ (or $\tilde{\mathbf{F}}^H$) represents taking 2D FFT (or inverse FFT) of all spectral bands. Similarly, multiplication by $\bar{\mathbf{F}}^H$ represents taking 2D inverse FFT of all measurements. The matrix $\mathbf{\Lambda}$ is a block matrix of diagonal matrices whose diagonals consist of the 2D FFT of the PSFs. Multiplication with $\mathbf{C}$ corresponds to element-wise multiplication with the coded aperture. After fixing the iteration number K, we unroll the iterative algorithm in Eq. (4.9b) and (4.9a) into an end-to-end trainable network

and learn the parameters in this manner. Filter weights between different iterations are shared, while the parameters ϵ and ν are not. The overall algorithm is illustrated in Fig. 4.3.

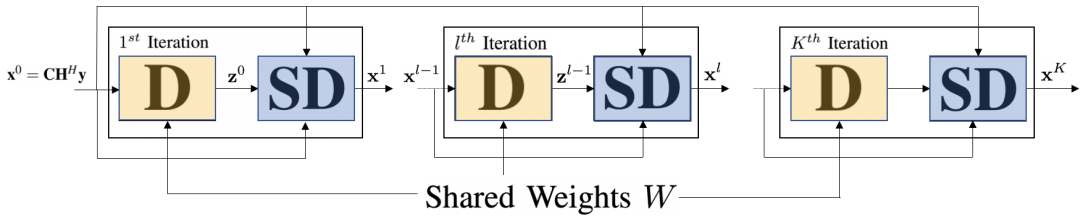


Figure 4.3: Architecture for the developed deep learning-based unrolled reconstruction method.

4.4.2 Method for the Truncated Model

In the truncated model, we have $\mathbf{A} = \tilde{\mathbf{T}}\mathbf{G}\mathbf{C}$. Inserting this expression into Eq. (4.10) yields:

$$\mathbf{x}^{l+1} = \left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G}\mathbf{C} + \nu\mathbf{I}\right)^{-1}\left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y} + \nu\mathbf{z}^l\right). \tag{4.13}$$

Direct computation of Eq. (4.13) requires forming, storing and inverting huge matrices, which have high computation costs. To have an efficient algorithm that exploits FFT, we do not form any matrices and instead solve this linear system of equations using the conjugate gradient method. Conjugate gradient method solves Eq. (4.13) iteratively by minimizing the difference between $\left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G}\mathbf{C} + \nu\mathbf{I}\right)\mathbf{x}^{l+1}$ and $\left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y} + \nu\mathbf{z}^l\right)$. Note that the conjugate gradient method requires multiplications with $\tilde{\mathbf{T}}\mathbf{G}\mathbf{C}$ and $\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H$ as given in Algorithm 1. We compute these multiplications without forming or storing any of the matrices. Consider the multiplication of a vector, say $\mathbf{x}$, with the matrix $\tilde{\mathbf{T}}\mathbf{G}\mathbf{C}$, i.e., $\tilde{\mathbf{T}}\mathbf{G}\mathbf{C}\mathbf{x}$. We calculate this multiplication efficiently as:

$$\mathcal{S}(x) = \sum_{p=1}^{P} \mathcal{T}\mathcal{F}^{-1}\left(\mathcal{F}\left(g_{r,p}[m,n]\right) \odot \mathcal{F}\left(c_p[m,n] \odot x_p[m,n]\right)\right). \tag{4.14}$$

In Eq. (4.14), $\mathcal{S}$ is the system operator for the r^{th} measurement which represents the forward model in Eq. (4.3). We implement this operator in the frequency domain by first element-wise multiplying each image slice x_p with the corresponding coded

aperture c_p, taking the 2D FFTs of each multiplication and PSF $g_{r,p}$, element-wise multiplying them, taking the 2D inverse FFT, truncating the result, and then summing over all image slices. Element-wise multiplication, i.e., Hadamard product, is denoted by $\odot$. Operators $\mathcal{F}$ and $\mathcal{F}^{-1}$ take 2D FFT and its inverse, which are of $(M_1 + N_1 - 1) \times (M_2 + N_2 - 1)$-point so that we get the result of the linear convolution. The operator $\mathcal{T}$ is the truncation operation. It truncates the center $N_1 \times N_2$ pixels.

Algorithm 2 Conjugate Gradient Method for Truncated Model

Input: $\mathbf{y}$: superimposed, blurred and coded measurements, $\mathbf{z}^l$: denoised image, E: conjugate gradient iteration number, ν: penalty parameter

Output: $\mathbf{x}^{l+1}$: reconstructed images

$\mathbf{x}^{l+1} \leftarrow \mathbf{z}^l$
$\mathbf{b} \leftarrow \mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y} + \nu\mathbf{z}^l$ {compute using Eq. (4.15)}
$\mathbf{r} \leftarrow \mathbf{b} - \left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G}\mathbf{C} + \nu\mathbf{I}\right)\mathbf{x}^{l+1}$ {compute using Eq. (4.14) and (4.15)}
$\mathbf{p} \leftarrow \mathbf{r}$
$\epsilon_1 \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$
$i \leftarrow 1$
for $i = 1$ to E **do**
    $\mathbf{q} \leftarrow \left(\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\tilde{\mathbf{T}}\mathbf{G}\mathbf{C} + \nu\mathbf{I}\right)\mathbf{p}$ {compute using Eq. (4.14) and (4.15)}
    $\alpha \leftarrow \epsilon_1 / \langle \mathbf{p}, \mathbf{q} \rangle$
    $\mathbf{x}^{l+1} \leftarrow \mathbf{x}^{l+1} + \alpha\mathbf{p}$
    $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{q}$
    $\epsilon_2 \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$
    $\beta \leftarrow \epsilon_2 / \epsilon_1$
    $\mathbf{p} \leftarrow \mathbf{r} + \beta\mathbf{p}$
    $\epsilon_1 \leftarrow \epsilon_2$
    $i \leftarrow i + 1$
end for
return $\mathbf{x}^{l+1}$

Now consider the multiplication of a vector, say $\mathbf{y}$, with the matrix $\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H$, i.e.,

$\mathbf{C}\mathbf{G}^H\tilde{\mathbf{T}}^H\mathbf{y}$. We calculate this multiplication efficiently as:

$$\mathcal{S}^*(y) = c_p[m,n] \odot \left(\sum_{r=1}^{R} \mathcal{T}\mathcal{F}^{-1}\left(\mathcal{F}\left(g_{r,p}^{\dagger}[m,n] \right) \odot \mathcal{F}\left(y_r[m,n] \right) \right) \right). \tag{4.15}$$

Here $\mathcal{S}^*$ is the adjoint of the system operator for the p^{th} spectral band with $g_{r,p}^{\dagger}[m,n] = g_{r,p}[M_1 - 1 - m, M_2 - 1 - n]$ being flipped version of the PSF $g_{r,p}$. We compute this operator similar to system operator in the frequency domain by first taking 2D FFTs, element-wise multiplying them, taking the 2D inverse FFT, truncating the inverse FFT, summing over all measurements, and then multiplying with the corresponding coded aperture. Hence, all matrix-vector multiplications can be computed using operators $\mathcal{S}$ and $\mathcal{S}^*$. Also, additions are element-wise, and inner products are computed using element-wise multiplications followed by element-wise additions. Hence, all steps of the conjugate gradient method are computed fast and efficiently without forming or storing any of the matrices. We give the updates in matrix-vector forms just for convenience. The conjugate gradient method runs for E iterations. After fixing the iteration number K, we unroll the algorithm into an end-to-end trainable network and learn the parameters in this manner. Filter weights and the penalty parameter ν between different iterations are shared during learning.

4.5 Numerical Results

In this section, we illustrate the reconstruction performance of the deep learning-based unrolled reconstruction methods both for the simplified and truncated models. We also compare their performances with the reconstruction method using analytical prior.

4.5.1 Results for the Simplified Model

For both training and testing, we use spectral images of size $128 \times 128 \times 16$ (wavelengths 480-630 nm with 10 nm interval) cropped from ICVL dataset [71]. For the diffractive lens, the photon sieve designs described in [9] are used. Using the forward model in Eq. (4.1), we simulate 3 measurements by focusing one of the equidistant wavelengths $\{500, 550, 600\}$ nm in each case. We choose unrolling number as 10.

The number of trainable parameters is $130,845$. Then, we train our network with 500 spectral image cubes by minimizing the l_2 loss between the ground truth and reconstructed data cubes using Adam solver [57]. Codes are implemented with Tensorflow on Nvidia GeForce RTX 2080 Ti GPU. Training takes approximately 13 hours.

The reconstruction performance averaged over 60 test data is given in Table 4.1 for measurement SNRs of 20 and 25 dB. For comparison, we also provide the results of the reconstruction method in [9], which uses an analytical prior with a Kronecker basis $\Phi = \Phi_1 \otimes \Phi_2$ with Φ_1 and Φ_2 denoting the 2D Symmlet-8 wavelet and 1D discrete cosine bases. We choose this analytical prior since it performs best among various other analytical priors including 3D TV. Spectral angle mapper (SAM) measures the spectral similarity between the original and reconstructed data cube. Lower SAM values indicate that reconstructed images can recover spectral information better. As seen from Table 4.1, the developed method with deep data-driven prior provides better spatial and spectral reconstruction quality than the earlier method with the handcrafted analytical prior. We have more than 2 dB improvement in PSNR, 0.05 improvement in SSIM, and around $0.7°$ improvement in SAM. Note that SAM values are significantly smaller than those reported in [9]. The reason is possibly the smaller number of spectral bands. The used spectral data cubes in [9] have 31 spectral bands. Since we are using only 16 spectral bands, spectral variations in the original data cubes are already somewhat limited. As future study, we plan to increase the number of spectral bands and explore this.

Table 4.1: Comparison of average PSNR (dB), SSIM and SAM values for different measurement SNRs (with best values shown in bold).

Input SNR	Developed Deep Prior	Analytical Prior
20dB	**32.72/0.90/2.36°**	30.54/0.85/3.06°
25dB	**33.88/0.93/1.95°**	31.15/0.88/2.66°

We also give average runtime in Table 4.2. Reconstruction with analytical prior runs maximum of 2000 iterations and terminates if the ratio of the Frobenius norm of the change in the image to the image's Frobenius norm becomes less than 0.001. As seen, we can reconstruct images only in 0.12 second when we use the developed deep prior. Analytical prior has varying reconstruction times depending on the structure of

the images, and the average reconstruction time is around 540 seconds. The developed deep prior is around 4500-fold faster than the analytical prior. Hence developed method not only outperforms analytical prior in terms of reconstruction quality but is also significantly faster than the analytical prior. This is desirable for real-time applications.

Table 4.2: Comparison of average runtime (second) for different measurement SNRs.

Input SNR	Developed Deep Prior	Analytical Prior
25dB	0.12	540

For visual evaluation, reconstructions for sample spectral test data cubes with different characteristics are shown in Fig. 4.4, 4.5, and 4.6. The reconstruction PSNR, SSIM, and SAM values are listed in Table 4.3. The developed method performs significantly better than the analytical prior both visually and in all evaluation metrics. Both methods perform worse for test image 1 than test images 2 and 3. This is expected since test image 1 is challenging to reconstruct due to high spectral and spatial variations. Still, even for this challenging test image, the developed method can successfully reconstruct the detailed structures. Reconstructions obtained with the analytical prior contain a significant amount of blur and noise.

Table 4.3: Comparison of PSNR (dB), SSIM and SAM values for different test images (with best values shown in bold).

Method	Developed Deep Prior	Analytical Prior
Test Image 1	**29.27/0.90/5.78**°	27.99/0.82/7.01°
Test Image 2	30.37/**0.94/1.41**°	**31.13**/0.86/1.93°
Test Image 2	**31.96/0.93/2.51**°	30.35/0.88/3.43°

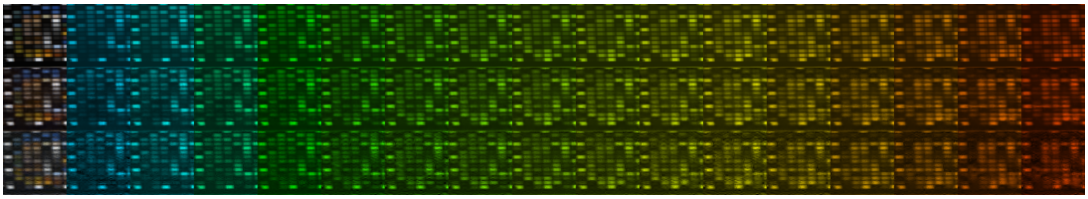


Figure 4.4: Reconstructed spectral images from compressive measurements for test image 1 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

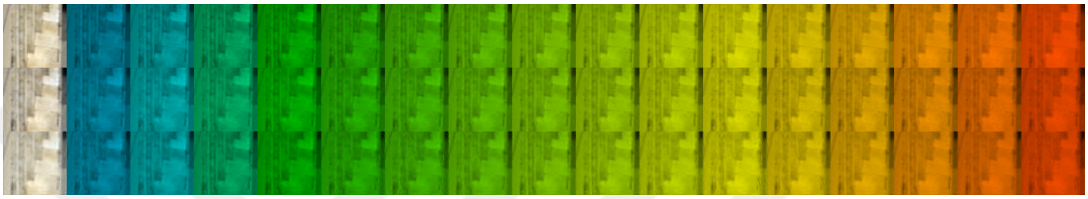


Figure 4.5: Reconstructed spectral images from compressive measurements for test image 2 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

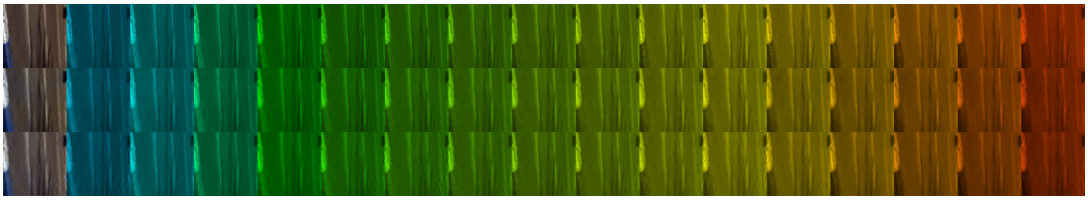


Figure 4.6: Reconstructed spectral images from compressive measurements for test image 3 using the simplified model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

Fig. 4.7 shows the histograms of reconstruction PSNR, SSIM, and SAM values at measurement SNR 25 dB for the reconstructions with the learned deep prior and analytical prior. We can see from Fig. 4.7 that deep prior learned reconstruction has better generalization capability than the analytical prior. The lowest PSNR is around 29 dB for the deep prior case, whereas it is around 26 dB for the analytical prior case.

Similarly, the lowest SSIM is around 0.85 for the deep prior case and around 0.75 for the analytical prior case. Similar performance observations are also valid for SAM values. Deep prior achieves a mean PSNR of 33.88 dB, SSIM of 0.93, and SAM of $1.95°$.

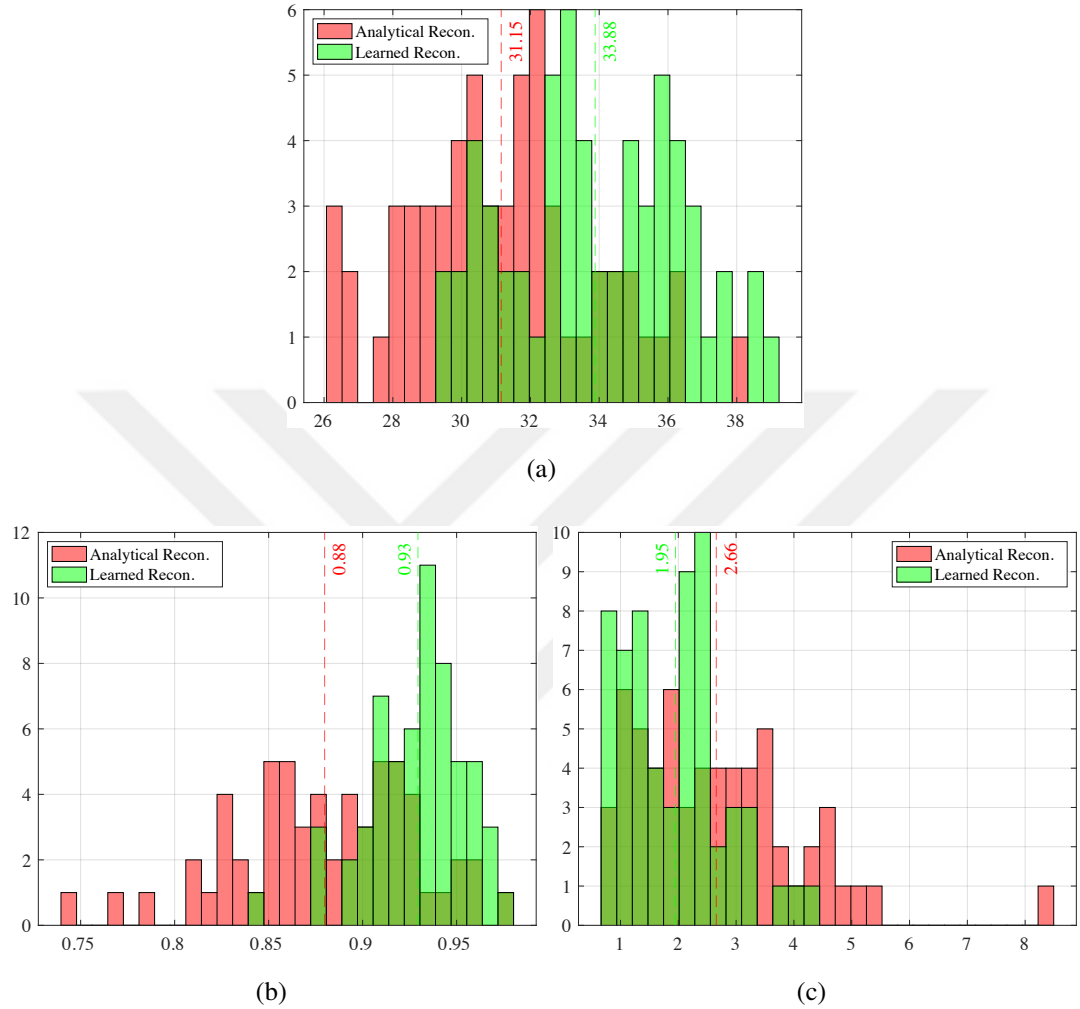


Figure 4.7: PSNR (a) and SSIM (b), and SAM (c) histograms of analytical and learned reconstructions using the simplified model. Red and green vertical lines show the mean values for analytical and learned reconstructions, respectively.

4.5.2 Results for the Truncated Model

For both training and testing, we use the spectral images described in Section 4.5.1 as the ground truth data. However, we use the truncated image formation model given in Eq. (4.3) to simulate the measurements. The number of training samples, number of measurements, photon sieve designs, unrolling number, and focused wavelengths are the same as in the previous section. The conjugate gradient method runs for 5 iterations. The number of trainable parameters is $130,826$. This is slightly smaller than the network for the simplified model since the penalty parameter is also shared, and there is no step-size parameter. We train our network by minimizing the l_1 loss between the ground truth and reconstructed data cubes using Adam solver [57]. Codes are implemented with Tensorflow on Nvidia GeForce RTX 2080 Ti GPU. Training takes approximately 32 hours.

The reconstruction performance averaged over 60 test data is given in Table 4.4 for a measurement SNR of 25 dB. For comparison, we also provide the results of the reconstruction method in [9]. As seen from Table 4.4, the developed method with deep data-driven prior provides better spatial and spectral reconstruction quality than the earlier method with the handcrafted analytical prior. We have more than 5 dB improvement in PSNR, 0.07 improvement in SSIM, and around $1.5°$ improvement in SAM.

Table 4.4: Comparison of average PSNR (dB), SSIM and SAM values (with best values shown in bold).

Input SNR	Developed Deep Prior	Analytical Prior
25dB	**34.96/0.89/2.04°**	29.64/0.82/3.63°

We also give average runtime in Table 4.5. Reconstruction with analytical prior runs maximum of 2000 iterations and terminates if the ratio of the Frobenius norm of the change in the image to the image's Frobenius norm becomes less than 0.001. As seen, we can reconstruct images only in 0.56 second when we use the developed deep prior. Reconstructions with the analytical prior are obtained in 1600 seconds. The developed deep prior is around 2857-fold faster than the reconstruction with analytical

prior. Hence again developed method outperforms analytical prior in terms of both reconstruction quality and speed. Compared to the simplified model, both methods require a longer time for reconstructions, which is expected since the simplified model has a diagonalizable forward model and does not need the conjugate gradient method.

Table 4.5: Comparison of average runtime (second) for different measurement SNRs.

Input SNR	Developed Deep Prior	Analytical Prior
25dB	0.56	1600

For visual evaluation, reconstructions for sample spectral test data cubes are shown in Fig. 4.8, 4.9, and 4.10, respectively. The reconstruction PSNR, SSIM, and SAM values are listed in Table 4.6. As seen, the developed method can successfully reconstruct the detailed structures, such as the lines, in the spectral images. Nevertheless, there is some noise and blurring left in some spectral bands due to the compressive setting. Still, the developed method performs significantly better than the analytical prior both visually and in all evaluation metrics. Reconstructions obtained with the analytical prior contain a significant amount of blur and noise.

Fig. 4.11 shows the histogram of reconstruction PSNR, SSIM, and SAM values for the reconstructions with the learned deep prior and analytical prior. We can see from Fig. 4.11 that deep prior learned reconstruction has better generalization capability than the analytical prior. The lowest PSNR is around 30 dB for the deep prior case, whereas it is around 26 dB for the analytical prior case. Similarly, the lowest SSIM is around 0.8 for the deep prior case and less than 0.7 for the analytical prior case. Similar performance observations are also valid for SAM values. Deep prior achieves a mean PSNR of 34.96 dB, SSIM of 0.89, and SAM of $2.04°$.

Table 4.6: Comparison of PSNR (dB), SSIM and SAM values for different test images using truncated model (with best values shown in bold).

Method	Developed Deep Prior	Analytical Prior
Test Image 1	**29.51/0.90/5.66°**	27.56/0.81/7.36°
Test Image 2	**34.79/0.88/1.59°**	27.97/0.76/3.07°
Test Image 3	**35.35/0.93/2.26°**	28.30/0.77/4.85°

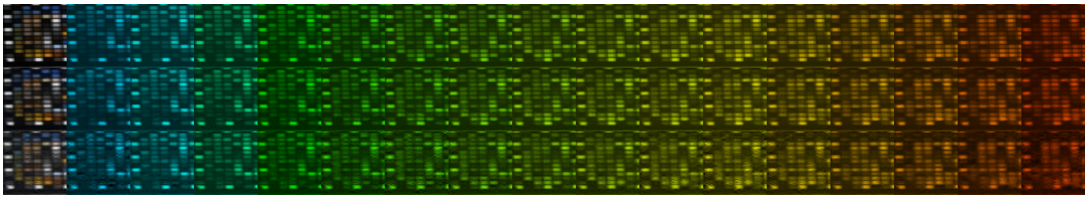


Figure 4.8: Reconstructed spectral images from compressive measurements for test image 1 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

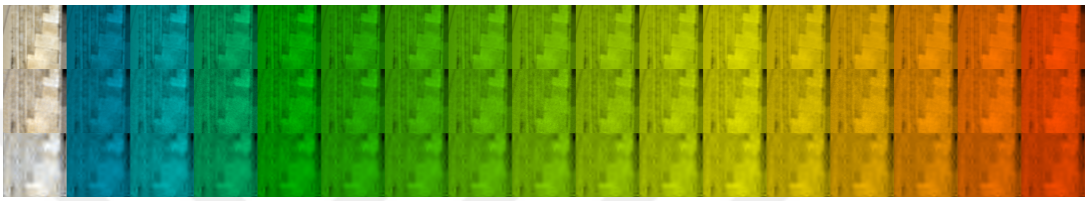


Figure 4.9: Reconstructed spectral images from compressive measurements for test image 2 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

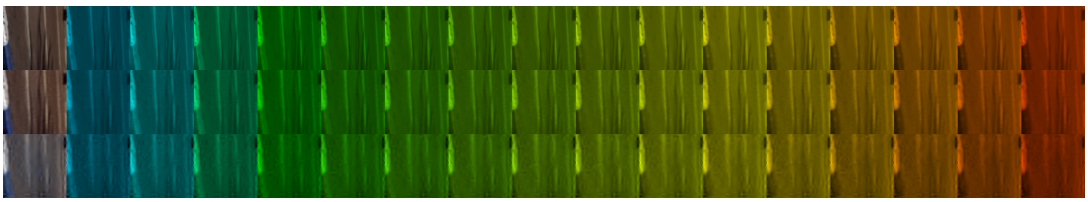


Figure 4.10: Reconstructed spectral images from compressive measurements for test image 3 using the truncated model - top to bottom: true images, reconstructions with the developed deep prior, and reconstructions with the analytical prior (on the leftmost column, superimposed image along the spectral direction is shown).

Compared to the simplified model, the performance of the analytical reconstruction gets worse. That is, the average PSNR and SSIM values are respectively decreased around 1.5 dB and 0.06. SAM value is increased around $1°$. This might stem from the sub-optimum parameter selections. Regarding the performance of the learned reconstruction, we have an increase in the PSNR and a decrease in the SSIM values. SAM

value stays almost the same. Therefore, learned reconstruction is robust to different problem formulations. The increase and the decrease in the average reconstruction values may arise from the different optimization methods used for the image update steps. Different parameter sharing strategies might also be affecting the results.

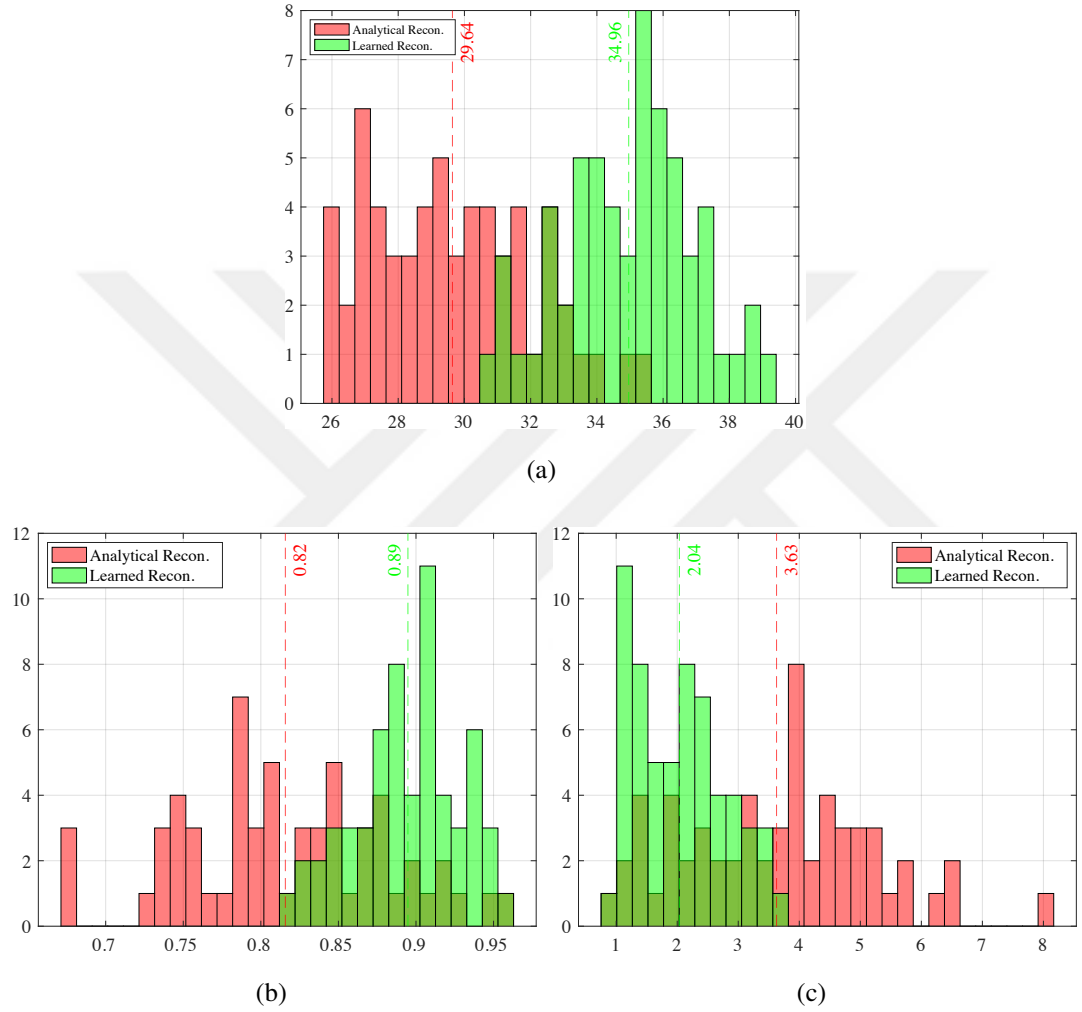


Figure 4.11: PSNR (a) and SSIM (b), and SAM (c) histograms of analytical and learned reconstructions using the truncated model. Red and green vertical lines show the mean values for analytical and learned reconstructions, respectively.

4.5.3 Model Mismatch Analysis

Here, we investigate the performance reductions due to the model mismatch. The model mismatch stems from the forward model discrepancies between measurements used for training and testing phases. This analysis provides us insights into how the performance will decrease if we use the simplified model instead of the truncated model. To this end, we use the network trained using the simplified model, i.e., the network in Section 4.5.1. However, we analyze its performance with the test images simulated using the truncated image formation model in Eq. 4.3. Remember that for the simplified model, we assume that the support of the spectral images is less than the support of the detectors. This approach effectively assumes zero end conditions for the spectral images and generally is not compatible with real-world imaging systems. We use the same coded mask both for training and testing.

We compute the average PSNR, SSIM, and SAM values over 60 test images for the analysis. As seen from Table 4.7, we have 13.46 dB decrease in the average PSNR and 0.55 decrease in the average SSIM values when there is a model mismatch. The average SAM value increases $5.23°$. Therefore, the reconstruction performance is significantly decreased.

Table 4.7: Average PSNR (dB), SSIM, and SAM values for learned reconstruction with and without mismatch.

Input SNR	With Mismatch	Without Mismatch
25dB	21.50/0.34/7.27°	34.96/0.89/2.04°

In Chapter 3, average PSNR and SSIM values decrease by 4.5 dB and 0.05 when there is a similar model mismatch. Here, the reduction in reconstruction performance is much more significant than that of in Chapter 3. There are several reasons for such an observation. In Chapter 3, we work on EUV solar images. These images have limited support where some pixels near edges are zero (see Fig. 3.1 and 3.5). However, in this chapter, we work on natural spectral images that do not have zero pixels near edges (see in Fig. 4.8 and 4.9). Therefore, the assumption of having smaller support for the spectral images than the detector range (i.e., zero end conditions) perform better for

the EUV solar imaging.

4.6 Conclusion

In this chapter, we developed deep learning-based unrolled reconstruction methods for compressive spectral imaging with diffractive lenses. We considered both the simplified and truncated models and highlighted the differences in formulation and solutions for these models. We used different approaches for the solution of the image update step. Still, both approaches are computationally efficient, i.e., they can be implemented without forming, storing, or inverting huge matrices.

Deep learning-based unrolled reconstruction methods achieve state-of-the-art reconstruction performance for both simplified and truncated models. They significantly outperform the earlier methods using analytical prior over a large test dataset. We obtain successful reconstructions with high mean PSNR, SSIM, and low SAM values. The developed method is robust to different measurement noise levels, as illustrated for the simplified model. Moreover, testing speed is much faster for the unrolled reconstruction since it basically requires 10 iterations (number of unrolling). In contrast, the reconstruction method with analytical prior runs in the order of hundred iterations.

When we visually examine the reconstructions obtained using the truncated model, we observe that the noise and blurring are not completely eliminated in several spectral bands. To increase the performance for the truncated model, one can train the network with a higher number of training samples, unroll the algorithm more than 10 times, or use non-shared weights between different layers of the network.

Lastly, the performance of the method can be further improved by system optimization. The entries of the coded mask are drawn from a Bernoulli distribution. We also choose focused wavelength empirically. These limit the quality of the measurements, and therefore the reconstruction performance. Jointly reconstructing the images and learning the optimum system design, i.e., optimum coded mask, optimum diffractive lens design, wavelengths to focus, etc., is possible. Several works have shown promising performance for such approaches [72–75]. The development of such tech-

niques is a current focus of our research group, and we will further focus on these topics in our future studies. Using more inner iterations, such as 10 iterations instead of 5 iterations in the conjugate gradient method, may also improve the performance. However, it will also increase the training time due to more computations involved.

CHAPTER 5

CONCLUSION

In this thesis, we developed novel deep learning-based unrolled reconstruction methods for computational imaging. We focused on diffractive multi-spectral imaging in Chapter 2. In Chapter 3, we addressed the limitations of the conventional image formation model used in Chapter 2. We formulated a realistic truncated image formation model and developed a deep learning-based unrolled reconstruction method to solve the associated inverse problems. In Chapter 4, we extended deep learning-based reconstruction methods to a compressive imaging modality. In each chapter, we first formulated the forward model and the inverse problem of the imaging system of interest. Then, we developed deep learning-based unrolled reconstruction methods using the HQS method. The performance of all the developed methods is comparatively evaluated for sample applications using different CNN architectures, including U-Net.

In Chapter 2, we developed different deep learning-based reconstruction methods for multi-spectral imaging with diffractive lenses. The key idea in these approaches is to replace a fixed analytical prior with a data-adaptive deep prior. The presented results demonstrated that the unrolled methods can achieve state-of-the-art reconstructions with diffraction-limited spatial and spectral resolutions. Unrolled methods also perform well for different levels of SNRs and are robust to model mismatches arising from the discretization along spectral dimension. The results also demonstrated that unrolled methods consistently outperform other learned and TV reconstruction methods. Unrolled approaches use the forward model during training. Hence, CNN learns not only denoising but also artifact cleaning. Therefore, they perform better than plug-and-play methods. In plug-and-play approaches, one should optimize

hyper-parameter ν and may need to train multiple networks as reconstruction SNR increases after each iteration. In our results, Unrolled-I outperformed Unrolled-II, which is due to the U-Net's ability to process the image at different spatial sizes and its multilevel decomposition. Hence, in the proceeded chapters, we concentrated on deep learning-based unrolled reconstruction methods with the U-Net.

Results also demonstrated that when the weights are not shared, there are improvements in the average PSNR and SSIM values compared to the weight sharing. As the input propagates through the network, the reconstruction artifacts decrease. Using different weights will allow each denoiser to learn to remove different levels of artifacts. Hence reconstruction performance improves. When there is limited memory and/or training samples, one can use the weight sharing strategy since it still provides a good reconstruction quality. However, when there are no such limitations, one can train the network without weight sharing since it improves performance.

In Chapter 3, we addressed the limitations and challenges of the image formation model used in Chapter 2 and formulated a truncated image formation model that can represent various two and multidimensional imaging modalities. The formulated image formation model is compatible with real-world imaging systems. Then, we developed a deep learning-based unrolled reconstruction method to solve the inverse problems associated with the truncated image formation model. To illustrate the utility and versatility, we applied the unrolled reconstruction to both 2D and 3D imaging problems. The results demonstrated that unrolled method consistently outperforms TV reconstruction method and has better generalization capability. Moreover, through the mismatch analysis for EUV solar imaging, we showed the performance gains obtained using the truncated image formation model.

When we use the truncated image formation model, the sub-problem containing the data fidelity term does not have a fast-to-compute closed-form solution since it requires forming, storing, and inverting huge matrices. As a result, we solved it using an efficient conjugate gradient implementation involving FFTs. We also developed an alternative solution to decrease the training time by avoiding the use of the conjugate gradient method. To this end, the alternative solution applies multiple variable splittings, such that each sub-problem can be solved using a closed-form solution. Our

training results show that the network can perform more iterations per second. Therefore, it has the potential for faster training. However, training loss may not converge. Resolving the converge issue will be a focus of our future research.

In Chapter 4, we extended the deep learning-based unrolled reconstruction methods to compressive spectral imaging with diffractive lenses. We considered both the simplified and truncated models and highlighted the differences in formulation and solutions for these models. We used different approaches for the solution of the image update step. Still, both approaches are computationally efficient, i.e., they can be implemented without forming, storing, or inverting huge matrices. For the denoising problem, we used a 3D U-Net instead of a 2D U-Net to exploit both spatial and spectral correlations. Results illustrated that deep learning-based unrolled reconstruction methods achieve state-of-the-art reconstruction performance for both simplified and truncated models. They significantly outperform the earlier methods using analytical prior over a large test dataset. We obtain successful reconstructions with high mean PSNR, SSIM, and low SAM values. The developed method is again robust to different measurement noise levels. The performance of the reconstruction method may be further improved by system optimization.

In each chapter, unrolled reconstruction methods performed better than purely analytical reconstruction methods for different levels of SNRs and image formation models. This shows the applicability and utility of the developed methods. Hence the methods developed in this thesis are applicable to similar imaging problems that involve reconstruction of images from their superimposed and blurred versions. Although training is long for unrolled methods, testing is much faster than plug-and-play and purely analytical reconstruction methods, which is desirable for practical scenarios. Unrolled methods run for the number of unrolling (e.g., 10, 20) iterations. In contrast, the plug-and-play and purely analytical reconstruction methods run in the order of hundred iterations. Hence, once the unrolled network is trained, it can be used for real-time applications.

Finally, the performance of the developed unrolled reconstruction methods can be further improved by using different networks. In Chapter 2, we have shown significant performance improvements only by changing the denoiser network. Therefore

other networks, such as DenseNets, Feature Pyramid Networks, etc., may perform better than the U-Net. Using another denoiser network or designing a new denoiser network to fit our imaging applications better will be a topic for future study. Using hybrid or different loss functions during training may also improve the performance.

REFERENCES

[1] L. Gao and L. V. Wang, "A review of snapshot multidimensional optical imaging: Measuring photon tags in parallel," *Physics Reports*, vol. 616, pp. 1–37, 2016.

[2] F. S. Oktem, L. Gao, and F. Kamalabadi, "Computational spectral and ultrafast imaging via convex optimization," in *Handbook of Convex Optimization Methods in Imaging Science*, pp. 105–127, Springer, 2017.

[3] R. M. Willett, M. F. Duarte, M. A. Davenport, and R. G. Baraniuk, "Sparsity and structure in hyperspectral imaging : Sensing, reconstruction, and target detection," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 116–126, 2014.

[4] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for dynamic mr image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2018.

[5] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3D imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[6] G. R. Arce, D. Brady, L. Carin, H. Arguello, and D. Kittle, "Compressive coded aperture spectral imaging: An introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 105–115, 2014.

[7] F. S. Oktem, F. Kamalabadi, and J. M. Davila, "High-resolution computational spectral imaging with photon sieves," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 5122–5126, IEEE, 2014.

[8] J. M. Davila, "High-resolution solar imaging with a photon sieve," in *Solar Physics and Space Weather Instrumentation IV*, pp. 263 – 273, International Society for Optics and Photonics, 2011.

[9] O. F. Kar and F. S. Oktem, "Compressive spectral imaging with diffractive lenses," *Optics Letters*, vol. 44, no. 18, pp. 4582–4585, 2019.

[10] X. Cao, T. Yue, X. Lin, S. Lin, X. Yuan, Q. Dai, L. Carin, and D. J. Brady, "Computational snapshot multispectral cameras: Toward dynamic capture of the spectral world," *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 95–108, 2016.

[11] A. Wagadarikar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Applied Optics*, vol. 47, no. 10, pp. B44–B51, 2008.

[12] Y. Wu, I. O. Mirza, G. R. Arce, and D. W. Prather, "Development of a digital-micromirror-device-based multishot snapshot spectral imaging system," *Optics Letters*, vol. 36, no. 14, pp. 2692–2694, 2011.

[13] E. Salazar, A. Parada-Mayorga, and G. R. Arce, "Spectral zooming and resolution limits of spatial spectral compressive spectral imagers," *IEEE Transactions on Computational Imaging*, vol. 5, no. 2, pp. 165–179, 2019.

[14] L. Denis, E. Thiébaut, F. Soulez, J.-M. Becker, and R. Mourya, "Fast approximations of shift-variant blur," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 253–278, 2015.

[15] F. Sroubek, J. Kamenicky, and Y. M. Lu, "Decomposition of space-variant blur in image deconvolution," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 346–350, 2016.

[16] A. La Camera, L. Schreiber, E. Diolaiti, P. Boccacci, M. Bertero, M. Bellazzini, and P. Ciliegi, "A method for space-variant deblurring with application to adaptive optics imaging in astronomy," *Astronomy & Astrophysics*, vol. 579, p. A1, 2015.

[17] X. Zhang and E. Y. Lam, "Edge-preserving sectional image reconstruction in optical scanning holography," *JOSA A*, vol. 27, no. 7, pp. 1630–1637, 2010.

[18] D. S. Badali and R. Dwayne Miller, "Robust reconstruction of time-resolved diffraction from ultrafast streak cameras," *Structural Dynamics*, vol. 4, no. 5, p. 054302, 2017.

[19] U. Rau and T. J. Cornwell, "A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry," *Astronomy & Astrophysics*, vol. 532, p. A71, 2011.

[20] I. Stewart, D. Fenech, and T. Muxlow, "A multiple-beam clean for imaging intraday variable radio sources," *Astronomy & Astrophysics*, vol. 535, p. A81, 2011.

[21] M.-O. Kim, S.-Y. Zho, and D.-H. Kim, "3d imaging using magnetic resonance tomosynthesis (mrt) technique," *Medical physics*, vol. 39, no. 8, pp. 4733–4741, 2012.

[22] D. Gol Gungor and L. C. Potter, "A subspace-based coil combination method for phased-array magnetic resonance imaging," *Magnetic resonance in medicine*, vol. 75, no. 2, pp. 762–774, 2016.

[23] M. I. Florea, A. Basarab, D. Kouamé, and S. A. Vorobyov, "An axially variant kernel imaging model applied to ultrasound image reconstruction," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 961–965, 2018.

[24] A. Besson, L. Roquette, D. Perdios, M. Simeoni, M. Arditi, P. Hurley, Y. Wiaux, and J.-P. Thiran, "A physical model of nonstationary blur in ultrasound imaging," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 381–394, 2019.

[25] D. Dogan and F. S. Oktem, "Convolutional inverse problems in imaging with convolutional sparse models," in *Mathematics in Imaging*, pp. JW2A–9, Optical Society of America, 2019.

[26] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: beyond analytical methods," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20–36, 2018.

[27] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[28] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439–1451, 2016.

[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, 2017.

[30] E. Nehme, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Deep-storm: super-resolution single-molecule microscopy by deep learning," *Optica*, vol. 5, pp. 458–464, Apr 2018.

[31] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2808–2817, IEEE, 2017.

[32] C. Işıl and F. S. Oktem, "Model-based phase retrieval with deep denoiser prior," in *Computational Optical Sensing and Imaging*, pp. CF2C–5, Optical Society of America, 2020.

[33] X. Yuan, Y. Liu, J. Suo, and Q. Dai, "Plug-and-play algorithms for large-scale snapshot compressive imaging," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1444–1454, IEEE, 2020.

[34] D. S. Jeon, S. H. Baek, S. Yi, Q. Fu, X. Dun, W. Heidrich, and M. H. Kim, "Compact snapshot hyperspectral imaging with diffracted rotation," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 117:1–13, 2019.

[35] L. Wang, C. Sun, M. Zhang, Y. Fu, and H. Huang, "DNU: Deep non-local unrolling for computational spectral imaging," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1658–1668, IEEE, 2020.

[36] F. S. Oktem, O. F. Kar, C. D. Bezek, and F. Kamalabadi, "High-resolution multi-spectral imaging with diffractive lenses and learned reconstruction," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 489–504, 2021.

[37] C. D. Bezek and F. S. Oktem, "Unrolling-based deep reconstruction for compressive spectral imaging," in *Computational Optical Sensing and Imaging*, pp. CM2E–3, Optical Society of America, 2021.

[38] Y. Sogabe, S. Sugimoto, T. Kurozumi, and H. Kimata, "ADMM-inspired reconstruction network for compressive spectral imaging," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2865–2869, IEEE, 2020.

[39] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model-based deep learning architecture for inverse problems," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2019.

[40] J. M. Davila, F. S. Oktem, F. Kamalabadi, J. O'Neill, A.-M. Novo-Gradac, A. N. Daw, and D. M. Rabin, "Milli-Arcsecond (MAS) Imaging of the Solar Corona," in *AAS/Solar Physics Division Abstracts #47*, vol. 47, pp. 3–10, 2016.

[41] O. F. Kar and F. S. Oktem, "Fast computational spectral imaging with a programmable diffractive lens," in *Computational Optical Sensing and Imaging*, pp. CTh3A–3, Optical Society of America, 2019.

[42] P. Wang and R. Menon, "Computational multispectral video imaging," *JOSA A*, vol. 35, no. 1, pp. 189–199, 2018.

[43] M. Nimmer, G. Steidl, R. Riesenberg, and A. Wuttig, "Spectral imaging based on 2D diffraction patterns and a regularization model," *Optics Express*, vol. 26, no. 22, pp. 28335–28348, 2018.

[44] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, vol. 7. SIAM, 2010.

[45] C. R. Vogel, *Computational methods for inverse problems*, vol. 23. SIAM, 2002.

[46] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," in *Readings in Computer Vision*, pp. 564–584, Elsevier, 1987.

[47] U. Kamaci, F. C. Akyon, T. Alkanat, and F. S. Oktem, "Efficient sparsity-based inversion for photon-sieve spectral imagers with transform learning," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1225–1229, IEEE, 2017.

[48] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[51] C. Isil, F. S. Oktem, and A. Koc, "Deep iterative reconstruction for phase retrieval," *Applied Optics*, vol. 58, pp. 5422–5431, Jul 2019.

[52] P. Brekke, W. Thompson, T. Woods, and F. Eparvier, "The extreme-ultraviolet solar irradiance spectrum observed with the Coronal Diagnostic Spectrometer (CDS) on SOHO," *The Astrophysical Journal*, vol. 536, no. 2, p. 959, 2000.

[53] J. R. Lemen, A. M. Title, D. J. Akin, P. F. Boerner, C. Chou, J. F. Drake, D. W. Duncan, C. G. Edwards, F. M. Friedlaender, G. F. Heyman, *et al.*, "The atmospheric imaging assembly (aia) on the solar dynamics observatory (sdo)," *Solar Physics*, pp. 1–24, 2011.

[54] G. Andersen, "Large optical photon sieve," *Optics Letters*, vol. 30, no. 22, pp. 2976–2978, 2005.

[55] P. R. Young, D. G. Zanna, H. E. Mason, K. P. Dere, E. Landi, M. Landini, G. A. Doschek, C. M. Brown, L. Culhane, L. K. Harra, *et al.*, "EUV emission lines and diagnostics observed with Hinode/EIS," *Publications of the Astronomical Society of Japan*, vol. 59, no. sp3, pp. S857–S864, 2007.

[56] R. Menon, D. Gil, G. Barbastathis, and H. I. Smith, "Photon-sieve lithography," *J. Opt. Soc. Am. A*, vol. 22, no. 2, pp. 342–345, 2005.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, pp. 1–15, 2015.

[58] I. Manisali, R. M. Cam, C. D. Bezek, and F. S. Oktem, "Deep cnn prior based image reconstruction for multispectral imaging," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2020.

[59] D. Attwood, *Soft x-rays and extreme ultraviolet radiation: principles and applications*. Cambridge University Press, Cambridge, 2000.

[60] J. A. Fessler, *Image Reconstruction: Algorithms and Analysis*. 2021.

[61] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, 2011.

[62] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[63] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 416–423 vol.2, 2001.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[65] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.

[66] X. Miao, X. Yuan, Y. Pu, and V. Athitsos, "lambda-net: Reconstruct hyperspectral images from a snapshot measurement," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4058–4068, 2019.

[67] Z. Meng, J. Ma, and X. Yuan, "End-to-end low cost compressive spectral imaging with spatial-spectral self-attention," in *Computer Vision – ECCV 2020*, pp. 187–204, 2020.

[68] S. Zheng, Y. Liu, Z. Meng, M. Qiao, Z. Tong, X. Yang, S. Han, and X. Yuan, "Deep plug-and-play priors for spectral snapshot compressive imaging," *Photonics Research*, vol. 9, 2 2021.

[69] X. Yuan, D. J. Brady, and A. K. Katsaggelos, "Snapshot compressive imaging: Theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 38, 3 2021.

[70] K. H. Jin and M. Unser, "3d bbpconvnet to reconstruct parallel mri," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 361–364, 2018.

[71] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *European Conference on Computer Vision*, pp. 19–34, Springer, 2016.

[72] H. K. Aggarwal and M. Jacob, "J-modl: Joint model-based deep learning for optimized sampling and reconstruction," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1151–1162, 2020.

[73] L. Wang, T. Zhang, Y. Fu, and H. Huang, "Hyperreconnet: Joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2257–2270, 2019.

[74] M. Kellman, E. Bostan, M. Chen, and L. Waller, "Data-driven design for fourier ptychographic microscopy," in *2019 IEEE International Conference on Computational Photography (ICCP)*, pp. 1–8, 2019.

[75] S.-H. Baek, H. Ikoma, D. S. Jeon, Y. Li, W. Heidrich, G. Wetzstein, and M. H. Kim, "Single-shot hyperspectral-depth imaging with learned diffractive optics," *arXiv preprint arXiv:2009.00463*, 2020.