

SANDIA REPORT

SAND2020-9671

Printed



Sandia
National
Laboratories

Cognition at the Point of Sensing

Eric A. Shields, Dennis J. Lee, Andrew Klotz

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <https://classic.ntis.gov/help/order-methods>



ABSTRACT

Over the last 15 years, compressive sensing techniques have been developed which have the potential to greatly reduce the amount of data collected by systems while preserving the amount of information obtained. A cost of this efficiency is that a computationally-intensive optimization routine must be used to put the sensed data into a form that a person can interpret.

At the same time, machine learning techniques have experienced tremendous growth as well. Machines have demonstrated the ability learn how to effectively perform tasks such as detection and classification at speeds much faster than humanly possible.

Our goal in this project was to study the feasibility of using compressive sensing systems "at the edge." That is, how can compressive sensing sensors be deployed such that information is created at the remote sensor rather than sending raw data to a central processing location? Studies were performed to analyze whether machine learning could be done on the compressively sensed data in its raw form. If a machine is performing the task, is it possible to do so without putting the data into a human interpretable form? We show that this is possible for some systems, in particular a compressive sensing snapshot imaging spectrometer. Machine learning tasks were demonstrated to be more effective and more robust to noise when the machine learning algorithm worked on data in its raw form. This system is shown to outperform a traditional spectrometer.

Techniques for reducing the complexity of the reconstruction routine were also analyzed. Techniques for such as data regularization, deep neural networks, and matrix completion were studied and shown to have benefits over traditional reconstruction techniques.

In this project we showed that compressive sensing sensors are indeed feasible at the edge. As always, sensors and algorithms must be carefully tuned to work in the constrained environment. In this project we developed tools and techniques to enable those analyses.

ACKNOWLEDGMENT

The authors would like to acknowledge Sandia's Laboratory Directed Research and Development (LDRD) Office. This work was funded and performed as a three-year LDRD project.

CONTENTS

1. Introduction	11
2. Compressive Sensing Theory	12
2.1. The 12-Ball Problem	12
2.2. Exact Recovery of Sparse Signals	15
2.3. Reconstruction Techniques	18
3. Single-Frame Super-Resolution via a Coded Aperture and Compressive Sensing	19
4. Compressive Sensing Snapshot Imaging Spectrometer	25
4.1. CSSIS Trade Space	26
4.1.1. Improved Spectral Resolution with Constant Spatial Resolution	27
4.1.2. Improved Spatial Resolution with Constant Spectral Resolution	27
4.1.3. Improved Area Coverage with Constant Spectral and Spatial Resolutions	27
4.2. Spectral Super-Resolution	27
4.3. Spectral Classification	36
4.4. Indian Pines Data Set with Intraclass Variation	40
4.5. Conclusion	43
5. Compressive Hyperspectral Imaging Using Total Variation Minimization	44
5.1. Introduction	44
5.2. Theory	44
5.2.1. Notation	44
5.2.2. Measurement model	46
5.2.3. Inverse problem formulation	48
5.2.4. Alternating direction method of multipliers	49
5.3. Results	52
5.4. Simplifying the update of the hyperspectral image	62
5.5. Conclusion	62
6. Deep Neural Networks for Compressive Hyperspectral Imaging	64
6.1. Background: Compressive Hyperspectral Imaging	64
6.2. Experiments	66
6.2.1. Dataset	66
6.2.2. Task 1: Reconstruction of compressive hyperspectral images	66
6.2.3. Task 2: Classification of compressive hyperspectral images	68
6.3. Results	69
6.3.1. Task 1: Reconstruction of compressive hyperspectral images	69

6.3.2. Task 2: Classification of compressive hyperspectral images	72
6.4. Additional Reconstruction Metrics	79
6.5. Additional Classification Metrics	80
6.6. Conclusion	83
7. Matrix Completion for Compressive Sensing Using Consensus Equilibrium	84
7.1. Introduction	84
7.2. Theory	85
7.2.1. Notation	85
7.2.2. Compressive sensing	86
7.2.3. Matrix completion	86
7.2.4. Consensus equilibrium	88
7.3. Experiment	90
7.4. Conclusion	93
References	94
References	94

LIST OF FIGURES

Figure 3-1.	Coded Aperture Compressive Sensing Super-Resolution Architecture	20
Figure 3-2.	Coded Aperture Magnitude and Phase	21
Figure 3-3.	Coded Aperture Point-Spread Function	21
Figure 3-4.	Compressive sensing measurements and reconstructions	22
Figure 3-5.	Interpolation and Compressive Sensing Reconstruction Errors	23
Figure 4-1.	Traditional Bayer Filter Geometry	25
Figure 4-2.	CSSIS Geometry	26
Figure 4-3.	Fabry-Perot Filter Thicknesses	28
Figure 4-4.	Super-Resolution Model Measurement Matrix	29
Figure 4-5.	Spectra used for Spectral Super-Resolution Analysis	29
Figure 4-6.	Downsampled Spectra	30
Figure 4-7.	CSSIS Measurements of Spectra 1 and 2	30
Figure 4-8.	CSSIS Reconstructions of Spectra 1 and 2	31
Figure 4-9.	Point-Spread Function at $10.24 \mu\text{m}$	31
Figure 4-10.	Point-Spread Function at $10.24 \mu\text{m}$ at Pixel Resolution	32
Figure 4-11.	Point-Spread Function at $10.24 \mu\text{m}$ at Super-Pixel Resolution	32
Figure 4-12.	Spatial Distributions of Spectra 1 and 3	33
Figure 4-13.	CSSIS Raw Image	34
Figure 4-14.	Traditional Spectrometer Raw Image	34
Figure 4-15.	CSSIS Raw Image	35
Figure 4-16.	Traditional Spectrometer Raw Image	35
Figure 4-17.	CSSIS Measurement Matrix for the Indian Pines Reconstruction Study	37
Figure 4-18.	CSSIS Measurement Matrix for the Indian Pines Reconstruction Study	38
Figure 4-19.	Classification Accuracy on Indian Pines Data Set	39
Figure 4-20.	SNR of CSSIS and Traditional Spectrometer Signals	40
Figure 4-21.	Classification Accuracy vs. Noise of the Indian Pines Data Set	42
Figure 5-1.	Concept of a hyperspectral imager using a filter array and detector array	46
Figure 5-2.	Indian Pines scene with labels assigned to each pixel	53
Figure 5-3.	Plots of the wavelength-dependent transmission of a Fabry-Perot resonator	54
Figure 5-4.	Transmission of a Fabry-Perot resonator as a function of mirror spacing and wavelength	55
Figure 5-5.	Convergence of our proposed Algorithm 1. The reconstructed hyperspectral data cube x converges in less than 80 iterations.	55
Figure 5-6.	Comparison of ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”)	56
Figure 5-7.	Continued: Comparison of ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”).....	57

Figure 5-8.	Continued: Comparison of ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”).	58
Figure 5-9.	Continued: Comparison of ground truth, the initial reconstruction, and the CSTV reconstruction.	59
Figure 5-10.	Comparison of spectral images from ground truth, the proposed Algorithm 1 (“CSTV”), and the initial reconstruction	60
Figure 5-10.	Continued: Comparison of spectral images from ground truth, the reconstruction from the proposed Algorithm 1 (“CSTV”), and the initial reconstruction. . .	61
Figure 5-10.	Continued: Comparison of spectral images from ground truth, the reconstruction from the proposed Algorithm 1 (“CSTV”), and the initial reconstruction. . .	62
Figure 6-1.	The proposed compressive hyperspectral imager filters light before the detector array [49]. Each measurement corresponds to a different spectral transmission on the filter array.	65
Figure 6-2.	Examples of spectral transmissions of a Fabry-Perot resonator. The spectral transmission depends the mirror spacing in the resonator.	65
Figure 6-3.	Spectral transmission as a function of mirror spacing and wavelength. The number of measurements equals the number of mirror spacings.	66
Figure 6-4.	Labels of the Indian Pines hyperspectral image. Each label corresponds to a type of farm crop.	67
Figure 6-5.	Example reconstructions from varying numbers of measurements with a single layer perceptron.	69
Figure 6-6.	Reconstructions of each class/crop from 10 measurements using a single layer perceptron, after running 2000 epochs. The plots show the average spectra with confidence intervals of one standard deviation, which shows intraclass variation. GT: Ground Truth. NN: Single layer perceptron.	70
Figure 6-7.	R^2 for reconstructions over two datasets (Indian Pines and random spectra). . .	71
Figure 6-8.	R^2 for reconstructions over two datasets (Indian Pines and random spectra). . .	72
Figure 6-9.	Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the compressed (non-reconstructed) measurements. The number of bands in the hyperspectral image is 220, and the compression ratio is the number of measurements divided by the number of bands. Note that the 3D CNN from He does not accept inputs of size 10, so these values are empty. Each model runs 20 times, with 100 epochs per run.	73
Figure 6-10.	Class accuracies with varying numbers of measurements (compressed signal length) using the recurrent neural network from Mou. The inputs to the classifier are the compressed (non-reconstructed) measurements.	74
Figure 6-11.	Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the reconstructed hyperspectral images. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. Each model runs 20 times, with 100 epochs per run.	75

Figure 6-12. Class accuracies with varying numbers of measurements (compressed signal length) using the multiscale 3D convolutional neural network from He. The inputs to the classifiers are the reconstructed hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images, so the values at 220 are empty in the table since the original signal length is 220.	76
Figure 6-13. Accuracy comparison of four different classifiers. Each plot compares two inputs to the classifier: the compressed (non-reconstructed) measurements, and the reconstructed spectra. Note that the reconstructed spectra has full size (220 bands), and the x-axis denotes the number of compressed measurements before reconstruction. The size of 220 corresponds to the full spectral input to the classifier. Each model runs 20 times with 100 epochs per run.	78
Figure 6-14. Mean squared error (MSE) in the reconstruction with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.	79
Figure 6-15. Mean squared error (MSE) for each dataset with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.	80
Figure 6-16. Average accuracy of classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.	81
Figure 6-17. Average kappa from classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.	82
Figure 7-1. Reconstruction of compressive measurements with missing data. The plots show the spectra from one pixel of the Indian Pines hyperspectral data. μ denotes the fraction of missing data. In all simulations, the compression ratio $\kappa = 50\%$. Radiance is in units of $[\text{W cm}^{-2} \text{ nm}^{-1} \text{ sr}^{-1}]$	91
Figure 7-2. Error increases with the amount of missing data in the compressive space. In all simulations, the compression ratio $\kappa = 50\%$	92

LIST OF TABLES

Table 2-1. Example 12-Ball Problem Adaptive Solution	13
Table 2-2. 12-Ball Problem Non-Adaptive Measurement Scheme	13
Table 2-3. 12-Ball Problem Non-Adaptive Solution Table	14
Table 3-1. CNN for NxN images	24
Table 3-2. Image classification accuracies	24
Table 5-1. Description of variables	45
Table 5-2. Description of the class labels of the Indian Pines dataset	51

1. INTRODUCTION

The theory of compressive sensing (CS) provides a new paradigm for data collection systems. For decades, the Shannon-Nyquist sampling theorem was thought to provide a limit on the amount of information that can be obtained from a signal. Compressive sensing provides a framework that allows fewer measurements to be made to recover the same amount of information.

CS has been applied to a wide variety of sensing modalities. Early demonstrations focused on two-dimensional imaging. The application of compressive sensing to tomography could potentially provide a path to medical imaging with reduced risk to patients. Compressive sensing techniques can be applied to imaging spectrometers to revolutionize the trades between spatial resolution, spectral resolution, temporal resolution, and field-of-view.

Under the current CS paradigm, data are first reconstructed to traditional forms that a person can understand. For instance, in a CS imaging system the raw data may look nothing like an image to a person, so optimization techniques are employed to reconstruct the image from the raw measurements. This optimization step does not add any information to the data, though. Once the data are in the reconstructed form, information extraction then occurs. Traditionally this step is done by an human analyst. However, machine learning algorithms are now increasingly used for this step.

Researchers are now beginning to pair compressive sensing system with modern machine learning algorithms. In this case the compressive sensing system acts somewhat like a feature extraction technique. The features are known to be good because the CS basis set is assumed *a priori* to be sufficient to represent the data.

Machine learning algorithms operating on compressively sensed data are expected to perform comparably to algorithm operating on reconstructed data. However, since the amount of data on which the algorithms operate is significantly reduced, gains in speed or reductions in power can be achieved.

This project innovates by bringing together cutting-edge systems (those based on compressive sensing) with cutting-edge algorithms (machine learning). The traditional model of casting the data into a human-readable form is unnecessary. By operating on the data in its compressed form the size, weight, and power of the processing electronics are significantly reduced. Such gains are critical for the deployment of remote autonomous systems.

By employing CS techniques the sensors will gather information much more efficiently. By performing data processing on board, these systems will be able to make decisions faster and be able to operate in degraded communications environments since large amounts of data will not need to be downloaded to the ground for processing. This has the potential to revolutionize our nations future autonomous systems.

2. COMPRESSIVE SENSING THEORY

Compressive sensing is a growing field with the potential to dramatically impact sensing systems of nearly every modality. Traditional Shannon-Nyquist sampling theory dictates that to reconstruct a signal of bandwidth B , uniform samples of width $1/(2B)$ should be collected in order to perfectly reconstruct the signal. The Shannon-Nyquist theory is a sufficient but not necessary condition, however. It assumes that no information about the signal is known *a priori*.

2.1. The 12-Ball Problem

Before jumping into the math associated with compressive sensing, we consider an example to provide intuitive insight. The problem we consider is known as the “12-Ball Problem.” Consider a set of 12 balls. All of these balls are exactly the same size, but one defective ball is either lighter or heavier than the other 11. How many measurements do we need to make to find the defective ball and to determine whether it is lighter or heavier?

An obvious solution is to use a scale to measure the weight of each ball separately. Since the weights of the balls are not known, measuring one ball is insufficient. Even measuring two is insufficient since we do not know if the defective ball is lighter or heavier than the others. By measuring three balls, we will find the defective ball if it is one of the balls that is measured. Therefore, a minimum of three measurements are needed.

But what if the defective ball is the last one selected for measurement? In that case, 12 measurements are needed. After 11 measurements we know that the 12th ball is defective, but we do not know if it is lighter or heavier. Hence a 12th measurement must be made. This direct technique will require at least three and at most 12 measurements.

By changing the measurement scheme, we can find this ball in three measurements. Instead of weighing each ball separately, we use a balance scale. This scale can hold multiple balls on each side. To demonstrate one solution technique, we consider that Ball 9 is lighter than the others. Table 2-1 shows an example measurement scheme.

After the first measurement, we know that balls 1-8 are identical and the defective ball is in the range of 9-12. After the second measurement we know that one of the balls in the range 9-11 is lighter. After the third measurement we know that ball 9 is lighter.

An interesting result is obtained in the first measurement. Since the scales balance, we know that the defective ball must be one of balls 9-12. Thus we learn about balls 9-12 even though we did not make a measurement on them. This demonstrates the power of *a priori* information. Indeed, if the first measurement had used six balls on each scale, we will need four measurements to obtain the correct answer. We are in fact better off *not* measuring balls 9-12 in this first measurement.

Table 2-1. This table demonstrates the measurements that could be made to determine that Ball 9 is lighter than the others.

Measurement Number	Balls on Side A	Balls on Side B	Result
1	1,2,3,4	5,6,7,8	Balanced
2	1,2,3	9,10,11	Side A Heavier
3	9	10	Side B Heavier

Also notice that we are not measuring the balls individually. To obtain the solution to the problem in the minimum number of measurements we make measurements on groups of balls.

Lastly, notice that this solution is adaptive - the results of measurements 2 and 3 depend on the results of measurements 1 and 2. A non-adaptive solution exists as well. Consider the measurement scheme shown in Table 2-2.

Table 2-2. This table demonstrates a non-adaptive measurement scheme to solve the 12-Ball Problem.

Measurement Number	Balls on Side A	Balls on Side B
1	1,2,3,4	5,6,7,8
2	1,4,8,9	2,3,11,12
3	3,7,9,12	1,2,5,10

This non-adaptive measurement technique provides a solution to the 12-Ball Problem in three measurements. Table 2-3 provides the answers.

Table 2-3. This table demonstrates the results of the non-adaptive solution to the 12-Ball Problem. M1 denotes Measurement 1, M2 denotes Measurement 2, and M3 denotes Measurement 3.

M1 Heavier	M2 Heavier	M3 Heavier	Result
B	B	A	1 is lighter
A	A	B	1 is heavier
B	A	A	2 is lighter
A	B	B	2 is heavier
B	A	B	3 is lighter
A	B	A	3 is heavier
B	B	Balanced	4 is lighter
A	A	Balanced	4 is heavier
A	Balanced	A	5 is lighter
B	Balanced	B	5 is heavier
A	Balanced	Balanced	6 is lighter
B	Balanced	Balanced	6 is heavier
A	Balanced	B	7 is lighter
B	Balanced	A	7 is heavier
A	B	Balanced	8 is lighter
B	A	Balanced	8 is heavier
Balanced	B	B	9 is lighter
Balanced	A	A	9 is heavier
Balanced	Balanced	A	10 is lighter
Balanced	Balanced	B	10 is heavier
Balanced	A	Balanced	11 is lighter
Balanced	B	Balanced	11 is heavier
Balanced	A	B	12 is lighter
Balanced	B	A	12 is heavier

This system can be represented by the equation $y = \Phi x$. In this equation y is a three-element column vector of measurement results. y_i is +1 if Side A is heavier for the i th measurement, -1 if Side B is heavier, and 0 if the scales are balanced. x is a 12-element column vector of ball weights. x_j is +1 if the j th ball is heavy, -1 if it is light, and 0 if it is not defective. Φ is a 3x12 measurement array given by

$$\Phi = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 \\ -1 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 1 \end{pmatrix}. \quad (2.1)$$

$\Phi_{i,j}$ is +1 if the j th ball is used on Side A for the i th measurement, -1 if it is used on Side B, and 0 if it is not used at all. Equation 2.1 encapsulates the measurement scheme described in Table 2-2.

For the case where Ball 9 is lighter, we have the problem of finding x such that

$$\begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 \\ -1 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{pmatrix}, \quad (2.2)$$

which is underdetermined (we have three equations and 12 unknowns).

Since this equation is underdetermined, an infinite number of solutions exist. Compressive sensing indicates that we should look for the solution with the smallest ℓ_1 -norm while maintaining $y = \Phi x$. The ℓ_1 -norm is given by

$$\ell_1\text{-norm}(x) = \sum_{\text{all } i} |x_i|. \quad (2.3)$$

We find this solution using optimization techniques and for our example problem this results in

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (2.4)$$

This problem is solvable because we know only one element of x is non-zero, we have made measurements of groups of x rather than individual elements of x , and because we have very carefully constructed Φ .

2.2. Exact Recovery of Sparse Signals

In the compressive sensing (CS) framework, we assume that the signal is known to be sparse. Consider a signal $x_s \in \mathbb{R}^{N \times 1}$. This signal has k -sparsity if at most k elements of x_s are non-zero.

Next, consider a measurement system that involves making $M < N$ measurements of the inner product of x_s with measurement vectors $\{\phi_i\}_{i=1}^M$. We represent the results of these measurements via $y \in \mathbb{R}^{M \times 1}$. We then have $y_j = \langle \phi_j | x_s \rangle$. If we construct an array $\Phi \in \mathbb{R}^{M \times N}$ whose rows are the transpose of the vectors ϕ_i , we can write

$$y = \Phi x_s. \quad (2.5)$$

The problem in compressive sensing is to determine x_s from knowledge of Φ and measurements y using the knowledge that y is k -sparse in Φ . This results in the problem

$$\hat{x} = \operatorname{argmin} \|x_s\|_0 \text{ such that } y = \Phi x_s \quad (2.6)$$

where $\|x_s\|_0$ denotes the ℓ_0 -norm of x_s (that is, it counts the number of non-zero elements of x_s). The solution to Equation 2.6 is in general numerically unstable and NP-hard. It requires an exhaustive search of all $\binom{N}{k}$ possible locations of the non-zero elements of x_s .

However, it can be shown that the solution to this equation can be solved via a complex optimization problem of the form

$$\hat{x} = \operatorname{argmin} \|x_s\|_1 \text{ such that } y = \Phi x_s \quad (2.7)$$

where $\|x_s\|_1$ is the ℓ_1 -norm of x_s and certain restrictions are made on Φ [15]. This optimization problem is known as basis pursuit.

A necessary and sufficient condition for exact reconstruction is that for any k -sparse vector v which has the same non-zero locations as x_s and for some $\varepsilon > 0$,

$$1 + \varepsilon \geq \frac{\|\Phi v\|_2}{\|v\|_2} \geq 1 - \varepsilon, \quad (2.8)$$

where $\|v\|$ denotes the ℓ_2 -norm of v . That is, Φ must approximately preserve the lengths of k -sparse vectors whose non-zero locations match those of x_s . This condition is known as the Restricted Isometry Property (RIP)[14]. The utility of Equation 2.8 is limited since the locations of the k non-zero entries of x_s are not known.

A sufficient (but not necessary) condition for k -sparse signal reconstruction is that Φ satisfies Equation 2.8 for an arbitrary $3k$ -sparse vector v .

A closely related property is known as incoherence[26]. Let T denote matrix transposition and G denote the Gram matrix $G = \Phi^T \Phi$. Then, assuming the columns of Φ are normalized to unity ℓ_2 -norm, the coherence $\mu(\Phi)$ is given by

$$\mu(\Phi) = \max_{1 \leq k, j \leq N, k \neq j} |G(k, j)|, \quad (2.9)$$

where the entries of G are denoted via $G(k, j)$. Incoherence applies when $\mu(\Phi)$ is small relative to unity. When Φ is incoherent, CS reconstruction is possible via the solution to Equation 2.7. More specifically, if

$$\mu(\Phi) < \frac{1}{2k-1} \quad (2.10)$$

then the solution to Equation 2.7 is unique and can be found [20].

Approximate Recovery of Compressible Signals

Most signals of interest are not sparse. However, they are often *compressible* in some basis set. That is, we can represent a non-sparse signal $x \in \mathbb{R}^{N \times 1}$ via basis functions ψ_i with

$$x = \sum_{i=1}^N \alpha_i \cdot \psi_i \quad (2.11)$$

where α_i is the coefficient of the i -th basis function. We can represent this more efficiently by constructing an array $\Psi \in \mathbb{R}^{N \times N}$ whose i -th column is equal to ψ_i and $\alpha \in \mathbb{R}^{N \times 1}$ is a column vector constructed from the coefficients α_i . In this case we can write

$$x = \Psi \alpha. \quad (2.12)$$

Since we have assumed that x is compressible in the basis set defined by Ψ , then the coefficient vector α is nearly sparse. That is, only $k < N$ elements of α are significant.

The formal definition of compressibility is that, when we arrange the coefficients α_i in order of decreasing magnitude such that $|\alpha_1| \geq |\alpha_2| \geq |\alpha_3| \geq \dots \geq |\alpha_N|$, then

$$|\alpha_n| \leq C \cdot n^{-s}, \quad (2.13)$$

where C is a constant and $s \geq 1$ is another constant [62]. If α satisfies Equation 2.13 and if we reconstruct a signal x_k from only the k largest coefficients via

$$x_k = \sum_{j=1}^k \alpha_j \psi_j \quad (2.14)$$

then the error between x_k and x also follows a power law:

$$\|x_k - x\| \leq C \cdot k^{-(s-\frac{1}{2})} \quad (2.15)$$

We combine the sparsity transform Ψ and the measurement array Φ via $A = \Phi \Psi$. We can then write

$$y = \Phi x = \Phi \Psi \alpha = A \alpha. \quad (2.16)$$

Similar to Equation 2.8, a necessary and sufficient condition for some $\epsilon > 0$ for a compressible signal is that

$$1 + \epsilon \geq \frac{\|A v\|_2}{\|v\|_2} \geq 1 - \epsilon, \quad (2.17)$$

and the RIP is that the matrix A satisfies Equation 2.17 for any arbitrary $3k$ -compressible vector v . The definition of coherence provided in Equation 2.9 holds as well provided the matrix G is defined via $G = A^T A$. Incoherence in this case means that the measurement vectors ϕ_i cannot be sparsely represented by the sparsity vectors ψ_j .

We obtain the solution to 2.16 by solving the minimization problem

$$\hat{\alpha} = \operatorname{argmin} \|\alpha\|_1 \text{ such that } y = A\alpha. \quad (2.18)$$

The signal is then reconstructed using $\hat{x} = \Psi\hat{\alpha}$.

Noise will be present in a real system. This is handled by adding a noise term to this equation such that

$$y = A\alpha + n, \quad (2.19)$$

where $n \in \mathbb{R}^{M \times 1}$ is a vector representing the noise term. In that case, the optimization problem is

$$\hat{\alpha} = \operatorname{argmin} \|\alpha\|_1 \text{ such that } \|y - A\alpha\| \leq \delta, \quad (2.20)$$

where δ is a parameter that is not usually known *a priori*. This problem is known as basis pursuit denoising.

2.3. Reconstruction Techniques

A wide variety of reconstruction algorithms are available to solve the basis pursuit and basis pursuit denoising problems. A popular library of algorithms for MATLAB [22] is $\ell_1 - \text{MAGIC}$ by Emmanuel Candès and Justin Romberg [19]. This library provides second-order cone program solutions to seven problems of interest to compressive sensing.

Another popular algorithm for MATLAB is SPGL1 [72, 73]. SPGL1 can be used to solve the basis pursuit or basis pursuit denoising problems.

The Two-step Iterative Shrinkage/Thresholding Algorithm (TwIST) [7, 8] algorithm is widely used. MATLAB code for it is available as well.

Research into reconstruction algorithms is ongoing. Recent research has investigated the use of neural networks to perform compressive sensing reconstructions [9].

3. SINGLE-FRAME SUPER-RESOLUTION VIA A CODED APERTURE AND COMPRESSIVE SENSING

The potential to increase resolution of a signal has drawn the interest of the imaging community. For medical imaging modalities this has the potential to both shorten collection times and reduce dosages, providing benefits to both providers and patients. For synthetic aperture radar systems CS provides the potential to perform image reconstruction at reduced sampling rates without a loss in image quality. For electro-optic systems CS provides the potential to produce high quality imagery from smaller, simpler detectors.

Multi-frame super-resolution techniques can be used to combine multiple low-resolution images into a single high-resolution image [30, 31, 39, 61]. These techniques can potentially provide significant improvements in image resolution, but they require collecting multiple, laterally-displaced low-resolution so that aliased spatial frequencies can be unwrapped.

Other single-frame super-resolution techniques rely on using self-similarity [63, 77], information learned from dictionaries [58, 78], or deep learning algorithms trained on large volumes of imagery [71, 79]. At best these techniques provide relatively minor enhancements to image sampling. Furthermore, they introduce risk that artifacts added to the imagery will be artificial, present only in other areas of the same image or in the dictionary/training data.

The single-frame super-resolution technique employed here was developed by Marcia and Willett [54, 55, 56, 76]. This technique addresses the reconstruction of a high-resolution image from a single low-resolution measurement. For example, an $N \times N$ pixelated detector could be used to reconstruct an image of size $2N \times 2N$. This potentially results in a system that results in reduced size, weight, and power due to the reduced data collection and transmission needs.

The technique relies on generating placing a coded aperture in the pupil plane of an imaging system. A coded aperture is a mask with a user-specified distribution of transmittance and/or phase to manipulate light in a meaningful fashion. Coded apertures have been used to perform tasks such as depth of field extension [27], depth estimation[51], and imaging without lenses [23].

Marcia and Willett demonstrate how to calculate a coded aperture that allows single-frame super-resolution reconstruction. This technique builds upon theoretical work on Toeplitz-structure matrices for compressive sensing [3]. This theoretical work showed that when compressive sensing systems use a random circulant matrix as the system matrix, exact recovery is possible. Marcia and Willett demonstrate how to generate an imagery geometry that leverages this advancement.

Figure 3-1 shows the system architecture. An input scene is imaged with the coded aperture imaging system onto a pixelated array of size $n \times n$. The measured image looks little like the input

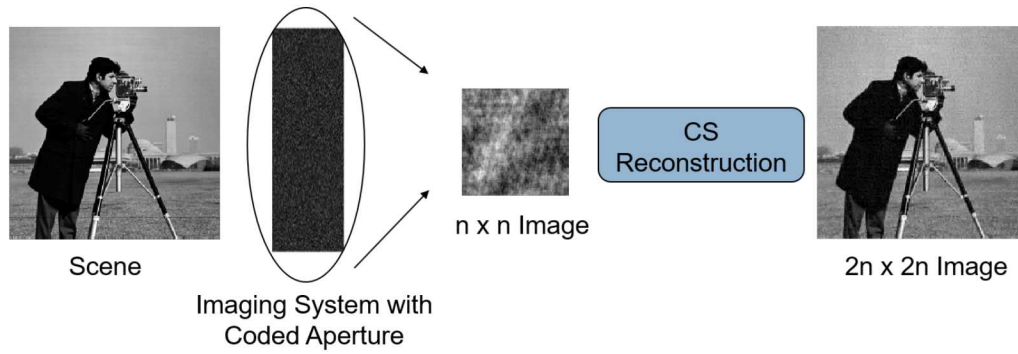


Figure 3-1. Coded Aperture Compressive Sensing Super-Resolution Architecture

scene. A compressive sensing reconstruction is performed to recreate an image of resolution $2n \times 2n$.

The reconstruction of the high-resolution image from the low-resolution measurement is computationally-intensive. Furthermore, the reconstruction does not add information to the system; all of the information present in the system is captured by the focal plane array. The reconstruction serves to put the information into a form that a human can understand. So if a machine learning task is going to be performed, does the reconstruction need to be performed at all?

Computer simulations were performed to investigate this question. The images used for this analysis come from the MNIST database of handwritten digits [40]. This database consists of a training set of 60,000 examples and a test set of 10,000 examples of handwritten digits in the range 0 to 10. These are grayscale images of size 28×28 pixels.

The simulation was performed assuming a pixelated detector of size 14×14 pixels. Classification was performed on four different data sets. The first data set was the original 28×28 MNIST imagery. Classification of this data set is used as comparison. The second data set was a traditional imaging system using 14×14 images that were downsampled from the original imagery. The third data set was the measured 14×14 raw compressed sensing images (i.e., without reconstruction). The last data set was the 28×28 image resulting from the compressive sensing reconstruction of the third data set.

The magnitude and phase of the coded aperture is shown in Figure 3-2. The average transmittance of the coded aperture is only 6%. This would be detrimental in a real-world application with noise as compared to a traditional imaging system without a mask. The radiometric throughput of the compressive sensing system is significantly reduced, thereby reducing the total system signal-to-noise ratio.

Figure 3-3 shows the point-spread function generated by the coded aperture imaging system.

First an investigation into the reconstruction error was performed. Figure 3-4 shows representative images of each class and their compressed sensing measurements.

Figure 3-5 shows the quality of the compressive sensing reconstructions as compared to the interpolation reconstructions. For each reconstruction technique the normalized mean square

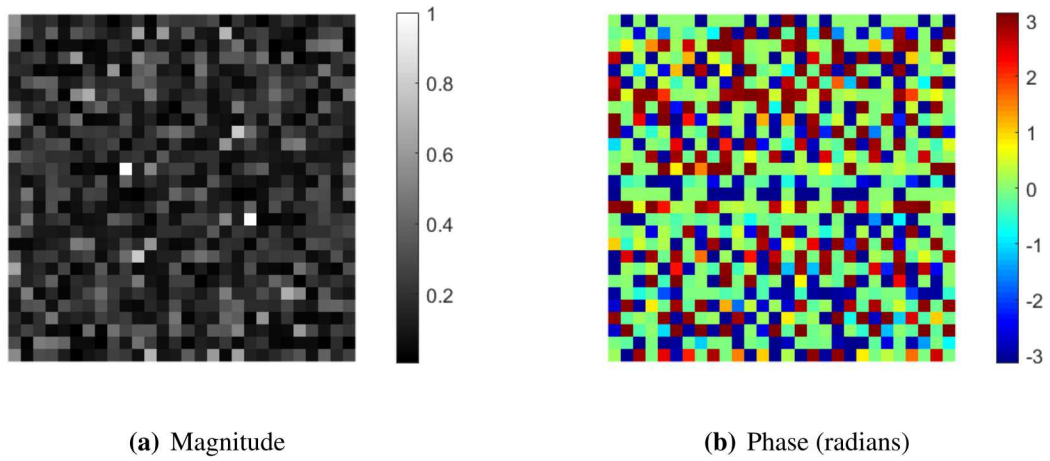


Figure 3-2. Coded Aperture Magnitude and Phase

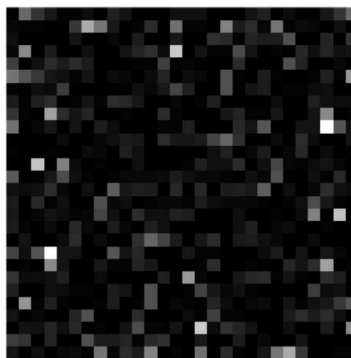


Figure 3-3. Coded Aperture Point-Spread Function

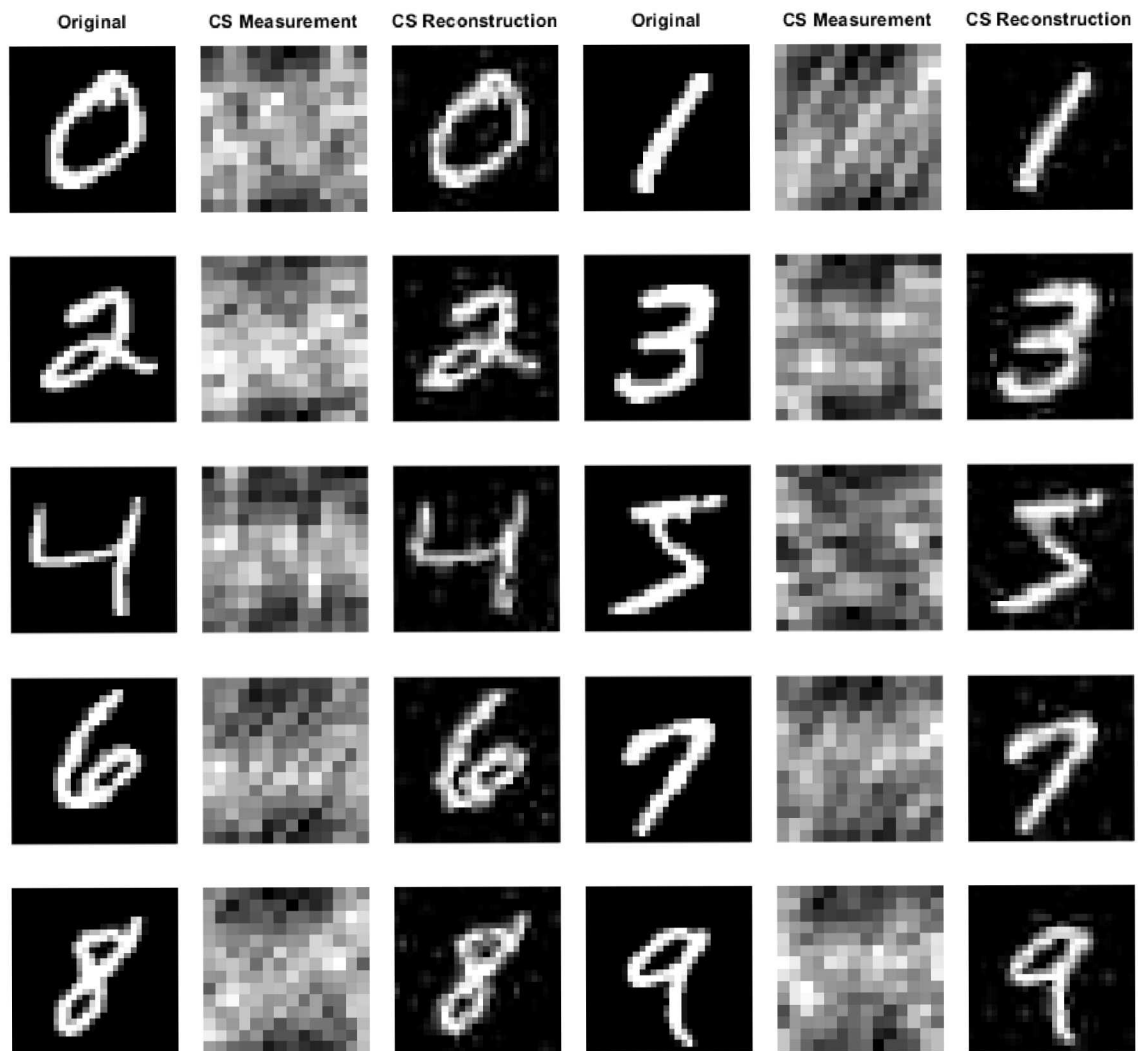


Figure 3-4. Compressive sensing measurements and reconstructions

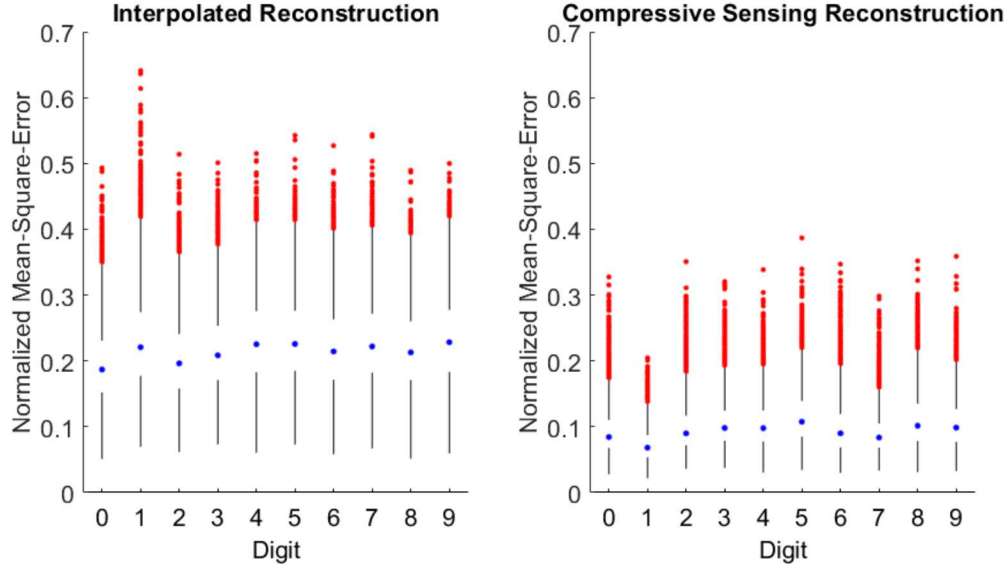


Figure 3-5. Interpolation and Compressive Sensing Reconstruction Errors

error (NMSE) is defined via

$$NMSE = \frac{\sum_{x,y} |g(x,y) - f(x,y)|^2}{\sum_{x,y} |f(x,y)|^2}, \quad (3.1)$$

where (x,y) denote the pixel coordinates, f denotes the original MNIST image, and g denotes the reconstructed image. In Figure 3-5 the black dots indicate the median error, the upper black line extends from the 75%th percentile to the highest point not considered an outlier, the lower black line extends from the 25%th percentile down to the smallest value not considered an outlier, and the red dots are statistical outliers. This figure demonstrates that a more accurate reconstruction of the original image is obtained with a 14×14 compressive sensing measurement as compared to a 14×14 traditional measurement followed by interpolation.

Classification was performed with a small convolutional neural network (CNN). Table 3-1 shows the network geometry for the case of classifying $N \times N$ images, where N is either 14 or 28. Table 3-2 show the classification results. The result from classifications based on the raw compressively sensed data is in line with the result from classifying the original imagery. Interestingly, classifications based on the downsampled imagery are in line with these other results as well. This is an indication that the images do not contain much information that would not be captured by a 14×14 sensor.

These results do indicate that the reconstruction is not necessary to perform the classification task. There are features in the compressive space that the CNN can find and use for accurate classification even though the images in this domain are not recognizable to a person.

Table 3-1. This table shows the network geometry for classifying NxN images.

Layer #	Type	Description
1	Image Input	NxNx1 images with 'zerocenter' normalization
2	Convolution	32 3x3 convolutions with stride [1 1] and padding [0 0 0 0]
3	ReLU	Rectified Linear Unit
4	Convolution	64 3x3 convolutions with stride [1 1] and padding [0 0 0 0]
5	ReLU	Rectified Linear Unit
6	Max Pooling	2x2 max pooling with stride [1 1] and padding [0 0 0 0]
7	Dropout	25% dropout
8	Fully Connected	128 node fully connected layer
9	Dropout	50% dropout
10	Fully Connected	10 node fully connected layer
11	Softmax	Softmax classifier
12	Classification	Crossentropyex

Table 3-2. Image classification accuracies

Imagery	Classification Accuracy (%)
28 × 28 traditional images	98.56
14 × 14 downsampled traditional images	98.82
14 × 14 raw compressively sensed (CS) images	97.14
28 × 28 reconstructions from CS images	98.78

4. COMPRESSIVE SENSING SNAPSHOT IMAGING SPECTROMETER

Compressive sensing techniques have application to a wide range of imaging modalities, including spectrometry. Whereas in the last chapter we considered the case of imaging, in this chapter we investigate the use of compressive sensing for spectrometry and imaging spectrometry.

Previous work demonstrated that a single liquid crystal retarder could be used in conjunction with a point detector to create a compressive sensing spectrometer [2]. They measured 1024 spectral bands by making only 108 measurements. Their measurements were made by temporally varying the voltage across the liquid crystal cell. Hence this approach does not directly translate to a snapshot spectrometer (i.e., a device capable of measuring spectral information in a single data collect).

However, if the spectral filter system were spatially tiled with a small number of cells and each cell were to have a different voltage, a Bayer-filter-like sensor could be created. A Bayer-filter, shown in Figure 4-1, is the type of spectral filter used in the vast majority of digital cameras. It consists of a 2x2 tiling of red, green, and blue spectral filters (generally two green filters are used). Spatial resolution is traded for spectral capability. With a traditional Bayer filter 4 measurements are used to obtain 3 spectral bins. In a traditional spectrometer M measurements results in M spectral bins; with compressive sensing we aspire to obtain $N > M$ spectral bins.

The trade-off with a Bayer filter geometry is that spectral resolution is obtained at the expense of reduced spatial resolution. The spatial resolution of the system is governed by the size of the super-pixels rather than the individual pixels.

This tiled geometry is shown in Figure 4-2. This geometry is referred to as the Compressive Sensing Snapshot Imaging Spectrometer (CSSIS). Broadband light is incident on an array of fixed spectral filters. In this figure a 4x4 geometry is shown, corresponding to 16 measurements. This

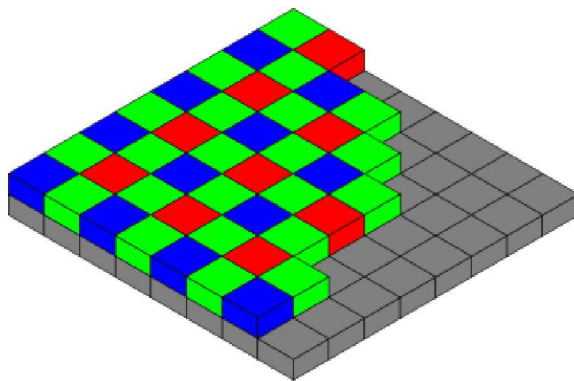


Figure 4-1. Traditional Bayer Filter Geometry

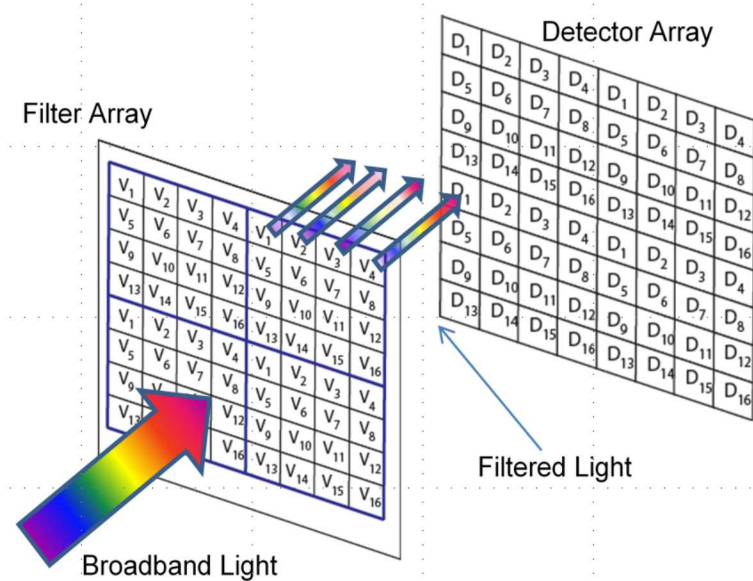


Figure 4-2. CSSIS Geometry

4x4 pattern defines a super-pixel and the pattern repeats across the entire detector array. In this chapter a "pixel" corresponds to the pattern seen by a single filter element. A "super-pixel" corresponds to a block containing a complete set of spectral filters.

The spectral filters need not be liquid crystal filters. They can also be Fabry-Perot filters [60], form birefringence filters, nano-antenna array filters, or any other type of device which can vary its transmittance (or reflectance) as a function of wavelength.

This approach has several benefits. First, it is a true snapshot spectrometer; spectra are captured at the sample rate of the detector. This allows for spectral analysis of fast transient events. Furthermore, no temporal scanning or physical motion is required. Next, the system optical throughput is high. A traditional tiled spectrometer system employs narrow-band spectral filters that reject most of the light, leaving relatively little to be detected. The average transmittance of each filter in the CSSIS is much higher, perhaps 50%. Radiometric efficiency is significantly increased as much more light strikes the detectors. Lastly, since compressive sensing techniques can be used, the number of spectral bins is greater.

4.1. CSSIS Trade Space

The traditional tiled-spectrometer geometry is used for a variety of commercial applications. Commercial devices with array sizes of 4x4 and 5x5 are available [21]. Employing compressive sensing techniques dramatically changes the trade space in designing such systems.

4.1.1. *Improved Spectral Resolution with Constant Spatial Resolution*

One way this system could be leveraged is to provide increased spectral resolution while maintaining a constant spatial resolution. Consider replacing a 4x4 traditional filter array from an existing commercial system with a 4x4 compressive sensing filter array. Both of these systems have degraded the resolution of the imaging system by 4X since a 4x4 grid is used.

The traditional system uses its 4x4 filter array to obtain 16 spectral bins. However, the compressive sensing system could potentially use its 4x4 filter array to obtain 80 spectral bins. Thus the spectral resolution has been increased by 5X while maintaining the same spatial resolution.

4.1.2. *Improved Spatial Resolution with Constant Spectral Resolution*

The compressive sensing architecture could also be used to increase area coverage while maintaining spectral resolution. Consider replacing a 5x5 traditional filter array from an existing commercial system with a 3x3 compressive sensing filter array. The traditional system has degraded the resolution by 5X while the compressive sensing system has degraded the resolution by only 3X.

The traditional system obtains 25 spectral bins from its 25 spectral filters. The compressive sensing system could likewise obtain 25 spectral bins from its 9 spectral filters. Thus the spatial resolution of the compressive sensing system is better while the spectral resolution is the same.

4.1.3. *Improved Area Coverage with Constant Spectral and Spatial Resolutions*

Consider extending the previous example of a 5x5 traditional filter array and a 3x3 compressive sensing filter array. The compressive sensing system could be flown at a higher altitude so that its spatial resolution matches that of the traditional system. By flying the compressive sensing system 60% higher than the traditional system, the resolutions of the two systems would match. However, the area coverage of the compressive sensing system would be 2.56X the area coverage of the traditional system. Hence the compressive sensing system could see more of the ground at one time.

4.2. Spectral Super-Resolution

First, a study to determine the ability to obtain more spectral bins than measurements was conducted. A geometry using a 7x7 array of Fabry-Perot spectral filters was assumed, yielding $M = 49$ measurements. All filters were assumed to have the same light incident upon them (i.e., no spatial variation). The wavelength regime considered was in the long-wave infrared (LWIR): 7.8 to 12.7 μm . The reflectance of the mirrors of the Fabry-Perot etalons were assumed to be 80%. The mirror separations were constrained to be in the range of 40 to 100 μm and were optimized to minimize the system matrix coherence when the Daubechies-8 wavelets were used. Figure 4-3

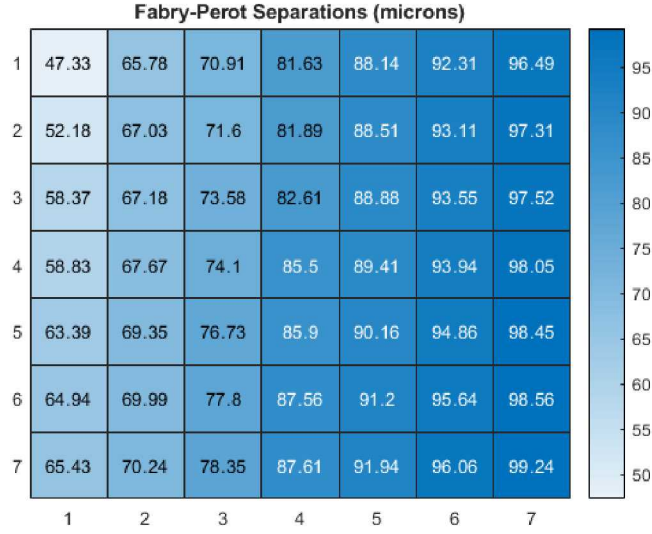


Figure 4-3. Fabry-Perot Filter Thicknesses

shows the filter thicknesses and Figure 4-4 shows the spectral transmittances as a function of wavelength.

Two spectra were simulated. Spectrum 1 consists of a Gaussian centered at a wavelength of $10.24 \mu\text{m}$ with a full-width half-maximum (FWHM) of $0.191 \mu\text{m}$. Spectrum 2 consists of a pair of Gaussians with the same FWHM of $0.191 \mu\text{m}$ but also separated by $0.191 \mu\text{m}$ centered about $11.49 \mu\text{m}$. Both spectra were normalized to have unity energy. Figure 4-5 shows these two spectra sampled with $N = 196$ data points.

When these spectra are downsampled to 64 spectral bins, the spectral resolution is diminished such that the dip in Spectrum 2 is no longer resolved. Figure 4-6 shows the results of the downsampling operation.

Measurement with the CSSIS was simulated and the results are shown in Figure 4-7. Note that 64 measurements are made, the same number as in the downsampled case.

Reconstructions from the measurements shown in Figure 4-7 are shown in Figure 4-8. As can be seen from this figure, the information about the dip is preserved. This indicates that more information is stored in the compressive sensing data than in the traditional spectrometer data.

Next, the impact of spatial variations was investigated. An optical imaging system with an f-number (defined as the ratio of the focal length to the aperture diameter) of 15 was used. The focal plane pixels were assumed to be $12 \mu\text{m}$ in size. The system point-spread function as a function of wavelength was modeled using standard Fourier optics theory. Figure 4-9 shows the PSF at a central wavelength at high spatial resolution. Figure 4-10 shows the PSF at a resolution of $12 \mu\text{m}$, the size of an individual filter. Figure 4-11 shows the PSF at a resolution of $84 \mu\text{m}$, the size of a super-pixel. The ensquared energy in a super-pixel ranges from 35.9% at the shortest wavelength to 16.5% at the longest wavelength.

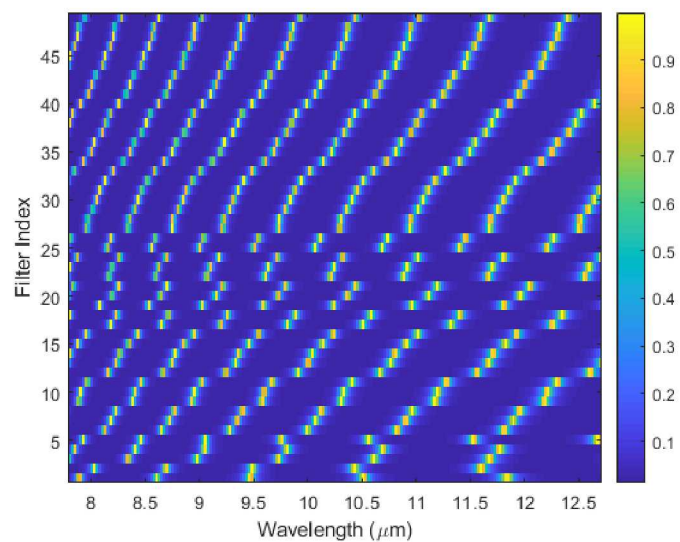


Figure 4-4. Super-Resolution Model Measurement Matrix

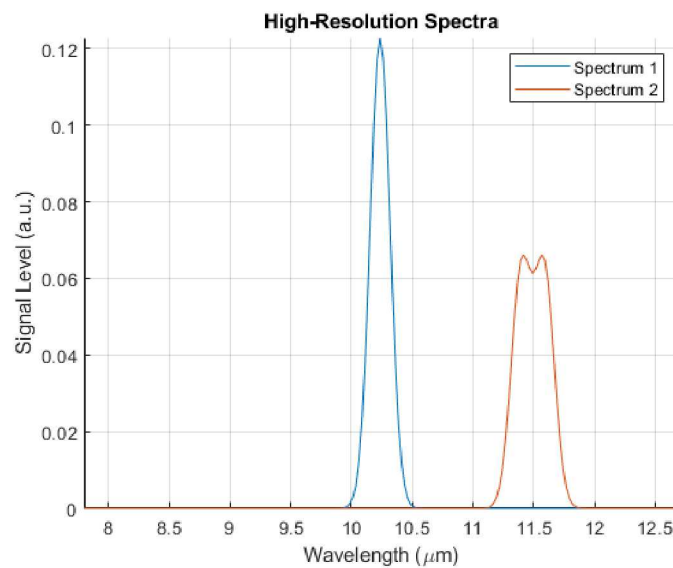


Figure 4-5. Spectra used for Spectral Super-Resolution Analysis

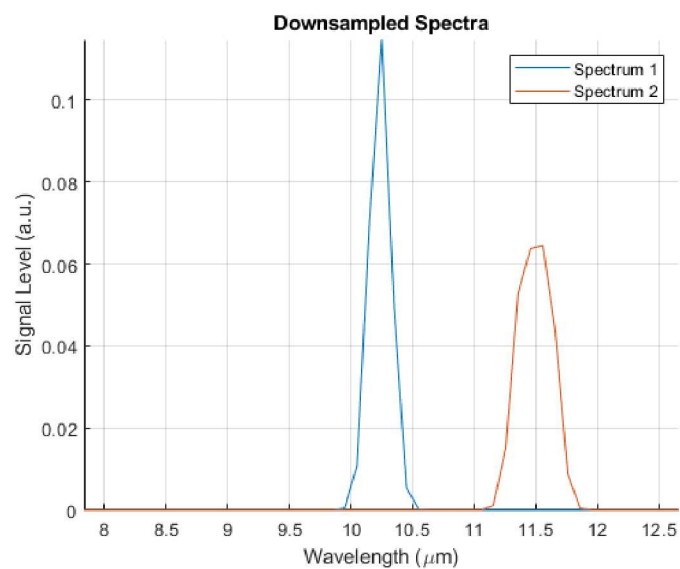


Figure 4-6. Downsampled Spectra

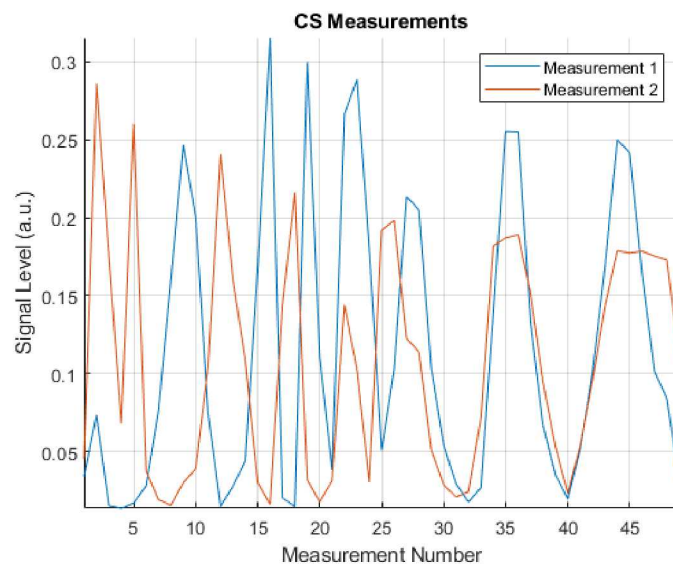


Figure 4-7. CSSIS Measurements of Spectra 1 and 2

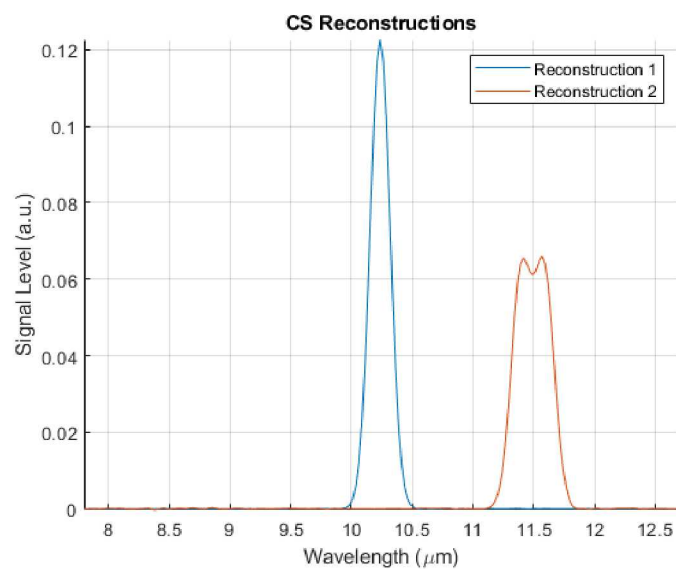


Figure 4-8. CSSIS Reconstructions of Spectra 1 and 2

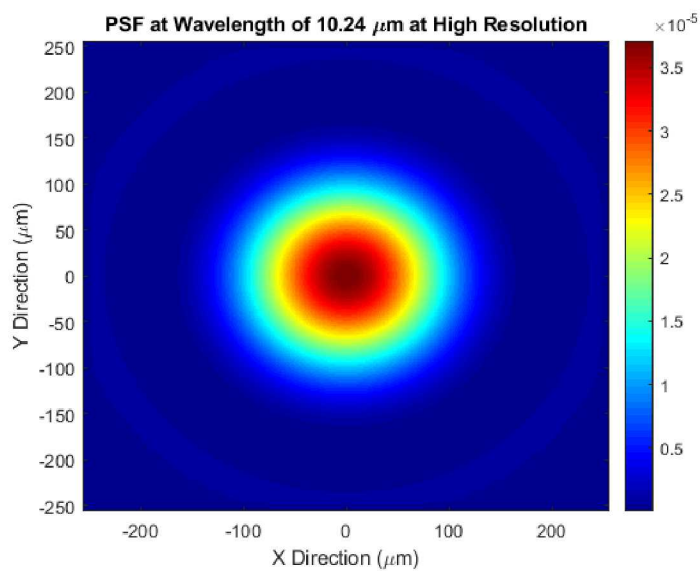


Figure 4-9. Point-Spread Function at 10.24 μm

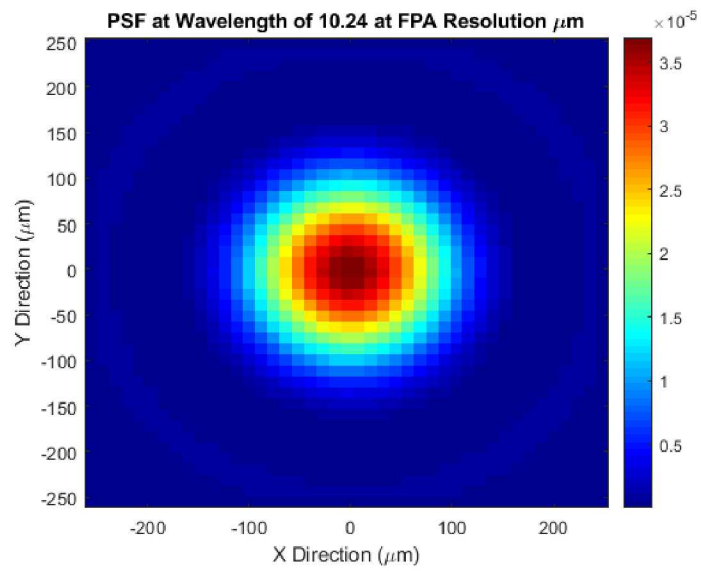


Figure 4-10. Point-Spread Function at 10.24 μm at Pixel Resolution

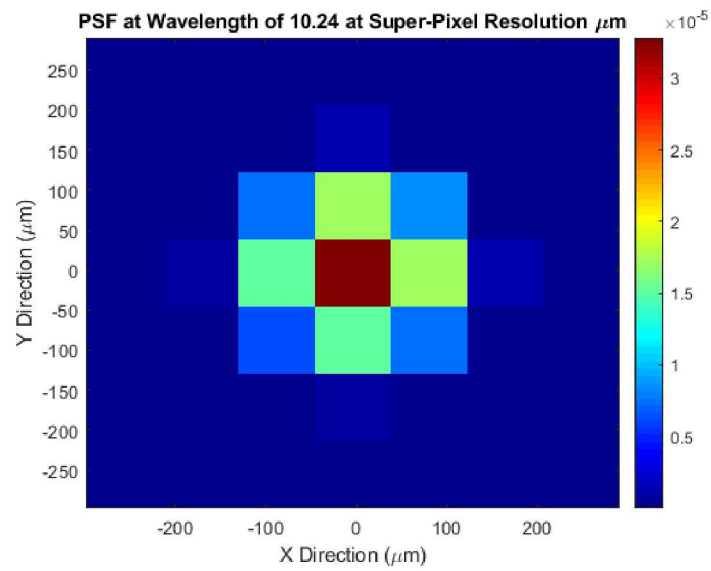


Figure 4-11. Point-Spread Function at 10.24 μm at Super-Pixel Resolution

A very simple spatial distribution of these two spectra was assumed. This distribution is shown in Figure 4-12. There is no variation in the Y-direction and a discontinuous change from Spectrum 1 to Spectrum 2 in the X-direction. These spatial distributions are convolved with the monochromatic point-spread functions to create spectral images at each wavelength. These spectral images are then sensed by the spectrometer.

Two spectrometers were simulated. The first is the CSSIS configuration described above. The second is a traditional spectrometer with 49 non-overlapping, perfect square-wave narrowband filters. The raw measurements from these systems are shown in Figures 4-13 and 4-14.

Spectral reconstructions shown across a slice in the X-direction are shown in Figures 4-15 and 4-16. Reconstructions in the Y-direction are not shown since there is no variation in the spectra in that dimension. These figures show that the CSSIS system is able to reconstruct the spectra accurately and that the effects of the spatial variation are localized to the spectral change boundary.

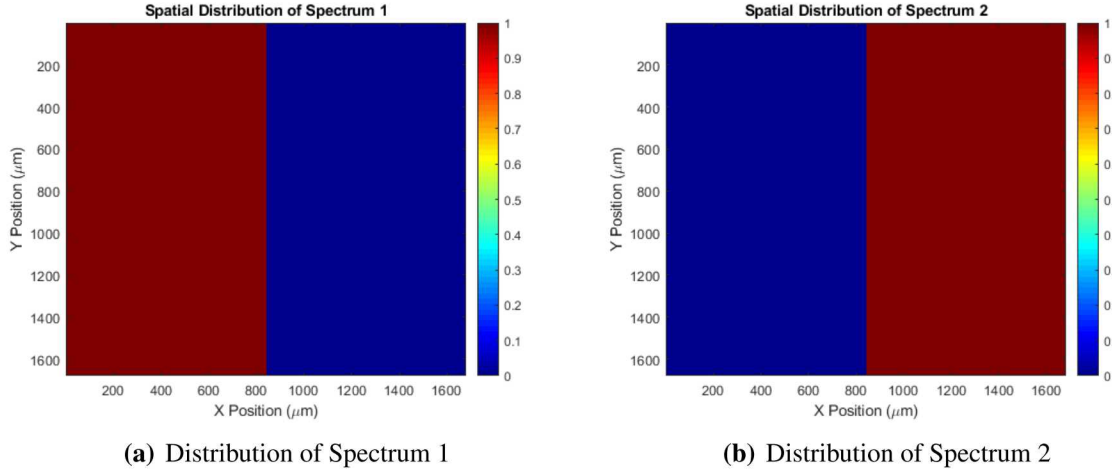


Figure 4-12. Spatial Distributions of Spectra 1 and 3

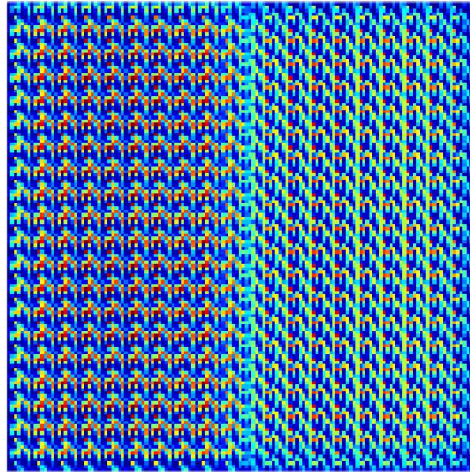


Figure 4-13. CSSIS Raw Image

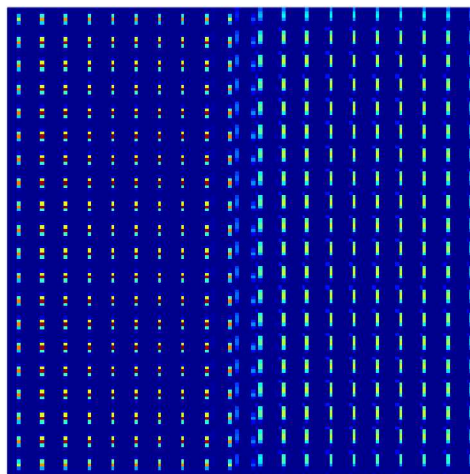


Figure 4-14. Traditional Spectrometer Raw Image

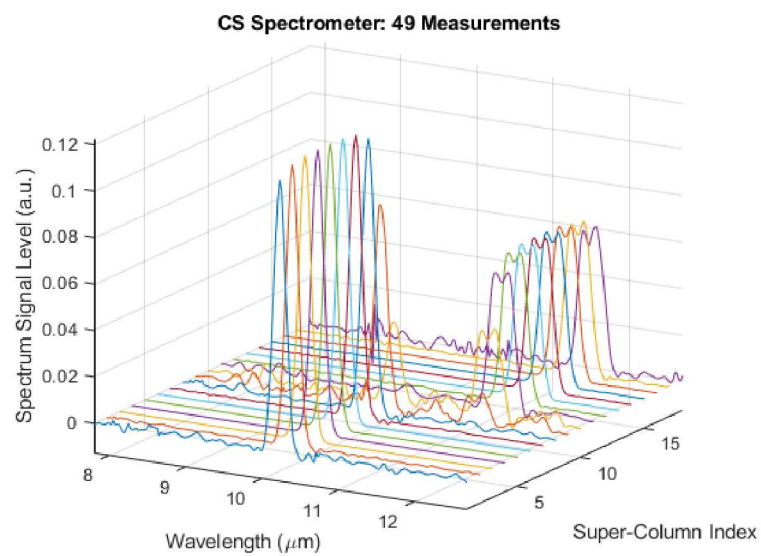


Figure 4-15. CSSIS Raw Image

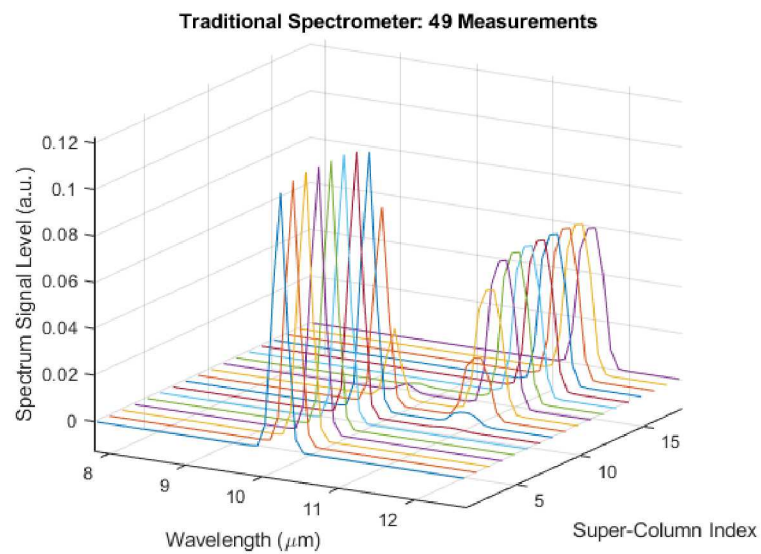


Figure 4-16. Traditional Spectrometer Raw Image

4.3. Spectral Classification

In the previous section we demonstrated that compressive sensing techniques can potentially store more information in the same number of measurements as compared to a traditional spectrometer system. In this section we explore how this impacts classification accuracy.

The Indian Pines data set is a hyperspectral image collected over the Purdue University agronomy farm in West Lafayette, Indiana [5]. The 145×145 image contains 220 spectral bands in the range $0.4 \mu\text{m}$ to $2.5 \mu\text{m}$ measured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). This data set includes ground truth data as well, making it useful for classification studies.

A different implementation of the CSSIS was used for this study. A 7×7 array of liquid crystal devices (LCDs) were used rather than Fabry-Perot spectral filters. The change in index of refraction of the LCD was varied between 0.001 and 1 in a linear fashion with an LCD thickness of $15 \mu\text{m}$. Figure 4-17 shows the spectral transmittances of the filters. The Haar wavelet basis set was used.

For reconstructions, the ℓ_1 -norm of the total variation (TV) of the wavelet coefficients was minimized rather than the ℓ_1 -norm of the coefficients themselves. The total variation of a signal is the root-sum-square of the horizontal and vertical discrete gradients. Let $D_{h,i,j}(f)$ denote the horizontal discrete gradient of the signal f . Similarly let $D_{v,i,j}(f)$ denote the vertical discrete gradient f . We define $D_{h,i,j}(f)$ and $D_{v,i,j}(f)$ via

$$D_{h,i,j}(f) \begin{cases} f_{i+1,j} - f_{i,j} & i < n \\ 0 & i = n \end{cases} \quad (4.1)$$

and

$$D_{v,i,j}(f) \begin{cases} f_{i,j+1} - f_{i,j} & j < n \\ 0 & j = n \end{cases}. \quad (4.2)$$

The total variation is then defined via

$$TV(f) \equiv \sum_{i,j} \sqrt{(D_{h,i,j}(f))^2 + (D_{v,i,j}(f))^2}. \quad (4.3)$$

By minimizing the total variation of the wavelet coefficients, we force them to go to a constant value. Since we know the signal is sparse, this value is zero. This optimization technique worked better for this study than minimizing the ℓ_1 -norm of the coefficients themselves.

A representative signal for each of the 17 classes in the Indian Pines data set was calculated by averaging all spectra assigned to that class. Figure 4-18 shows these spectra (blue lines) along with the CSSIS reconstructions (red lines). A compression factor of 4 was used, meaning that the 49 spectral measurements were used to produce 196 spectral bins.

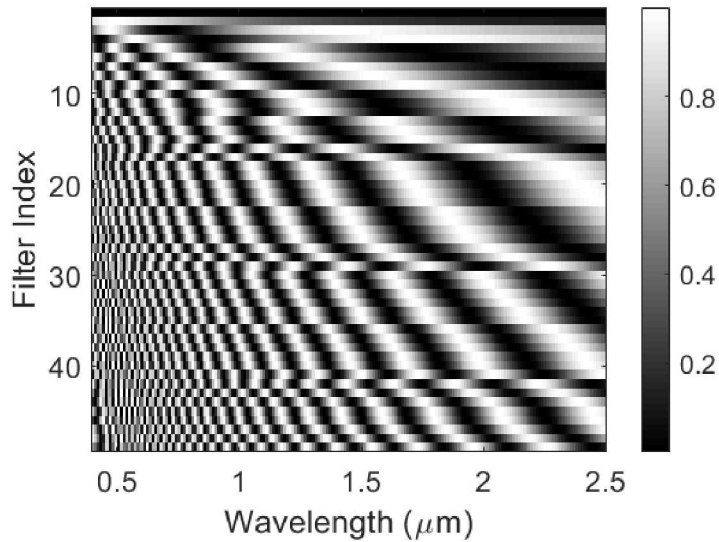


Figure 4-17. CSSIS Measurement Matrix for the Indian Pines Reconstruction Study

Next, a classification study was conducted. The performance of the CSSIS system was compared with the performance of a traditional 7x7 tiled spectrometer array. In the traditional spectrometer the spectral filters were again considered to be ideal, non-overlapping square-wave filters with unity transmittance. The detector modeled for this analysis was assumed to have a 12-bit analog-to-digital converter (ADC) and full well depth corresponds to 100,000 electrons. A read noise of 4 electrons was assumed and the signals were nominally set to fill the well halfway. A Gaussian background noise was assumed to be present as well, with standard deviations of 0, 5, 10, 15, or 20 counts. For each class in the Indian Pines data set, 1,000 measurements were simulated. 70% of these were used for training purposes and 30% of these were used for testing purposes.

Classifications were performed on three different data sets. The first was the traditional spectrometer with 7x7 non-overlapping, ideal narrowband spectral filters. The second was the CSSIS raw data, consisting of 49 measurements through the CSSIS filters. The third was the CSSIS reconstruction into 196 spectral bins. In this case, the noise was added to the CSSIS data prior to reconstruction to simulate the physics of the device.

The classification algorithm was k-Nearest Neighbor classification where 5 nearest neighbors were used in the predictors. Figure 4-19 shows the results of these classifications. The classification accuracy is highest when performed on the CSSIS raw data. Performing classification on the reconstructed data reduces the accuracy somewhat in the presence of noise. The traditional spectrometer performed the poorest.

One reason for the relatively poor performance of the traditional spectrometer in the presence of noise is the narrowband nature of the filters. The average transmittance across the spectral band for each filter is $1/49 = 2.04\%$. The average transmittance for the CSSIS spectral filters is roughly 50%. Therefore the CSSIS has a significant advantage in radiometric throughput over the traditional spectrometer. This is shown in Figure 4-20. In this case we define the signal-to-noise

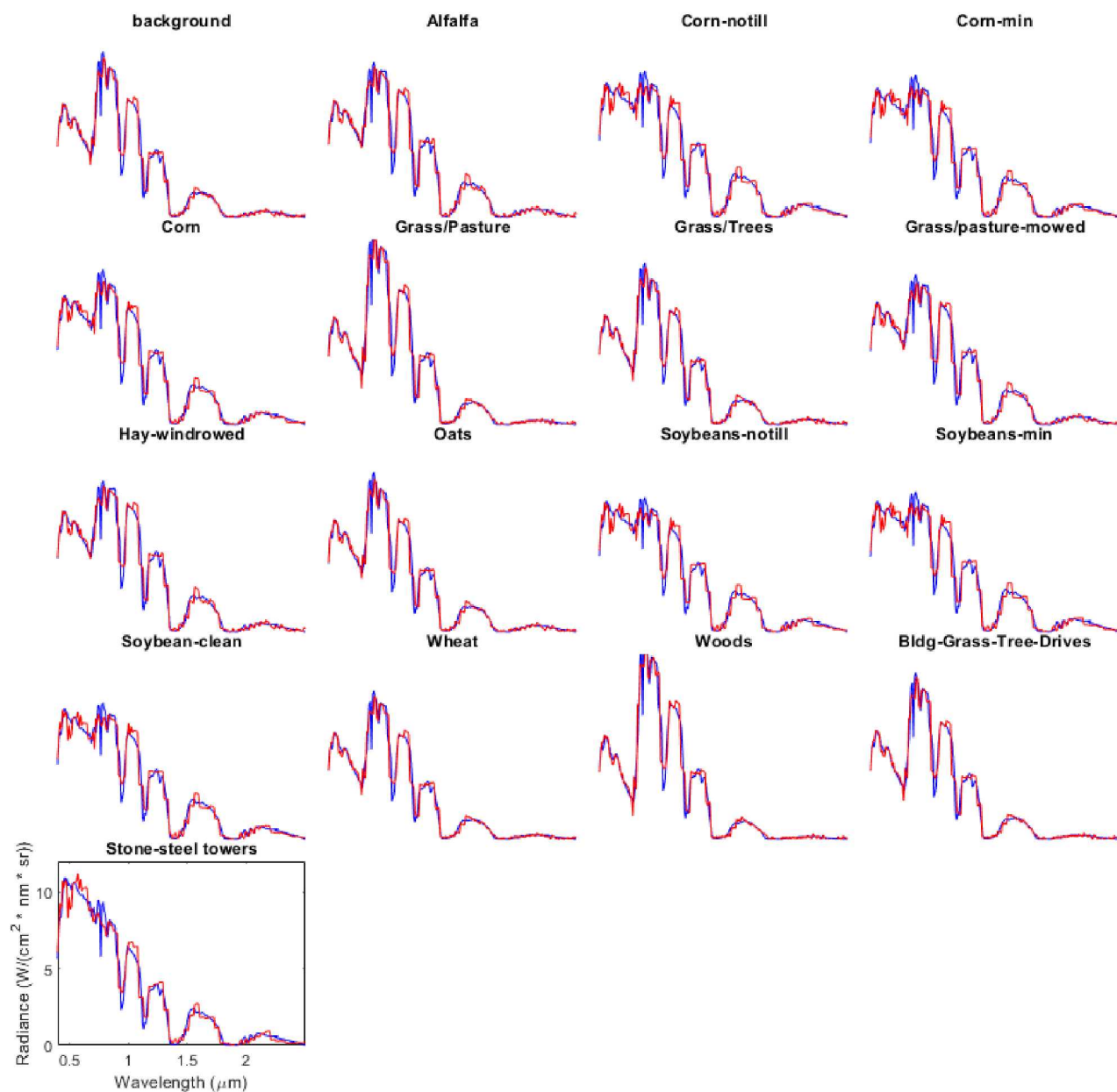


Figure 4-18. Indian Pines spectra (blue) and reconstructions (red) are shown. The last sub-plot shows the wavelength and radiance ranges used on all plots.

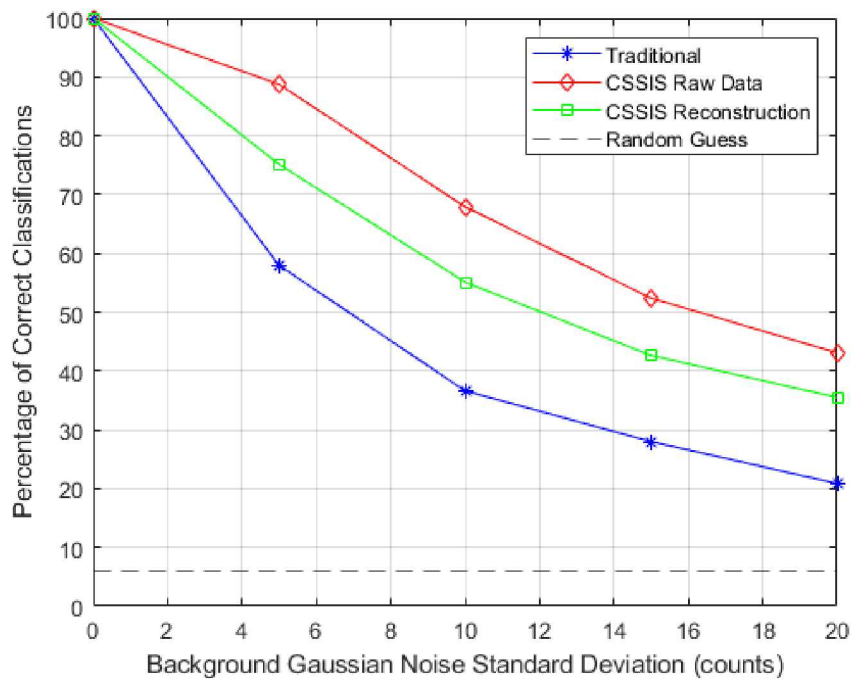


Figure 4-19. Classification Accuracy on Indian Pines Data Set

ratio (SNR) of the system as the mean of the spectral signal divided by its standard deviation. The CSSIS raw data has a significantly higher SNR in this sense that the traditional spectrometer or the CSSIS reconstruction.

The traditional spectrometer performance could likely be increased by increasing the integration time, but in that case the temporal performance of the spectrometer is reduced.

This study demonstrated that classification can be performed on the CSSIS raw data. In fact, the classification accuracy is higher for the CSSIS in the presence of noise as compared to an idealized traditional pixelated spectrometer.

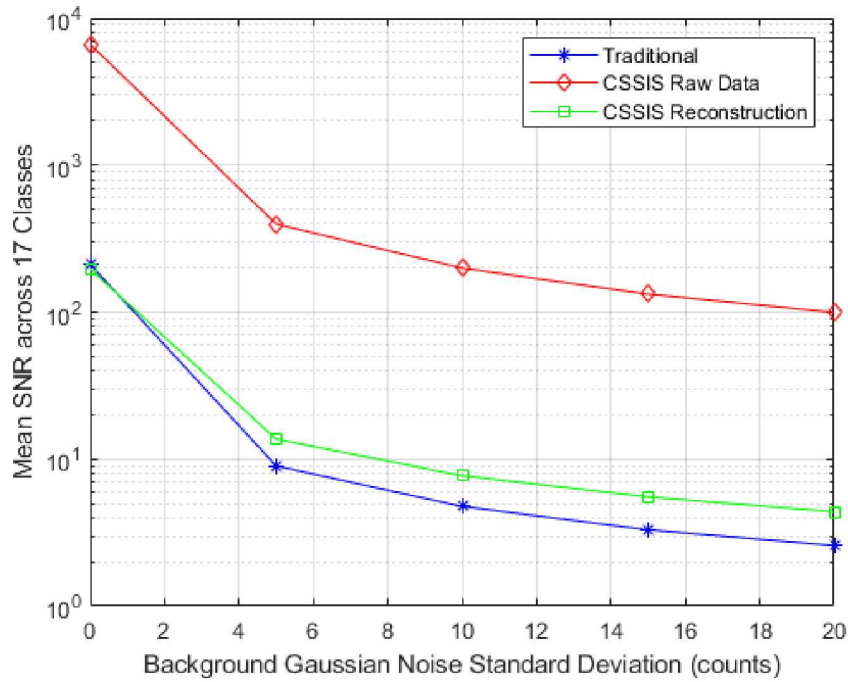


Figure 4-20. SNR of CSSIS and Traditional Spectrometer Signals

4.4. Indian Pines Data Set with Intraclass Variation

The previous study demonstrated that the CSSIS could distinguish between the classes in the Indian Pines data set. Intraclass variation was artificially induced via simulated read noise and Gaussian noise. Next, a study with the actual intraclass variation present within the Indian Pines data set was used.

The same CSSIS system as the previous section was used. The data used were the 145x145 spectral measurements in the Indian Pines data set. Ran *et al*, demonstrated that the Indian Pines data set can be effectively classified using a combination of an adaptive weighted filter (AWF) followed by a local Fisher discriminant analysis(LFDA) and K-nearest-neighbor (KNN) classification[65]. That same approach was used here, with one change. The AWF was modified to also include a Gaussian spatial weight distribution in the same manner as traditional bilateral filtering used for noise removal in digital images [70]. This filter is referred to here as the bilateral adaptive weighted filter (BAWF). The filter used a window size of 7 pixels and a spatial Gaussian distribution with a $1/e$ -width of 3 pixels.

The Indian Pines data set were corrupted with noise in a manner similar to that previously described. For this simulation, noise was added directly to the Indian Pines data, though, rather than to the averages of the spectra of the distinct classes. 70% of the 145x145=21,025 pixels were used as the training set while the rest were used for testing. The following measurements were analyzed:

- Traditional: A traditional pixelated spectrometer using the BAWF-LFDA-KNN algorithm

- **CSSIS Raw Data:** The CSSIS spectrometer using the BAWF-LFDA-KNN algorithm on the raw CS data without reconstruction
- **CSSIS Recon before Filtering:** The CSSIS spectrometer where TV-reconstruction was performed and then followed with the BAWF-LFDA-KNN algorithm
- **CSSIS Recon after Filtering:** The CSSIS spectrometer where the BAWF algorithm was performed on the TV-reconstructed spectra; LFDA-KNN were then performed on the reconstructed spectra.

Figure 4-21 shows the classification accuracy as a function of noise level for the different measurement techniques considered. For all noise levels the performance is best when the data processing and classification are performed directly on the data in the compressed space. If the reconstruction is needed, the classification task works better if the total variation reconstruction is performed prior to the filtering operation.

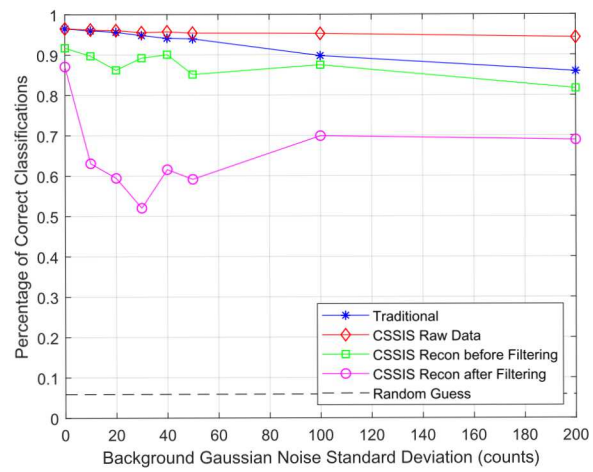


Figure 4-21. Classification Accuracy vs. Noise of the Indian Pines Data Set

4.5. Conclusion

The benefits of a compressive sensing snapshot imaging spectrometer have been demonstrated. This study demonstrated that a tiled array spectrometer based on compressive sensing techniques can outperform a traditional tiled array spectrometer. The CSSIS can be used to reconstruct spectral features that are lost in a traditional spectrometer. It also provides a radiometric throughput advantage that results in more light striking the imaging sensor.

These benefits directly impact performance of a classification task. Spectral signatures can be more accurately classified with a CS system than a traditional system, especially in the presence of noise.

5. COMPRESSIVE HYPERSPECTRAL IMAGING USING TOTAL VARIATION MINIMIZATION

5.1. Introduction

Hyperspectral imaging provides a key capability for diverse applications, including remote sensing and biomedical imaging [29, 67]. Sensors can measure spectral signals directly or indirectly. Direct methods include gratings that disperse light and filters that select wavelength bands. Indirect techniques, such as infrared Fourier Transform and Hadamard spectroscopy [59], multiplex or encode bands and can improve signal-to-noise ratio at the cost of a less intuitive sensor design and more intricate post-processing.

Compressive sensing shows promise for sensors that collect fewer samples than required by traditional Shannon-Nyquist sampling theory [25]. Recent sensor designs modulate light with spatial light modulators, liquid crystal phase retarders, and Fabry-Perot resonators [1, 60]. These sensors encode spectra with less measurements than the number of bands in the signal, making reconstruction an underdetermined problem [48, 47]. A single measurement consists of an inner product of the transmission profile of the spectral modulator (e.g., a Fabry-Perot resonator) with the spectral signal. The modulator filters incoming light according to the transmission. Reconstruction algorithms like basis pursuit [75] assume that the unknown signal is sparse relative to a basis such as the wavelet transform.

We propose a reconstruction algorithm for hyperspectral images encoded through spectral modulators. Our approach constrains pixels to be similar to their neighbors in space and wavelength, as natural images tend to vary smoothly [12]. It combines L_1 minimization in the wavelet domain to enforce sparsity, as inspired by compressive sensing, and total variation in the image domain for smoothness. Our algorithm constrains encoded, compressed hyperspectral images to be smooth in their reconstruction.

5.2. Theory

5.2.1. Notation

We represent the hyperspectral image x as a tensor of size $X \times Y \times B$, where X and Y are the number of pixels in the x and y directions respectively, and B is the number of spectral bands. $x_{i,j}$ denotes pixel (i, j) of the hyperspectral image, and it is a vector of length B . Taking the wavelet transform of each pixel results in $\hat{x}$ of size $X \times Y \times C$, where C is the number of wavelet

coefficients per pixel. $\hat{x}_{i,j}$ denotes the wavelet coefficients at pixel (i, j) , and it is a vector of length C . The matrix W transforms the wavelet coefficients back to the spectral signal:

$$x_{i,j} = W\hat{x}_{i,j}. \quad (5.1)$$

Table 5-1 summarizes the notation used in this paper.

Table 5-1. The counting variables are integers, while the optical and optimization variables consist of real numbers.

Type	Variable	Description	Shape
Counting variables	B	Number of spectral bands	1
	C	Number of wavelet coefficients	1
	M	Number of measurements	1
	X	Number of pixels in the x direction	1
	Y	Number of pixels in the y direction	1
Optical variables	d	Gap width between Fabry Perot mirrors	$M \times 1$
	F	Optical finesse	1
	λ	Wavelength	1
	n	Index of material between Fabry Perot mirrors	1
	R	Mirror reflectivity	1
	θ	Incident angle of light	1
Optimization variables	A	System model	$M \times C$
	α	Total variation weights: α_x , α_y , and α_λ	3×1
	$b_{i,j}$	Variable used in updating x	$B \times 1$
	η	Penalty term in the augmented Lagrangian	1
	n	Noise	$X \times Y \times M$
	p	Dual variable	$X \times Y \times B$
	$p_{i,j}$	Dual variable from pixel (i, j) of p	$B \times 1$
	T	Fabry perot transmissions	$M \times B$
	W	Discrete wavelet transform	$B \times C$
	x	Hyperspectral image	$X \times Y \times B$
	$x_{i,j}$	Spectrum from pixel (i, j) of x	$B \times 1$
	$\hat{x}$	Wavelet coefficients	$X \times Y \times C$
	$\hat{x}_{i,j}$	Coefficients from pixel (i, j) of $\hat{x}$	$C \times 1$
	y	Measurements	$X \times Y \times M$
	$y_{i,j}$	Measurements from pixel (i, j) of y	$M \times 1$

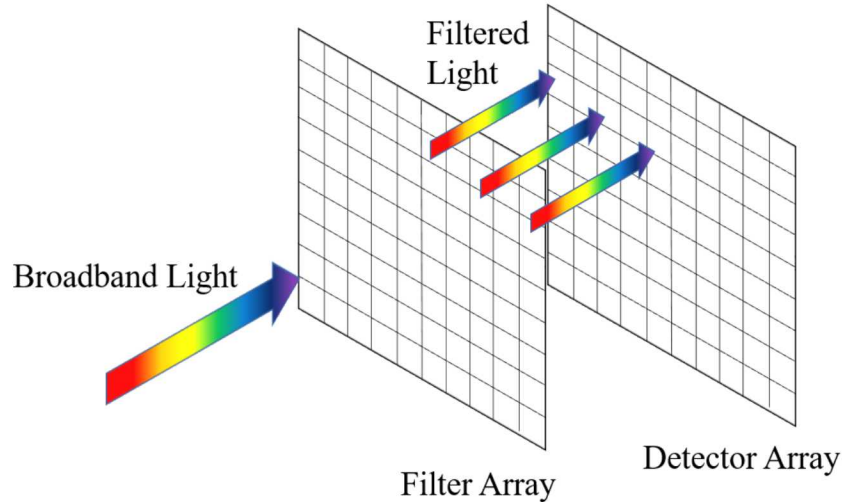


Figure 5-1. The filter array consists of Fabry-Perot resonators, and the detector array can be a focal plane array. The Fabry-Perot resonators modulate the spectrum of each pixel. A single measurement results from the inner product of the filter transmission and the light spectrum. The resonators filter incoming light according to the transmission. Our filters vary with time because the Fabry-Perot resonators modulate their mirror spacing to produce different transmissions and measurements. Alternatively, the filters can vary in space; a cluster or “superpixel” of filters can provide measurements of one area, enabling snapshot hyperspectral imaging.

5.2.2. *Measurement model*

Hyperspectral imagers can measure light directly or indirectly. Direct methods include narrow band spectral filters and gratings to disperse light. In contrast, indirect techniques, such as infrared Fourier transform spectroscopy and Hadamard coding spectroscopy [59], multiplex or encode spectral bands, and they can improve signal-to-noise ratio at the cost of a less intuitive sensor design and more intricate post-processing. Compressive sensing shows promise for sensors that collect fewer samples than required by traditional Shannon-Nyquist sampling theory [44, 46]. Recent sensor designs modulate light with spatial light modulators, liquid crystal phase retarders, and Fabry-Perot resonators. These sensors encode spectra with less measurements than the number of bands in the signal, making reconstruction an underdetermined problem.

Mathematically, spectral modulators compute an inner product between their transmission and a spectral signature, resulting in a single measurement. Changing the transmission profile creates a different measurement. The modulator filters incoming light according to the transmission. The analysis in this work applies to spectral modulators in general. We model the transmission of a spectral modulator through a transmission matrix T , and different transmission profiles will change the T matrix, as described in this section.

Figure 5-1 illustrates the concept of a hyperspectral imager using a filter array and detector array. In this paper, the hyperspectral imager consists of Fabry-Perot resonators on top of a focal plane array or detector array. A Fabry-Perot resonator modulates each pixel of the focal plane array [1]. We simulate a measurement on one pixel by computing an inner product between the resonator transmission and the spectral signature. The resonators filter incoming light according to the transmission. Changing the mirror spacing in a resonator, as described in this section, results in a

different transmission profile, and hence a new measurement. Our filters vary with time because the Fabry-Perot resonators modulate their mirror spacing to produce different measurements. Alternatively, the filters can vary in space; a cluster or “superpixel” of filters can provide measurements of one area, enabling snapshot hyperspectral imaging. The superpixel approach trades a loss of spatial resolution for a gain in temporal resolution (snapshot imaging).

In a Fabry-Perot resonator, two partially reflective mirrors create constructive and destructive interferences at periodic wavelengths. Piezo-controlled actuators control the precise spacing between the mirrors, with response rates of hundreds of Hertz. Let R be the reflectivity of both mirrors in the Fabry-Perot resonator. The finesse F is

$$F = \frac{4R}{(1-R)^2}. \quad (5.2)$$

Let n be the refractive index of the material between the mirrors, d be the distance between the mirrors, θ be the angle of incidence of light, and λ be the wavelength of light. The optical thickness δ is

$$\delta(d_i, \lambda) = \frac{2\pi(2nd)\cos(\theta)}{\lambda}. \quad (5.3)$$

The transmission of the Fabry-Perot resonator depends on d and λ . For the i th measurement, we set the mirror spacing to be d_i . The transmission profile of the i th measurement is

$$t_i(d_i, \lambda) = \frac{1}{1 + F \sin^2(\delta(d_i, \lambda)/2)}. \quad (5.4)$$

The transmission matrix T consists of the transmission profiles from all measurements. Each of the M rows represent one measurement, and each of the B columns signify a different wavelength:

$$T = \begin{bmatrix} t_1(d_1, \lambda_1) & \dots & t_1(d_1, \lambda_B) \\ \vdots & \ddots & \vdots \\ t_M(d_M, \lambda_1) & \dots & t_M(d_M, \lambda_B) \end{bmatrix}. \quad (5.5)$$

The matrix W relates the spectral signal $x_{i,j}$ and its wavelet coefficients $\hat{x}_{i,j}$, as described in Eq. (5.1). The matrix A combines the transmission and wavelet transforms:

$$A = TW. \quad (5.6)$$

y denotes the tensor of measurements of size $X \times Y \times M$. $y_{i,j}$ is the measurement at pixel (i, j) , and it is a vector of length M . The matrix A relates the measurements and the wavelet coefficients of the spectral signal:

$$y_{i,j} = A\hat{x}_{i,j}. \quad (5.7)$$

A more realistic model is to add noise to the measurements:

$$y_{i,j} = A\hat{x}_{i,j} + n_{i,j} \quad (5.8)$$

where n is a tensor of size $X \times Y \times M$. For example, the entries of n may be independent and identically distributed Gaussian random variables.

This model reduces the original hyperspectral data cube of size $X \times Y \times B$ into a smaller data cube of size $X \times Y \times M$. For each pixel, we take M measurements to reconstruct B bands, for a compression ratio of B/M , where $M < B$.

5.2.3. Inverse problem formulation

Our goal is to reconstruct a spectral signal with B bands from M measurements, where $B < M$. To solve this underdetermined system, we constrain the spectral signal to be sparse with a wavelet representation, according to Eq. (5.1). A penalized least-squares formulation matches the measurement $y_{i,j}$ with the model $A\hat{x}_{i,j}$ while minimizing the L_1 norm of the wavelet coefficients $\hat{x}_{i,j}$:

$$\underset{\hat{x}_{i,j}}{\text{minimize}} \quad \frac{1}{2} \|A\hat{x}_{i,j} - y_{i,j}\|_2^2 + \beta \|\hat{x}_{i,j}\|_1. \quad (5.9)$$

To solve for the full hyperspectral image, we sum over all pixels:

$$\underset{\hat{x}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \|A\hat{x}_{i,j} - y_{i,j}\|_2^2 + \beta \|\hat{x}_{i,j}\|_1. \quad (5.10)$$

This optimization can be performed in parallel, where each pixel is treated independently, but it does not account for relations between pixels. In natural images, pixels vary smoothly in space and wavelength. Pixel differences quantify smoothness in each dimension:

$$(\nabla_x x)_{i,j,k} = x_{i+1,j,k} - x_{i,j,k}, \quad (5.11)$$

$$(\nabla_y x)_{i,j,k} = x_{i,j+1,k} - x_{i,j,k}, \quad (5.12)$$

and

$$(\nabla_\lambda x)_{i,j,k} = x_{i,j,k+1} - x_{i,j,k}. \quad (5.13)$$

The total variation term measures the pixel differences over the entire image:

$$\text{TV}(x, \alpha) = \sum_{i=1}^X \sum_{j=1}^Y \sum_{k=1}^B \alpha_x |(\nabla_x x)_{i,j,k}| + \alpha_y |(\nabla_y x)_{i,j,k}| + \alpha_\lambda |(\nabla_\lambda x)_{i,j,k}|, \quad (5.14)$$

where

$$\alpha = [\alpha_x, \alpha_y, \alpha_\lambda]^T. \quad (5.15)$$

We add the total variation term to the penalized least-squares problem:

$$\begin{aligned} &\underset{\hat{x}, x}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \|A\hat{x}_{i,j} - y_{i,j}\|_2^2 + \beta \|\hat{x}_{i,j}\|_1 + \text{TV}(x, \alpha) \\ &\text{subject to} \quad x_{i,j} = W\hat{x}_{i,j}, \quad 1 \leq i \leq X, \quad 1 \leq j \leq Y \end{aligned} \quad (5.16)$$

In our formulation, we minimize the L_1 norm in the wavelet domain and the total variation in the hyperspectral image domain. Since we combine elements of compressive sensing and total variation, we use the abbreviation ‘‘CSTV’’ for our algorithm.

5.2.4. Alternating direction method of multipliers

The alternating direction method of multipliers (ADMM) breaks the objective into more tractable sub-problems. It combines dual decomposition and the augmented Lagrangian [11] to solve equality constraints in convex analysis. To solve (5.16), we write the augmented Lagrangian as

$$\begin{aligned} L_{\eta}(\{\hat{x}_{i,j}\}, \{x_{i,j}\}, \{p_{i,j}\}) = & \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \|A\hat{x}_{i,j} - y_{i,j}\|^2 + \beta \|\hat{x}_{i,j}\|_1 \\ & + \langle p_{i,j}, x_{i,j} - W\hat{x}_{i,j} \rangle + \frac{\eta}{2} \|x_{i,j} - W\hat{x}_{i,j}\|^2 \\ & + \text{TV}(x, \alpha) \end{aligned} \quad (5.17)$$

where $\{p_{i,j} : 1 \leq i \leq X, 1 \leq j \leq Y\}$ are dual variables, $\eta > 0$ is a penalty term, and $\langle \cdot, \cdot \rangle$ denotes an inner product. If a and $b \in \mathbb{R}^N$,

$$\langle a, b \rangle = \sum_{i=1}^N a_i \cdot b_i \quad (5.18)$$

where the subscript i indicates the i th element of a vector. ADMM updates the variables iteratively, in an alternating order:

$$\hat{x}_{i,j}^{n+1} = \underset{\hat{x}_{i,j}}{\text{argmin}} L_{\eta}(\{\hat{x}_{i,j}\}, \{x_{i,j}^n\}, \{p_{i,j}^n\}) \quad \text{for } 1 \leq i \leq X, 1 \leq j \leq Y, \quad (5.19)$$

$$x^{n+1} = \underset{x}{\text{argmin}} L_{\eta}(\{\hat{x}_{i,j}^{n+1}\}, \{x_{i,j}\}, \{p_{i,j}^n\}), \quad (5.20)$$

and

$$p_{i,j}^{n+1} = p_{i,j}^n + \eta(x_{i,j}^{n+1} - W\hat{x}_{i,j}^{n+1}) \quad \text{for } 1 \leq i \leq X, 1 \leq j \leq Y, \quad (5.21)$$

where the superscript n denotes the n th iteration. We can simplify the expressions for the updates:

$$\begin{aligned} \hat{x}_{i,j}^{n+1} = \underset{\hat{x}_{i,j}}{\text{argmin}} & \|A\hat{x}_{i,j} - y_{i,j}\|^2 + \beta \|\hat{x}_{i,j}\|_1 \\ & + \langle p_{i,j}^n, x_{i,j}^n - W\hat{x}_{i,j} \rangle + \frac{\eta}{2} \|x_{i,j}^n - W\hat{x}_{i,j}\|^2 \end{aligned} \quad (5.22)$$

and

$$x^{n+1} = \underset{x}{\text{argmin}} \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \|x_{i,j} - b_{i,j}^n\|^2 + \frac{1}{\eta} \text{TV}(x, \alpha), \quad (5.23)$$

where

$$b_{i,j}^n = -\frac{1}{\eta} p_{i,j}^n + W\hat{x}_{i,j}^{n+1}. \quad (5.24)$$

The updates for $\hat{x}_{i,j}$ and $p_{i,j}$ are independent of other pixels and can be calculated in parallel. These updates can be calculated by convex solvers and proximal algorithms [24, 4]. Section 5.4 derives a simplification of the expression for the update of the hyperspectral image x . Algorithm 1 summarizes the procedure.

Algorithm 1. CSTV: Reconstruction for compressive hyperspectral imaging using total variation minimization

1: **Inputs:**

1. Measured compressed hyperspectral data y
2. Parameters β and η , and total variation weights $\alpha_x, \alpha_y, \alpha_\lambda$
3. Measurement model A

2: Reconstruct each pixel of x independently according to Eqs. (5.1) and (5.10). Set x^0 to the initial reconstruction. Set x^{-1} so that all entries are greater than ϵ compared to the initial reconstruction.

3: Set p^0 to all zeroes.

4: $n \leftarrow 0$

5: **while** $\|x^n - x^{n-1}\|_2 > \epsilon$ **do**

6: Update $\hat{x}_{i,j}^{n+1}$ for each (i, j) according to Eq. (5.22).

7: Update x^{n+1} according to Eq. (5.23).

8: Update $p_{i,j}^{n+1}$ for each (i, j) according to Eq. (5.21).

9: $n \leftarrow n + 1$

10: **end while**

11: $x \leftarrow x^n$

12: **Output:** The hyperspectral image x .

Table 5-2. For each class label, we choose an example pixel location from the full 145×145 image, listed in (row, column) format. Later we plot spectra from each class at these example pixel locations.

Class label	Description	Example pixel location (row, column)
0	Background	(0, 20)
1	Alfalfa	(63, 96)
2	Corn: no till	(16, 11)
3	Corn: min	(0, 0)
4	Corn	(31, 3)
5	Grass/Pasture	(6, 25)
6	Grass/Trees	(42, 26)
7	Grass/Mowed pasture	(72, 108)
8	Hay, windrowed	(32, 126)
9	Oats	(61, 22)
10	Soybeans: no till	(6, 28)
11	Soybeans: min	(0, 97)
12	Soybeans: clean	(1, 28)
13	Wheat	(117, 24)
14	Woods	(8, 120)
15	Building-Grass-Tree-Drives	(0, 71)
16	Stone-steel towers	(12, 46)

5.3. Results

The Indian Pines data set is a hyperspectral image over the Purdue University agronomy farm in West Lafayette, Indiana [5]. The 145×145 image consists of $B = 220$ bands, as measured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), for a data cube of size $145 \times 145 \times 220$. Figure 5-2 displays the Indian Pines scene, labeled by crops or materials. Table 5-2 describes each class label and provides an example pixel location on the 145×145 image of each class. Later we plot example spectra from each class based on these pixel locations.

Figure 5-3 shows transmission plots of the Fabry-Perot resonators at four mirror spacings. The Fabry-Perot mirror reflectivity is $R = 0.8$, and the mirror spacings range from $2 \mu\text{m}$ to $32 \mu\text{m}$. Wavelengths vary from $0.4 \mu\text{m}$ to $2.5 \mu\text{m}$ in the Indian Pines dataset. Figure 5-4 plots the transmission over all mirror spacings and wavelengths simulated in this paper. The matrix T from Eq. (5.5) contains these transmissions. We simulate a measurement on one pixel by computing an inner product between the transmission and spectral signature. The resonator filters incoming light according to the transmission. To produce $M = 160$ measurements per pixel, we vary the mirror spacing for each measurement, which results in different transmission profiles. With $B = 220$, the compression ratio is 1.4. Hence, the measured data cube y has size $145 \times 145 \times 160$. Our goal is to recover the original hyperspectral data cube of size $145 \times 145 \times 220$ from the measured data cube.

To test our algorithm’s robustness to noise, we add noise to each measurement, as described by Eq. (5.8). The noise tensor n has distribution

$$n_{i,j} \sim \mathcal{N}(\mu = 0, \sigma = 3) \quad (5.25)$$

where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and standard deviation σ , and $n_{i,j}$ for $1 \leq i \leq X$ and $1 \leq j \leq Y$ are independent and identically distributed, and they share the same units as the data. Note the hyperspectral data represents radiance values with units of $\frac{W}{\text{cm}^2 \text{ nm sr}}$.

We set the L_1 weight $\beta = 1$ and the penalty parameter $\eta = 10$. For the total variation weights, $\alpha_x = 0.1$, $\alpha_y = 0.1$, $\alpha_\lambda = 3$. The wavelet transform W from Eq. (5.1) represents the Daubechies 8 (db8) wavelet coefficients. We run our proposed Algorithm 1 until it converges, as plotted in Fig. 5-5.

Figure 5-6 compares ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”). We plot spectral signatures from selected pixels for each class from the Indian Pines dataset; Table 5-2 lists the class descriptions and pixel locations of each class plotted here. For the initial reconstruction, we solve a penalized least squares problem as described by Eq. (5.10) for each pixel independently. Our proposed algorithm, labeled as “CSTV,” combines compressive sensing and total variation to produce a more accurate and noise-robust reconstruction, as shown by the magenta line.

Figure 5-10 compares spectral images from ground truth, the reconstruction from the proposed Algorithm 1 (“CSTV”), and the initial reconstruction. We display the 145×145 Indian Pines scene at selected bands from the $0.4 \mu\text{m}$ – $2.5 \mu\text{m}$ range of the dataset. Compared to the initial reconstruction, our proposed algorithm produces more accurate and noise-robust reconstructions.

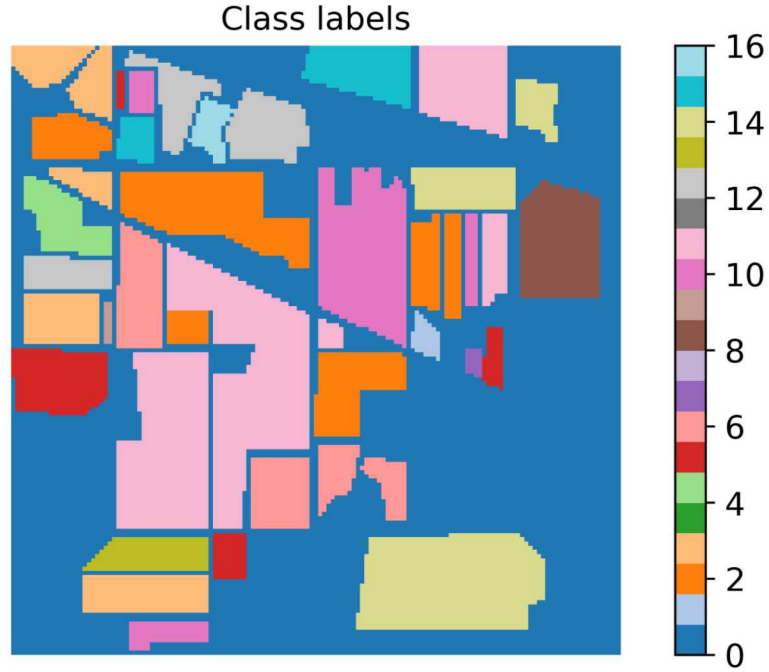


Figure 5-2. The label numbers correspond to different farm crops as described in Table 5-2. Each pixel of the 145×145 scene represents $B = 220$ bands, for a data cube of size $145 \times 145 \times 220$. We will take M measurements per pixel, where $M < B$, and reconstruct the original B bands.

The total variation term constrains each pixel to be similar to its neighbors in space and wavelength, so the algorithm converges to a more physically meaningful solution.

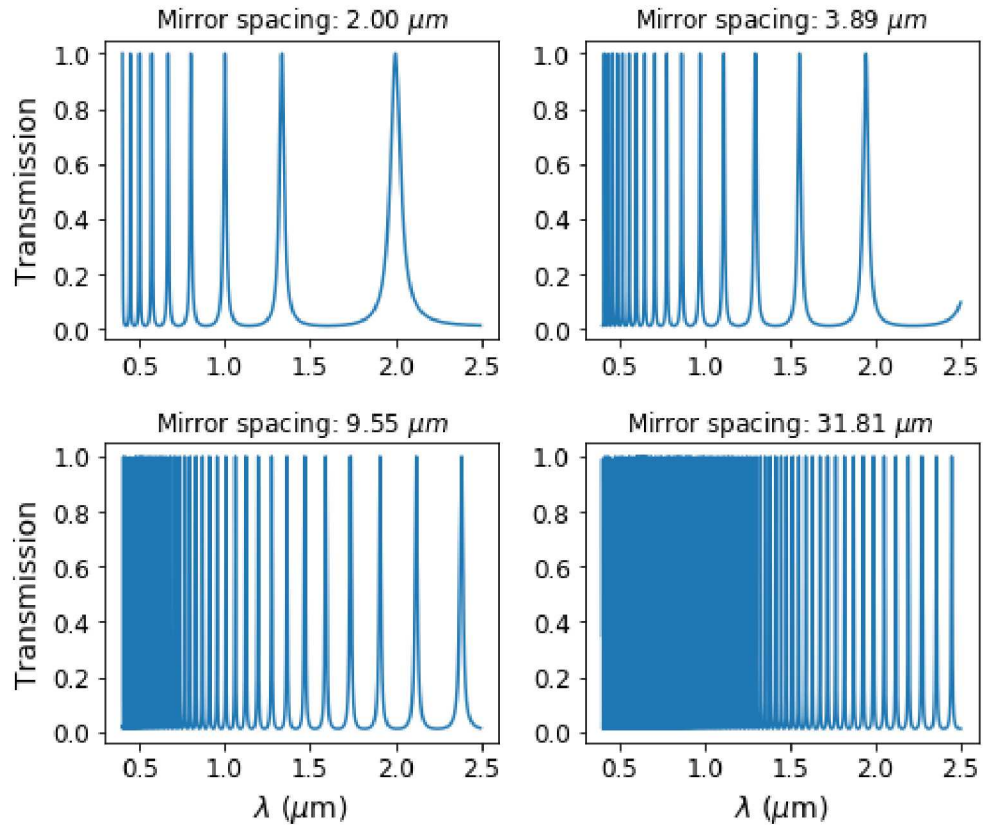


Figure 5-3. The transmission also depends on the mirror spacing of the resonator, and each of the four plots represents a different mirror spacing. We simulate a measurement on one pixel by computing an inner product between the transmission and spectral signature. Each measurement corresponds to a different mirror spacing. The resonator filters incoming light according to the transmission.

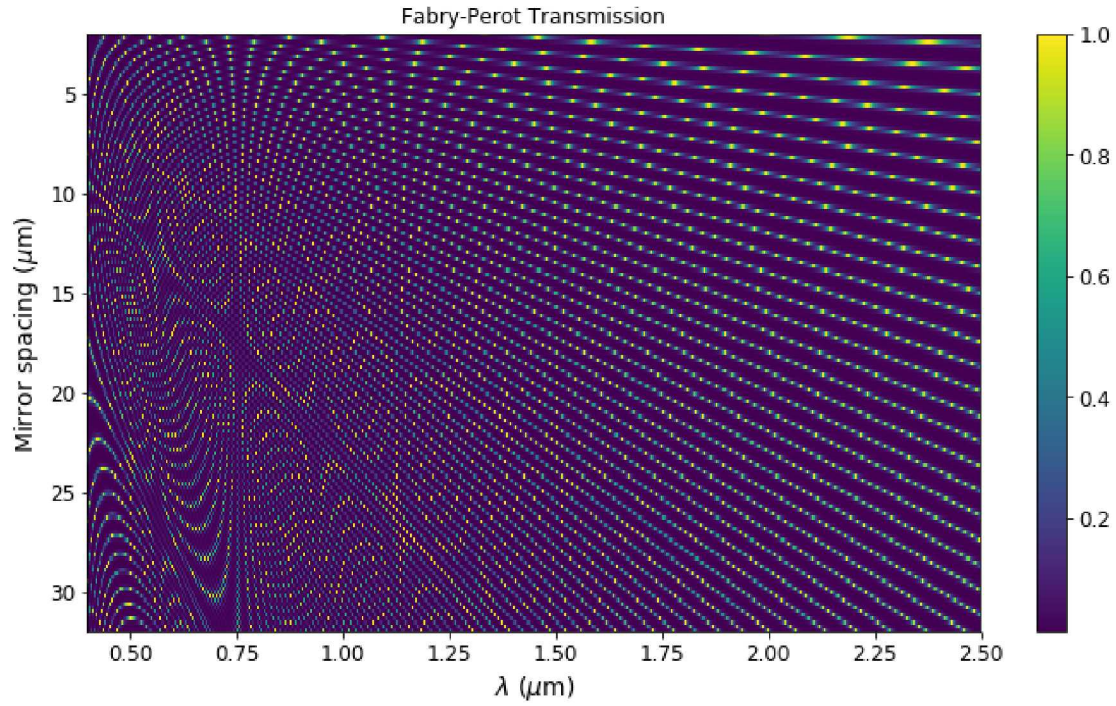


Figure 5-4. In our simulations, the mirror spacing ranges from $2\ \mu\text{m}$ to $32\ \mu\text{m}$. The ground truth for the Indian Pines dataset measures wavelengths from approximately $0.4\ \mu\text{m}$ to $2.5\ \mu\text{m}$, as plotted here.

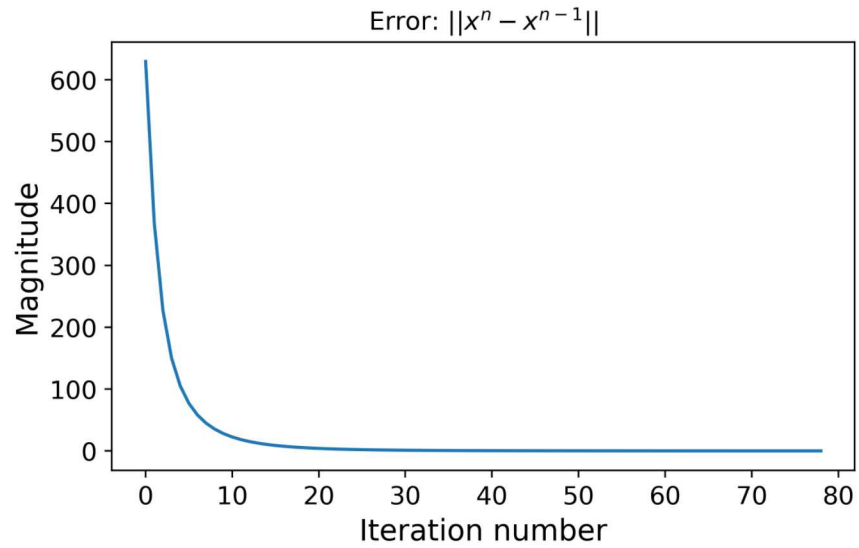
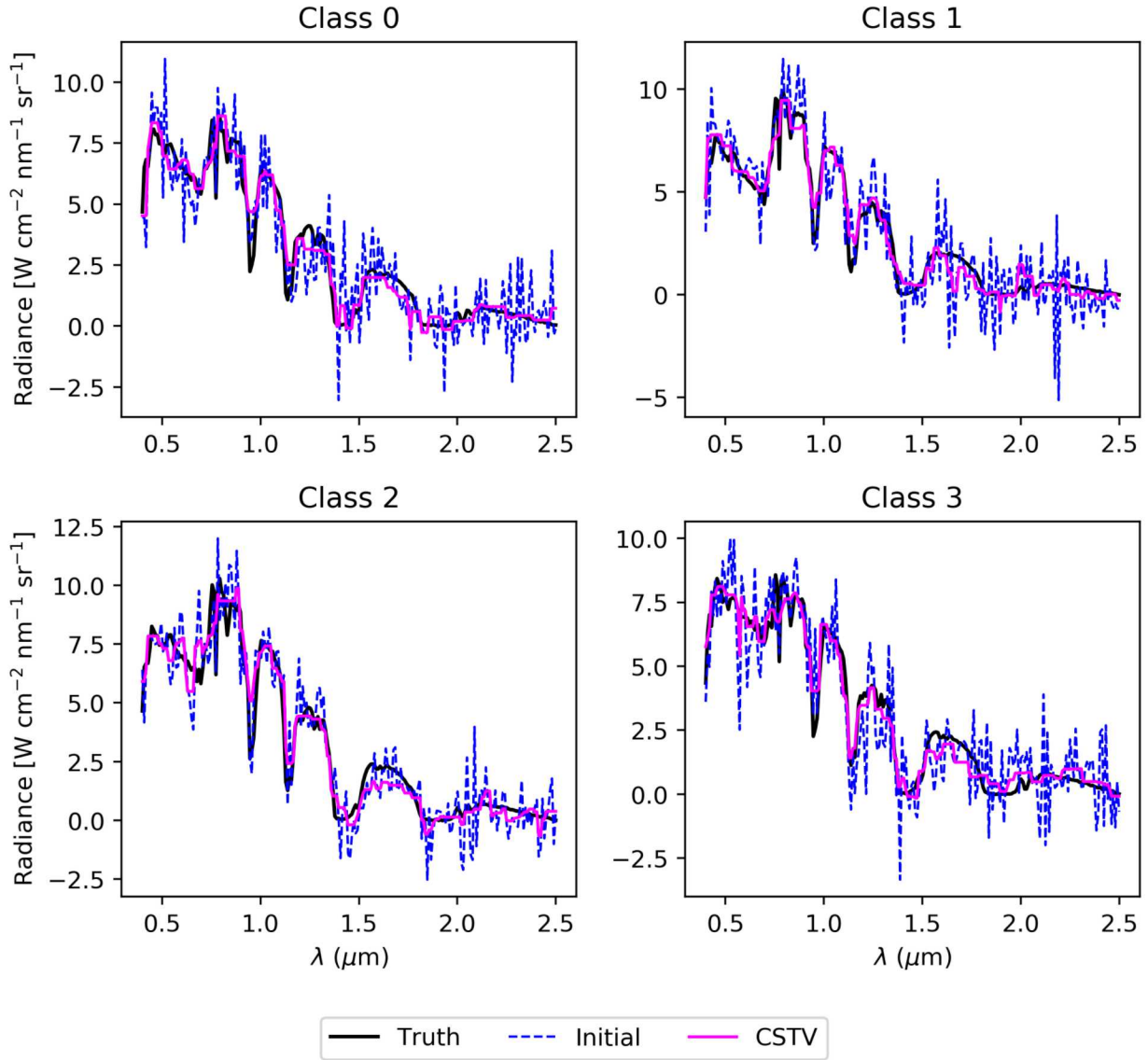
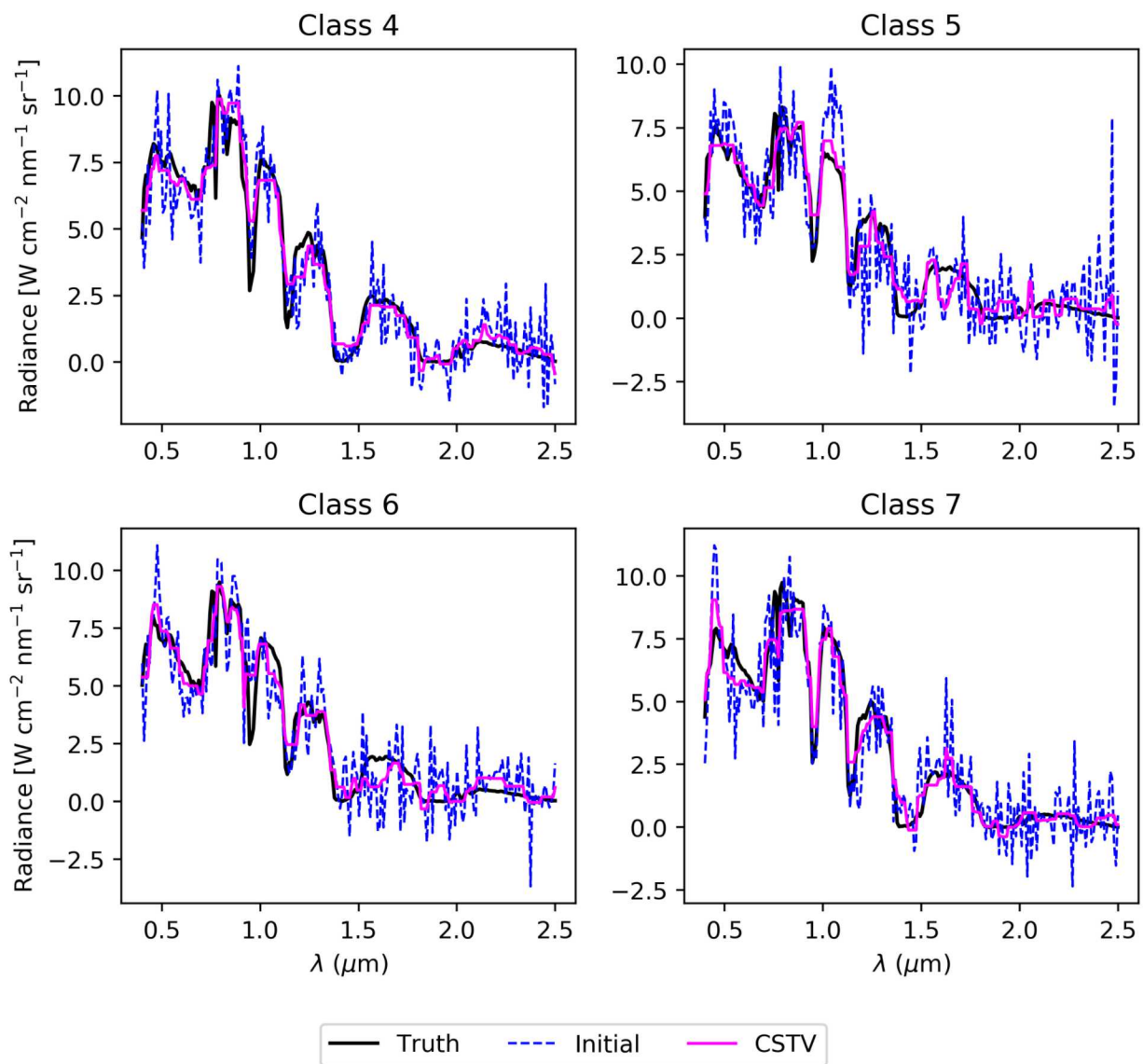


Figure 5-5. Convergence of our proposed Algorithm 1. The reconstructed hyperspectral data cube x converges in less than 80 iterations.



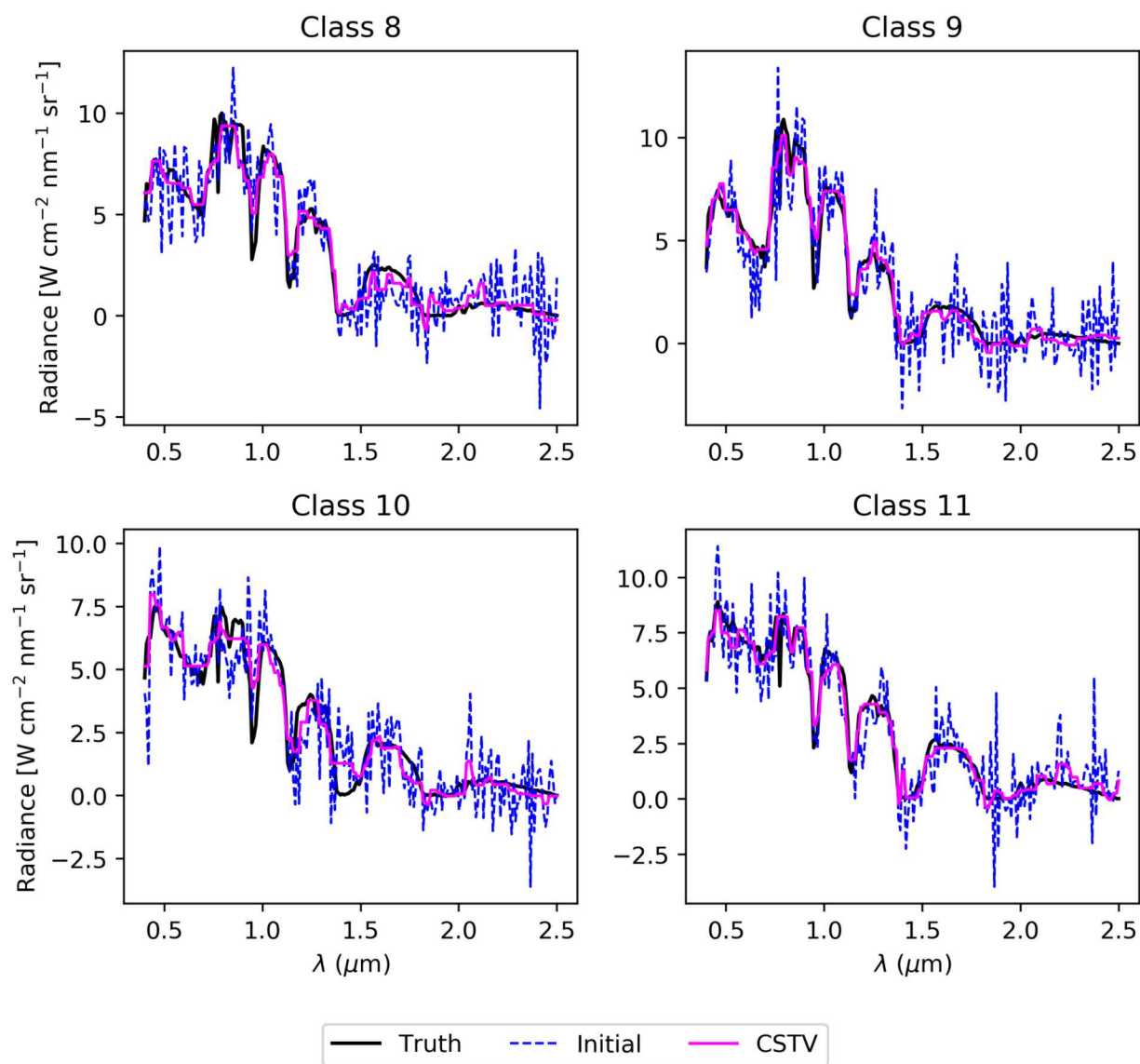
(a) Spectral signatures for classes 0–3.

Figure 5-6. We plot spectral signatures from selected pixels for each class from the Indian Pines dataset; Table 5-2 lists the class descriptions and pixel locations of each class plotted here. Our goal is to reconstruct the hyperspectral data cube of size $145 \times 145 \times 220$ from the measured data cube of size $145 \times 145 \times 160$. For the initial reconstruction, we solve a penalized least squares problem as described by Eq. (5.10) for each pixel independently. Our proposed algorithm, labeled as “CSTV,” combines compressive sensing and total variation to produce a more accurate and noise-robust reconstruction, as shown by the magenta line.



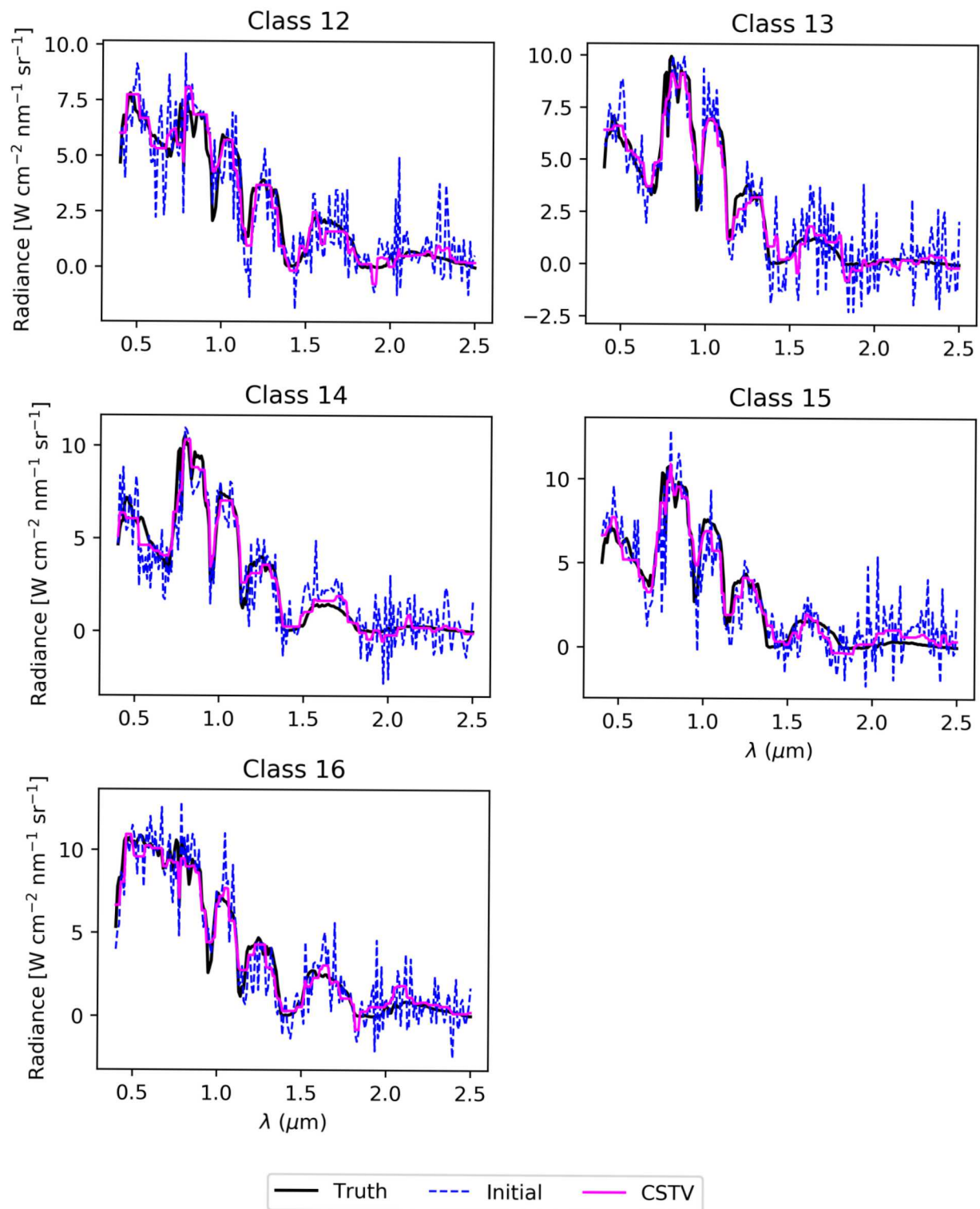
(a) Spectral signatures for classes 4–7.

Figure 5-7. Continued: Comparison of ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”).



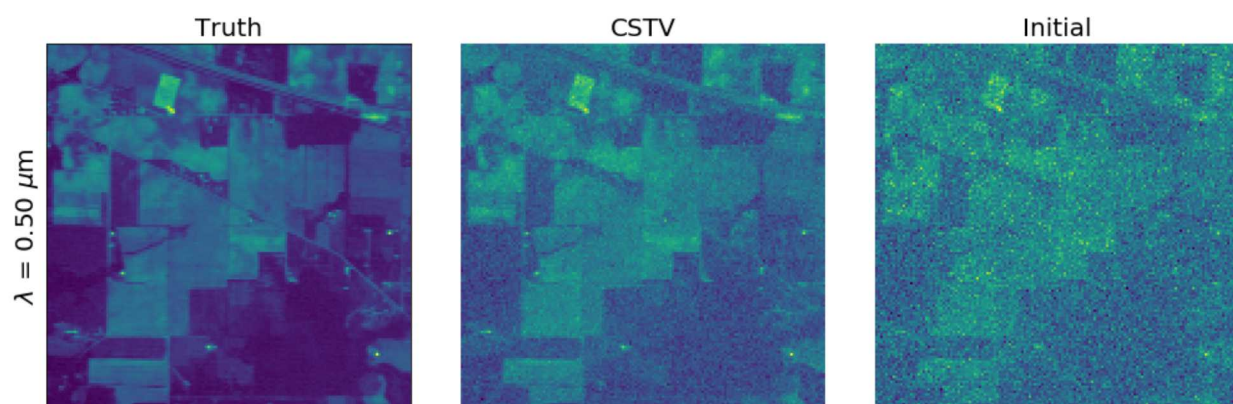
(a) Spectral signatures for classes 8–11.

Figure 5-8. Continued: Comparison of ground truth, the initial reconstruction, and the reconstruction using the proposed Algorithm 1 (“CSTV”).

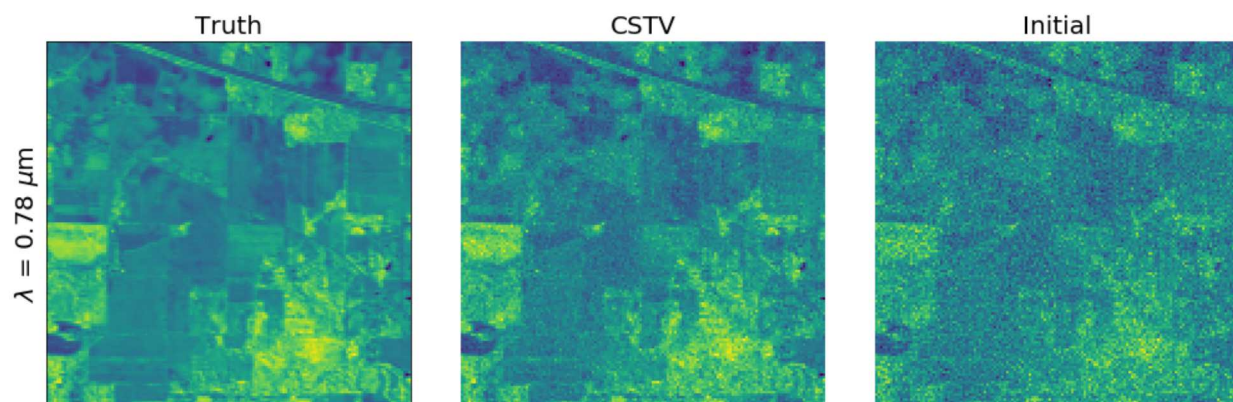


(a) Spectral signatures for classes 12–16.

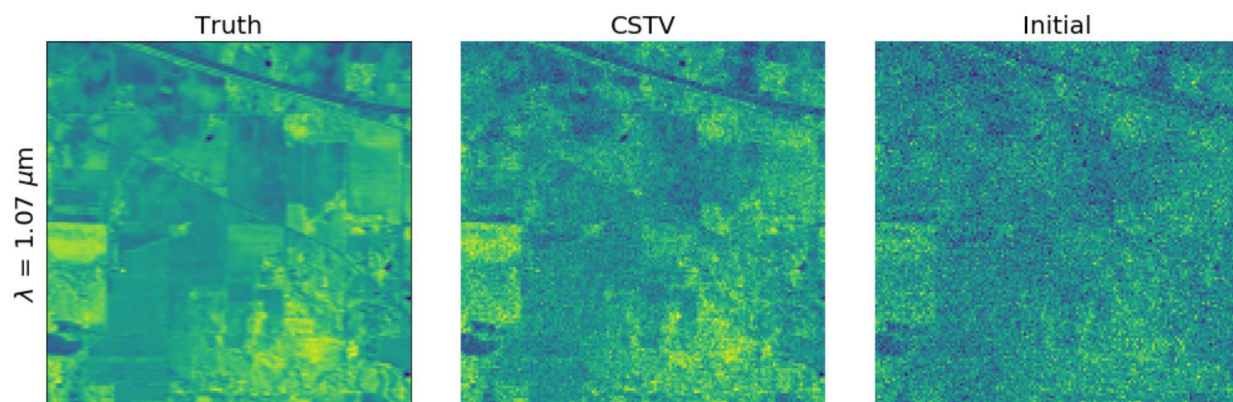
Figure 5-9. Continued: Comparison of ground truth, the initial reconstruction, and the CSTV reconstruction.



(a) Spectral images at $\lambda = 0.50 \mu\text{m}$.

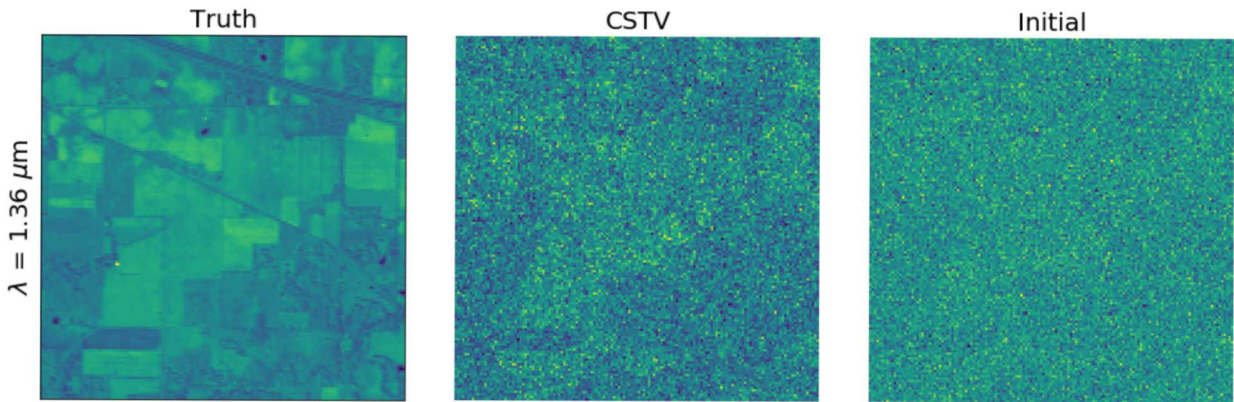


(b) Spectral images at $\lambda = 0.78 \mu\text{m}$.

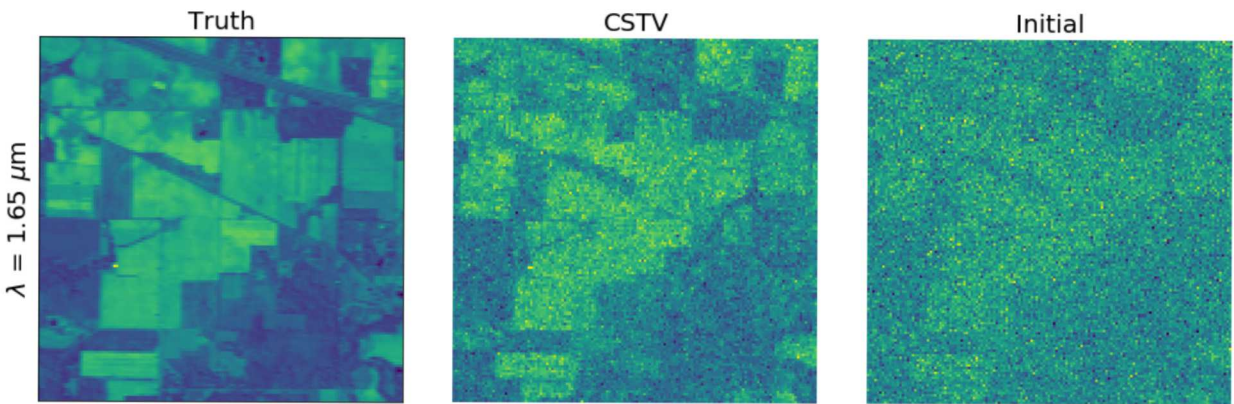


(c) Spectral images at $\lambda = 1.07 \mu\text{m}$.

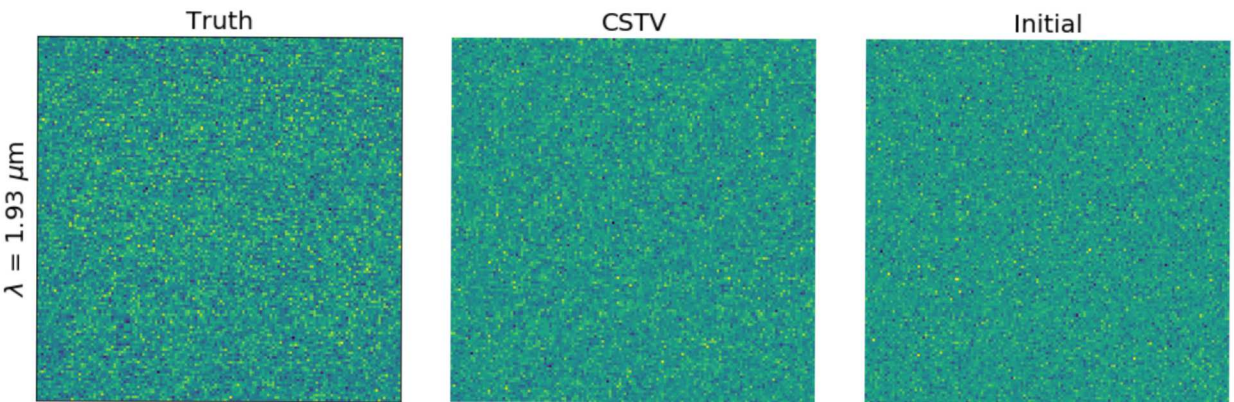
Figure 5-10. Our goal is to reconstruct the hyperspectral data cube of size $145 \times 145 \times 220$ from the measured data cube of size $145 \times 145 \times 160$. The initial reconstruction solves the penalized least squares problem from Eq. (5.10) for each pixel independently. CSTV produces more accurate and noise-robust reconstructions.



(a) Spectral images at $\lambda = 1.36 \mu\text{m}$.

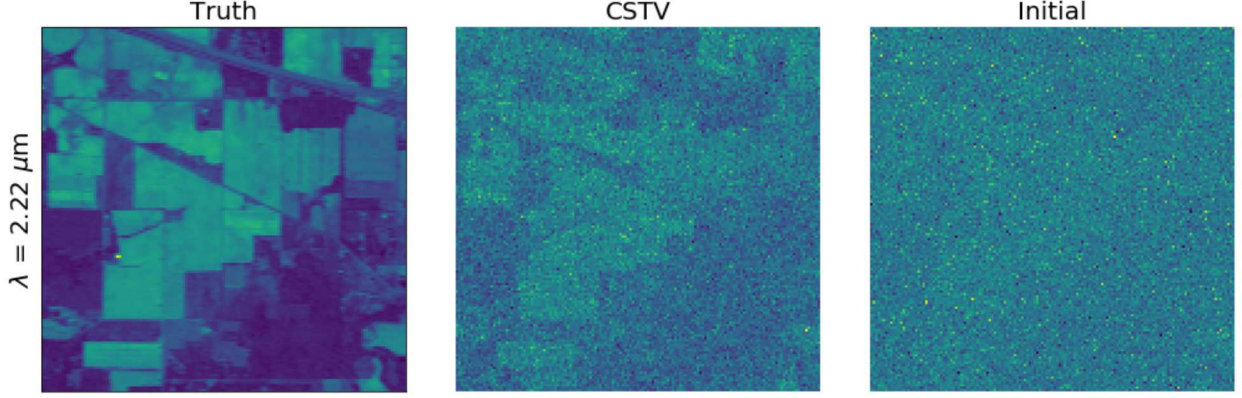


(b) Spectral images at $\lambda = 1.65 \mu\text{m}$.



(c) Spectral images at $\lambda = 1.93 \mu\text{m}$.

Figure 5-10. Continued: Comparison of spectral images from ground truth, the reconstruction from the proposed Algorithm 1 ("CSTV"), and the initial reconstruction.



(d) Spectral images at $\lambda = 2.22 \mu\text{m}$.

Figure 5-10. Continued: Comparison of spectral images from ground truth, the reconstruction from the proposed Algorithm 1 (“CSTV”), and the initial reconstruction.

5.4. Simplifying the update of the hyperspectral image

In this section, we simplify the expression of the update for the hyperspectral image x :

$$\begin{aligned} x^{n+1} &= \underset{x}{\operatorname{argmin}} L_{\eta}(\{\hat{x}_{i,j}^{n+1}\}, \{x_{i,j}\}, \{p_{i,j}^n\}) \\ &= \underset{x}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \langle p_{i,j}^n, x_{i,j} - W\hat{x}_{i,j}^{n+1} \rangle + \frac{\eta}{2} \|x_{i,j} - W\hat{x}_{i,j}^{n+1}\|^2 + \text{TV}(x, \alpha). \end{aligned} \quad (5.26)$$

The inner product term can be further simplified:

$$\langle p_{i,j}^n, x_{i,j} - W\hat{x}_{i,j}^{n+1} \rangle + \frac{\eta}{2} \|x_{i,j} - W\hat{x}_{i,j}^{n+1}\|^2 = \frac{\eta}{2} \|x_{i,j} - b_{i,j}^n\|^2 + c \quad (5.27)$$

where c is a constant, and

$$b_{i,j}^n = -\frac{1}{\eta} p_{i,j}^n + W\hat{x}_{i,j}^{n+1}. \quad (5.28)$$

Hence, the update for x becomes

$$x^{n+1} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^X \sum_{j=1}^Y \|x_{i,j} - b_{i,j}^n\|^2 + \frac{1}{\eta} \text{TV}(x, \alpha). \quad (5.29)$$

5.5. Conclusion

We have presented a reconstruction algorithm for hyperspectral images encoded by spectral modulators. The hyperspectral imager takes fewer measurements than the number of bands in a spectral signal using principles of compressive sensing. To solve this underdetermined problem, we constrain the spectral signal to be sparse in its wavelet coefficients. Our algorithm also

constrains pixels to be similar to their neighbors in space and wavelength, as natural images tend to vary smoothly. It combines L_1 minimization in the wavelet domain to enforce sparsity and total variation in the image domain for smoothness. The alternating direction method of multipliers (ADMM) simplifies the optimization procedure. Our algorithm constrains encoded, compressed hyperspectral images to be smooth in their reconstruction and increases robustness to noise. We have simulated a hyperspectral imager consisting of Fabry-Perot resonators on top of a focal plane array. Our goal is to reconstruct the original hyperspectral data cube of size $145 \times 145 \times 220$ from measurements of size $145 \times 145 \times 160$, for a compression ratio of 1.4 per pixel. As a reference, we reconstruct each pixel independently using a penalized least-squares formulation. The compression ratio and the simulated noise results in a grainy image where the underlying scene can be hard to distinguish. In contrast, our algorithm converges to a more physically meaningful solution because it constrains each pixel to be similar to its neighbors in space and wavelength. This work improves the reconstruction of hyperspectral images from encoded, multiplexed, and sparse measurements.

6. DEEP NEURAL NETWORKS FOR COMPRESSIVE HYPERSPECTRAL IMAGING

6.1. Background: Compressive Hyperspectral Imaging

Compressive hyperspectral imaging reduces measurements

Hyperspectral imagers typically require scanning across a scene to collect spectral measurements. A snapshot hyperspectral imager reduces the number of measurements and avoids the need for spatial scanning. It requires reconstruction algorithms to recover the hyperspectral image from measurements. The reconstruction becomes an underdetermined inverse problem: the number of measurements is less than the number of bands. Regularization techniques such as total variation minimization help to make the problem less ill-posed by adding physical constraints such as smoothness along the spatial and spectral dimensions [49]. However, these compressive sensing algorithms demand heavy computation [49, 48, 41, 47, 44, 46], while greedy algorithms may converge to local minima [50, 45]. In this work, we explore neural networks for two tasks:

1. Reconstruct the hyperspectral image from compressed measurements.
2. Classify the hyperspectral image from compressed measurements.

For Task 2, we compare two inputs for classification: the compressed measurements, and the reconstructed hyperspectral image from the compressed measurements. These tasks address our research questions:

1. Can neural networks reduce computation time for reconstruction?
2. Can neural networks improve classification accuracy for hyperspectral images, either from compressed measurements or reconstructed spectra, compared with traditional classifiers like support vector machines?
3. How does classification performance compare using compressed measurements as input, versus using reconstructed spectra as input?

Filter light by varying spectral transmissions

The proposed compressive hyperspectral imager filters light before the detector array, as shown in Fig. 6-1. Each measurement corresponds to a different spectral transmission on the filter array. Examples of filter arrays include spatial light modulators (liquid crystal displays) and Fabry-Perot resonators, where spectral transmittance varies by applying a digital signal or voltage. This paper will consider Fabry-Perot resonators, in which mirror spacing can vary to control the spectral

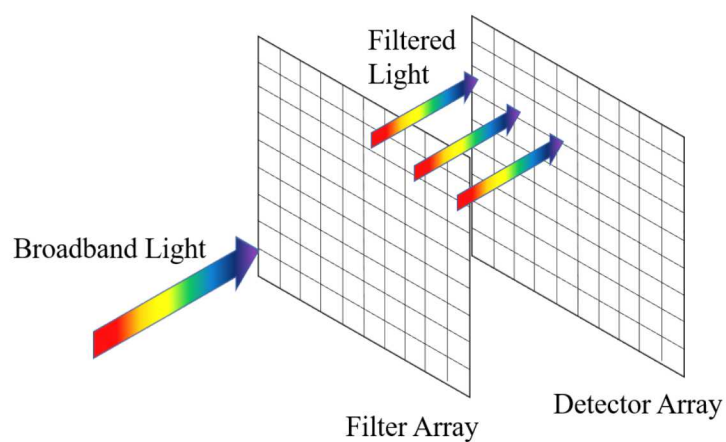


Figure 6-1. The proposed compressive hyperspectral imager filters light before the detector array [49]. Each measurement corresponds to a different spectral transmission on the filter array.

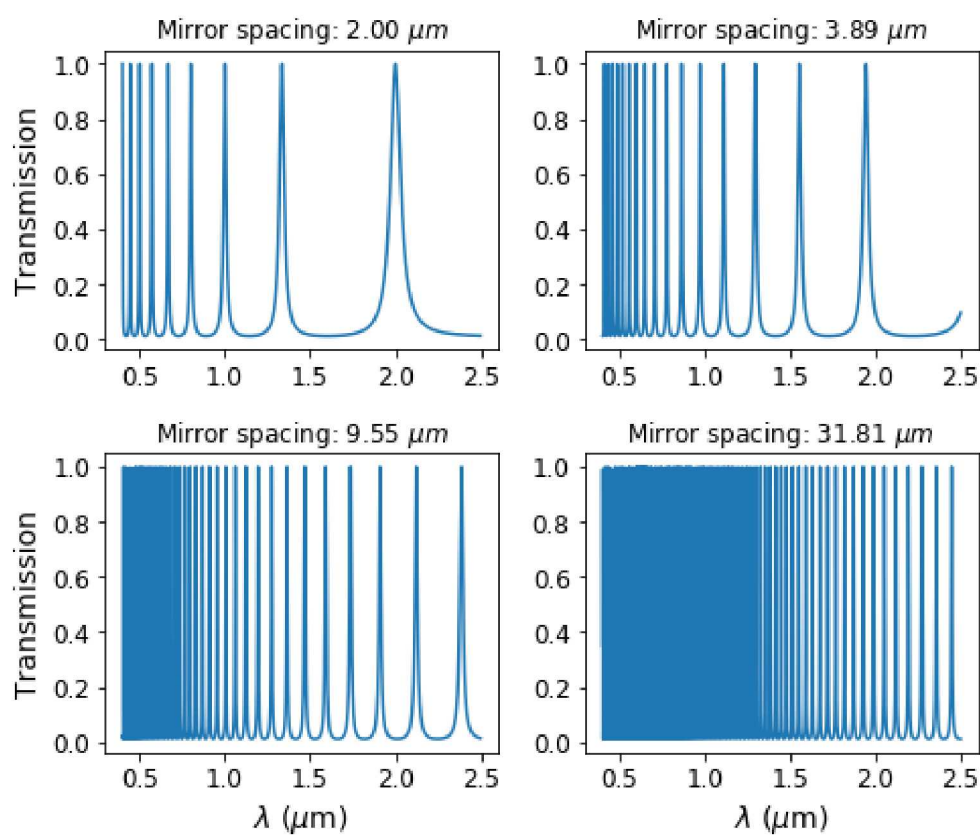


Figure 6-2. Examples of spectral transmissions of a Fabry-Perot resonator. The spectral transmission depends the mirror spacing in the resonator.

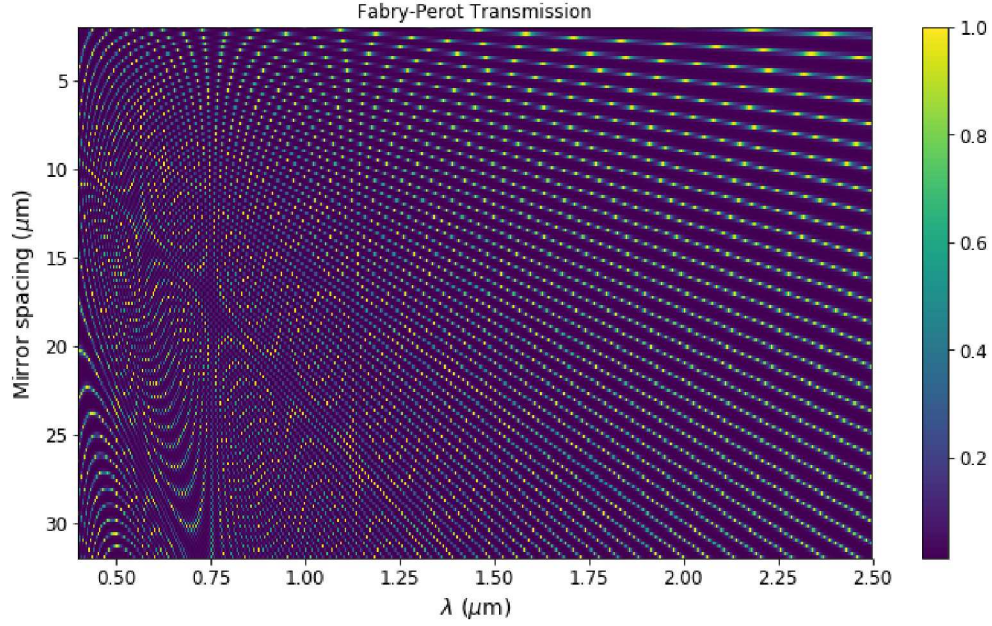


Figure 6-3. Spectral transmission as a function of mirror spacing and wavelength. The number of measurements equals the number of mirror spacings.

transmission. Figure 6-2 shows examples of spectral transmissions at four different mirror spacings. Figure 6-3 displays the spectral transmittance with mirror spacings ranging from $2\ \mu\text{m}$ to $32\ \mu\text{m}$ over wavelengths from $0.45\ \mu\text{m}$ to $2.5\ \mu\text{m}$. *Compressive hyperspectral imagery* is hyperspectral imagery collected from compressive sensing measurements. We refer to Lee [49] for more details on the proposed compressive hyperspectral imager.

6.2. Experiments

6.2.1. Dataset

The Indian Pines dataset is a hyperspectral image of 145×145 pixels with a 20 m spatial resolution and 10 nm spectral resolution over the range of 400–2500 nm, divided into 220 bands [5]. The filter array of the hyperspectral imager modulates all of the 220 bands, including the water absorption region. Each pixel label corresponds to a farm crop as shown in Fig. 6-4. Note that each crop exhibits intraclass variation.

6.2.2. Task 1: Reconstruction of compressive hyperspectral images

This task aims to reconstruct compressive hyperspectral images using neural networks, which will help reduce computation time with fast inference.

Data augmentation with random spectra and additive noise helps to prevent overfitting

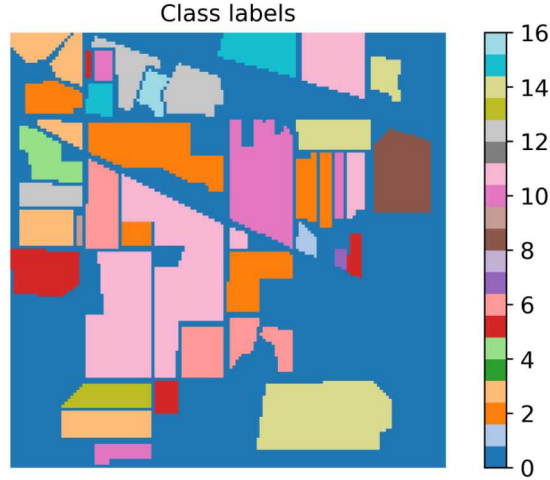


Figure 6-4. Labels of the Indian Pines hyperspectral image. Each label corresponds to a type of farm crop.

Task 1 draws from two datasets:

1. Indian Pines dataset
2. Random spectra dataset

For the second dataset, we generate random spectra with 220 bands from a normal distribution with zero mean and unit variance. Then we apply a Hanning filter with window sizes that vary from 11, 21, 31, and 41. This random dataset contains $145 \times 145 = 21025$ spectra, the same number as the Indian Pines dataset.

For each dataset, we reserve 60% for training, 20% for validation, and 20% for testing, maintaining the class imbalance between each split. Note that each split contains equal amounts of each dataset.

For the training split, we add random noise to the Indian Pines dataset, generated from the same process as the second dataset. The additive noise and the second dataset help to prevent overfitting, so that the neural network does not memorize the training data. We normalize each dataset to zero mean and unit variance.

Multilayer perceptrons reconstruct compressive hyperspectral images

In our experiments, the compressed signal length varies from 160, 80, 40, 20, to 10 measurements. The goal is to reconstruct 220 bands, so the number of measurements is less than the length of the original spectrum.

We investigate multilayer perceptrons to reconstruct compressive hyperspectral images. The input layer maps the compressed input signal to 220 outputs, with ReLU activation. Then K hidden, fully-connected layers follow, each with ReLU activation. The last layer is linear and outputs the reconstructed spectrum. Applications in video compression have utilized multilayer perceptrons for reconstructing compressive sensing measurements [35].

In our experiments, we vary the number of hidden layers: $K = 1, 2, 4, 7, 14$. The multilayer perceptron iterates over each pixel to reconstruct the entire hyperspectral image.

6.2.3. Task 2: Classification of compressive hyperspectral images

Evaluate classifier performance on compressed inputs

This task aims to answer the questions

- Can neural networks improve classification accuracy for hyperspectral images, either from compressed measurements or reconstructed spectra, compared with traditional classifiers like support vector machines?
- How does classification performance compare using compressed measurements as input, versus using reconstructed spectra as input?

We evaluate a variety of classifiers: support vector machines, K nearest neighbors, and three different neural networks, as described below. The input size varies from 160, 80, 40, 20, to 10 measurements, which are less than the full size of 220 bands.

3D convolutions extract spatial and spectral features

We consider two different 3D convolutional neural networks (CNNs). The first 3D CNN extracts blocks of size $5 \times 5 \times I$, where I is the input size, which varies from 220, 160, 80, 40, 20, to 10 in our experiments. Then two convolutional layers follow, with sizes of $3 \times 3 \times 7$ and $3 \times 3 \times 3$, respectively, each followed by ReLU. Next a fully connected layer reads a flattened feature vector and outputs a feature vector with the classification scores for the 16 classes. We train the model with Adagrad and cross-entropy loss. We refer to Li [52] for more details.

The second model is a multiscale 3D CNN. The input patch has size $7 \times 7 \times I$, where I is defined above. It passes through a convolutional layer with $16 \ 3 \times 3 \times 11$ kernels, followed by ReLU. A special layer consists of 16 parallel 3D convolution blocks, each of sizes $1 \times 1 \times 1$, $1 \times 1 \times 3$, $1 \times 1 \times 5$, and $1 \times 1 \times 11$, which are summed and passed through ReLU. A second, identical parallel layer follows. Next is a convolutional layer with 16 kernels of size $2 \times 2 \times 3$, followed by ReLU, $2 \times 2 \times 3$ pooling, and dropout. A fully connected layer outputs a score for each class. We refer to He [33] for more details.

Previous approaches have applied principle components analysis to the spectral dimension, independently of the spatial dimension. Performing convolutions in 3D can simultaneously extract spatial and spectral features.

Recurrent networks characterize spectral correlations

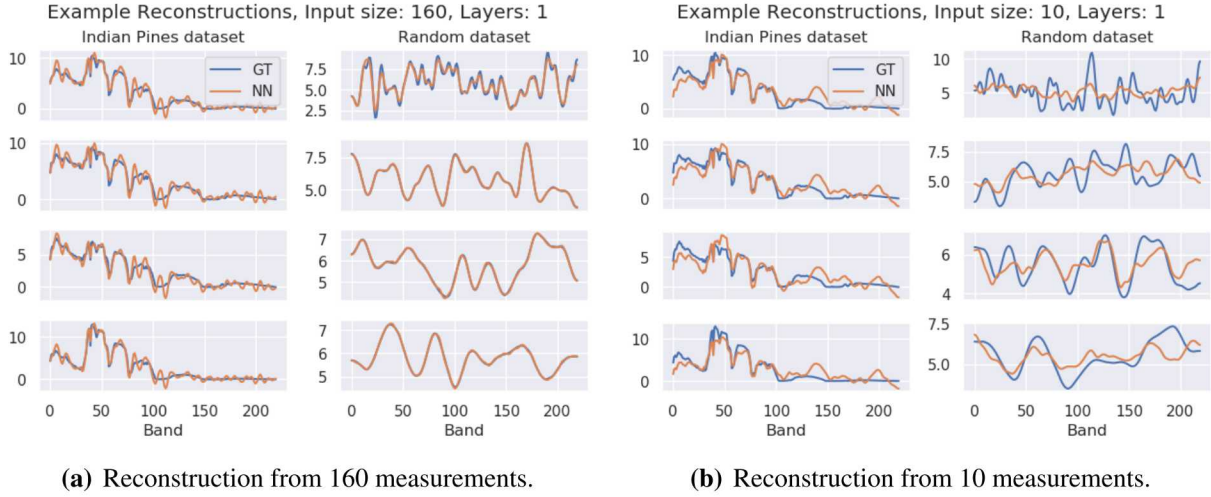


Figure 6-5. Example reconstructions from varying numbers of measurements with a single layer perceptron.

The input to the recurrent neural network (RNN) is a hyperspectral pixel. The recurrent layer reads one band, while the next band is input simultaneously. The RNN predicts the label of the pixel after looping through the entire hyperspectral pixel sequence. We consider a model with gated recurrent units of size 64, followed by batch norm and tanh activation. A final fully connected layer outputs scores for each class. Note that this model considers spectral correlations but not spatial relations between neighboring pixels. We refer to Mou [57] for more details.

6.3. Results

6.3.1. Task 1: Reconstruction of compressive hyperspectral images

Moderate compression (160 / 220) results in small reconstruction error

For Task 1, we study the reconstruction error as the number of measurements decreases from 160, 80, 40, 20, to 10, and the reconstructed spectra has 220 bands. The error also depends on the number of layers in the multilayer perceptron. Figure 6-5(a) shows examples of reconstructions with 160 measurements using a single layer perceptron. Both the Indian Pines and random spectra resemble ground truth closely at a moderate compression ratio (160 measurements of 220 bands).

Larger compression (10 / 220) results in larger reconstruction error

Figure 6-5(b) shows examples of reconstructions with 10 measurements using a single layer perceptron. The error increases in both datasets: the random spectra shows the most error in the higher frequencies, and the Indian Pines dataset shows some bias in bands 0–30 and artifacts near the water absorption region near band 130. As expected, a larger compression (10 measurements of 220 bands) results in larger reconstruction errors.

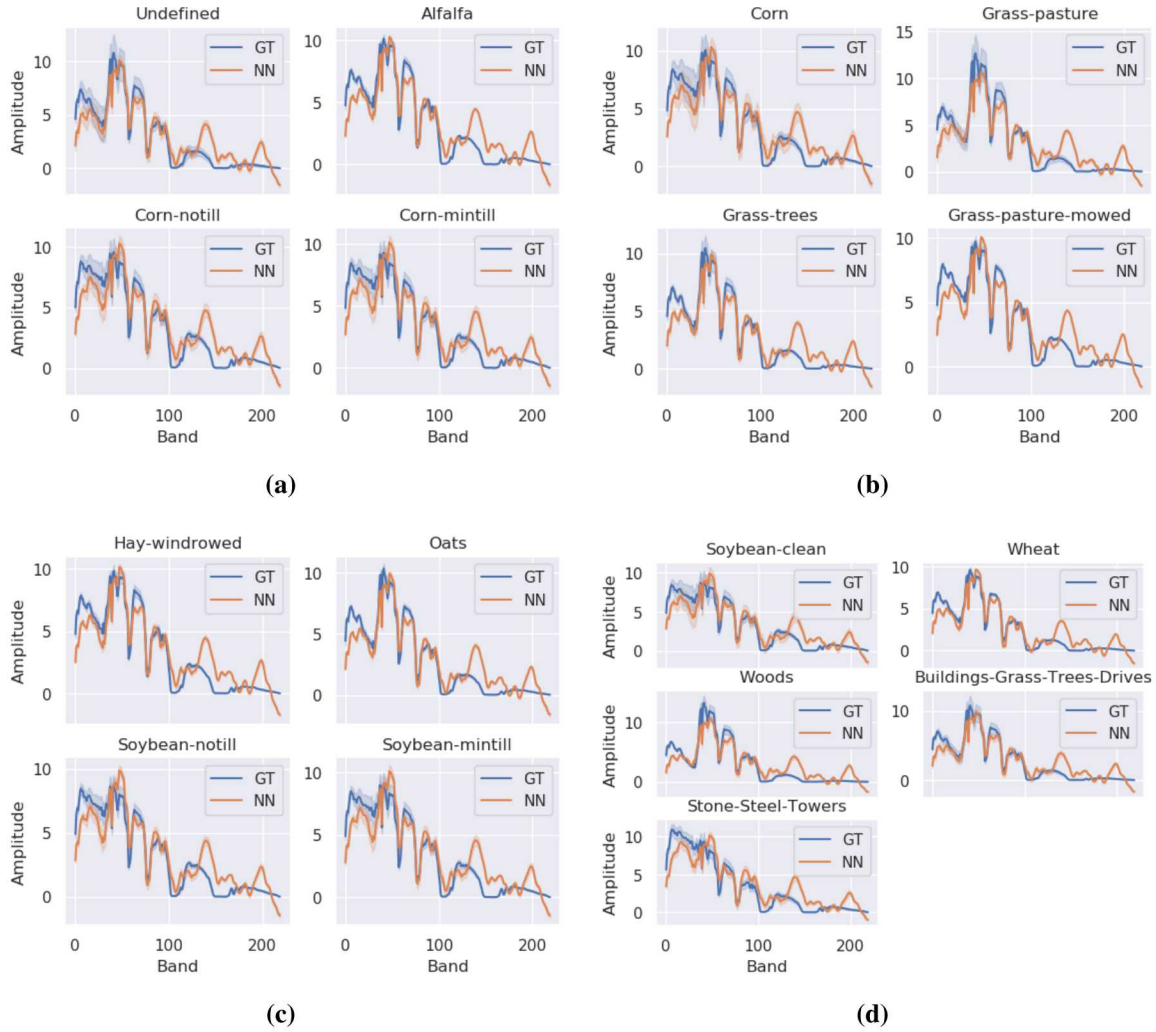
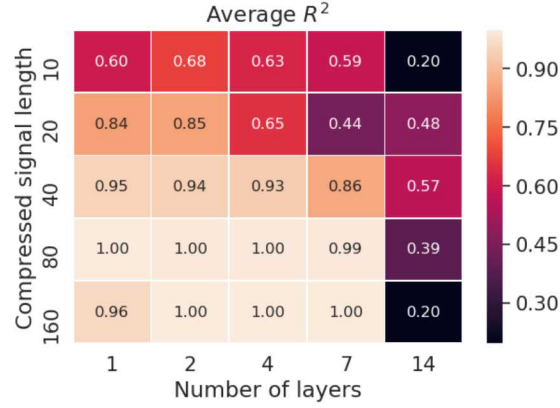


Figure 6-6. Reconstructions of each class/crop from 10 measurements using a single layer perceptron, after running 2000 epochs. The plots show the average spectra with confidence intervals of one standard deviation, which shows intraclass variation. GT: Ground Truth. NN: Single layer perceptron.

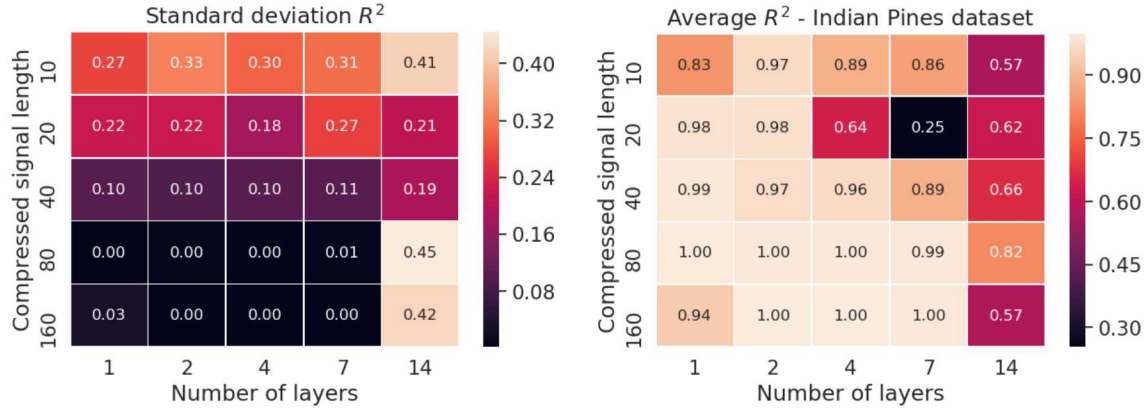
Figure 6-6 shows the reconstruction distribution of the Indian Pines classes from 10 measurements using a single layer perceptron, after running 2000 epochs. The plots show the average spectra of each class with confidence intervals of one standard deviation, which indicates intraclass variation in reconstruction error.

Overall R^2 shows too many layers increases error

Figures 6-7(a) and 6-7(b) show the overall R^2 with standard deviation as a measure of similarity of the reconstruction with the ground truth. The heatmap illustrates the reconstruction quality as the number of measurements and the number of layers vary. Note the training, validation, and testing sets remain fixed, and the standard deviation measures the variation in error over the entire testing set. If the number of layers is too large (e.g. 14 layers), the model begins to memorize the

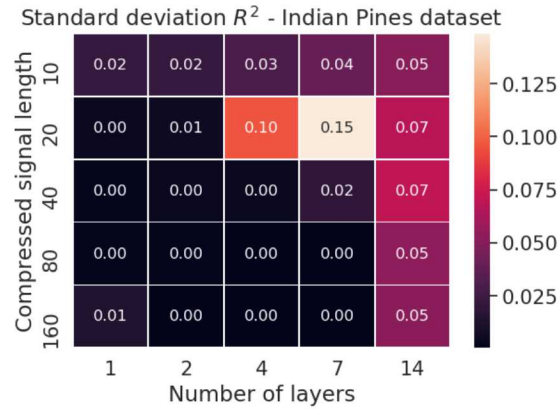


(a) Average R^2 for both datasets (Indian Pines and random spectra).



(b) Standard deviation of R^2 for both datasets.

(c) Average R^2 for the Indian Pines dataset.



(d) Standard deviation of R^2 for the Indian Pines dataset.

Figure 6-7. R^2 for reconstructions over two datasets (Indian Pines and random spectra).

waveforms in the training set and does not generalize well to the testing set. Consequently, the overall R^2 degrades compared to models with fewer layers.

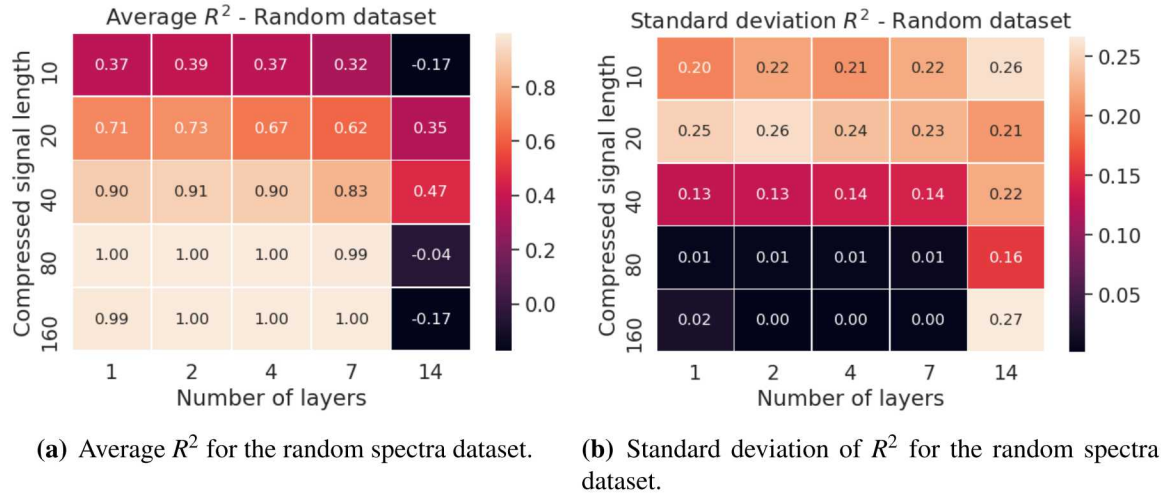


Figure 6-8. R^2 for reconstructions over two datasets (Indian Pines and random spectra).

Single layer shows least overfitting

Figures 6-7(c) and 6-7(d) show R^2 with standard deviation over the Indian Pines dataset, excluding the random spectra dataset. Figures 6-8(a) and 6-8(b) show R^2 with standard deviation over the random spectra dataset, excluding the Indian Pines dataset. A high R^2 value in one dataset may indicate overfitting if the other dataset has a corresponding low R^2 . For example, when the number of layers is 2, 4, and 7, the Indian Pines R^2 is relatively high for 10 measurements, but the corresponding R^2 over the random spectra dataset is much lower. We find that the single layer perceptron shows the least overfitting based on comparing R^2 values between the two datasets.

Regularize the model to further reduce overfitting

Regularization may help to reduce overfitting on the Indian Pines dataset. For example, dropout may be added to the multilayer perceptron. Other random spectra or additive noise may further augment the training dataset. Comparing the reconstruction of the random spectra with the Indian Pines dataset helps to measure how much the model overfits one dataset compared to the other.

6.3.2. Task 2: Classification of compressive hyperspectral images

Recurrent networks perform best on compressed inputs

We split the Indian Pines dataset into training, validation, and testing according to 60/20/20 proportions. All the experiments below use the same split throughout. The standard deviation in the accuracies reflect the variance caused dropout in the neural networks.

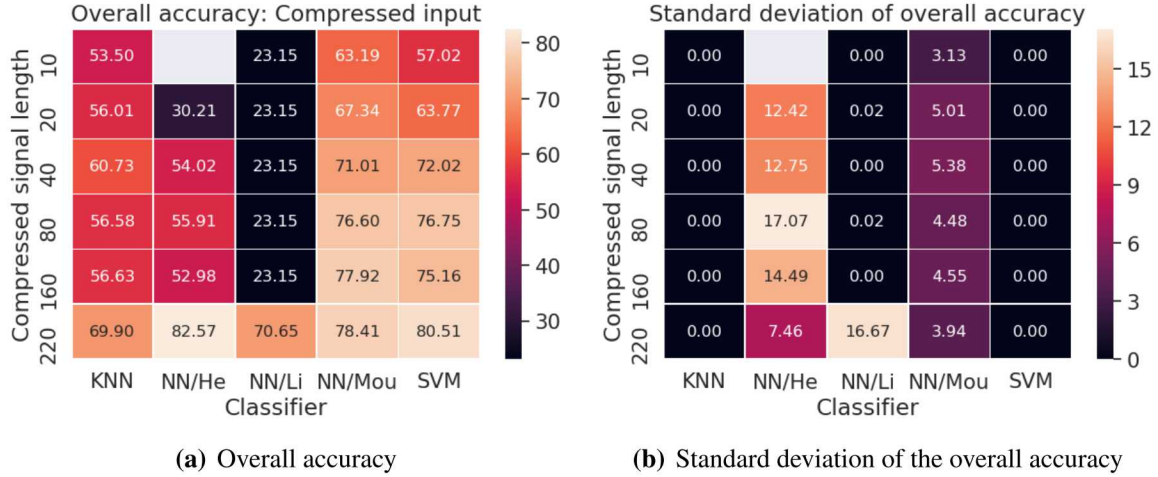


Figure 6-9. Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the compressed (non-reconstructed) measurements. The number of bands in the hyperspectral image is 220, and the compression ratio is the number of measurements divided by the number of bands. Note that the 3D CNN from He does not accept inputs of size 10, so these values are empty. Each model runs 20 times, with 100 epochs per run.

Figure 6-9 shows the overall accuracy with varying input size from 220, 160, 80, 40, 20, to 10 measurements. Note the number of measurements is less than or equal to the number of bands B in the hyperspectral image ($B = 220$ corresponds to the full or uncompressed spectral input). The heatmap compares different classifiers: support vector machines (SVM), K nearest neighbors (KNN), and three types of neural networks as described in Section 6.2.3. For SVM on compressed inputs, we use a radial basis function (RBF) kernel with $C = 1000$ and $\gamma = 0.001$, as determined by grid search. For KNN, we search over $K = 1, 3, 5, 10, 20$. In most cases, $K = 10$ performed best (e.g. for compressed sizes of 10 and 160).

The recurrent neural network (RNN) from Mou [57] performs the best on compressed inputs. Figure 6-10 shows the average class accuracies using the RNN from Mou.

3D CNNs perform best on reconstructed spectra

We test classifier performance on reconstructed inputs as a comparison with the compressed inputs. Figure 6-11 shows the classifiers' accuracies with varying numbers of measurements. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220, and this reconstructed image is the input to the classifiers. We use the single layer perceptron to calculate the reconstructed spectra, since it shows the least overfitting. Each model runs 20 times, with 100 epochs per run.

The multiscale 3D CNN from He performs the best over reconstructed inputs, based on the comparison from Fig. 6-11. Figure 6-12 shows the class accuracies on the reconstructed inputs using the multiscale 3D CNN from He [33]. Note that the values at 220 are empty in the table since the original signal length is 220.

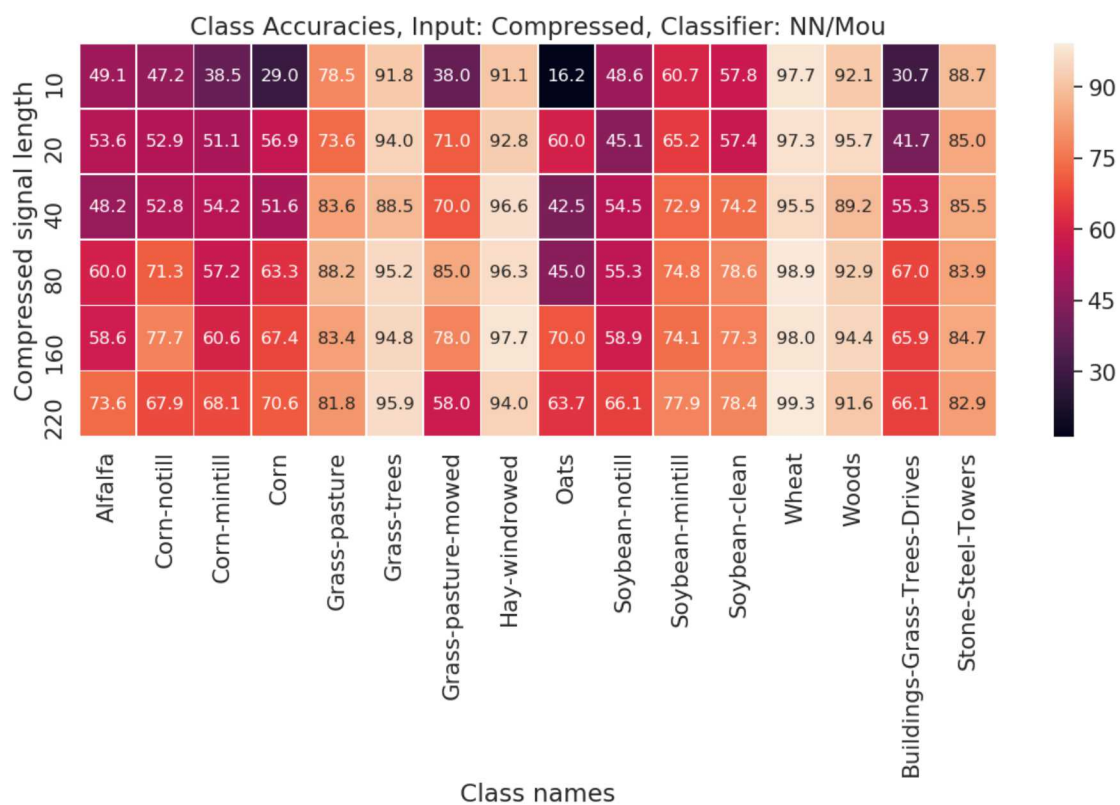
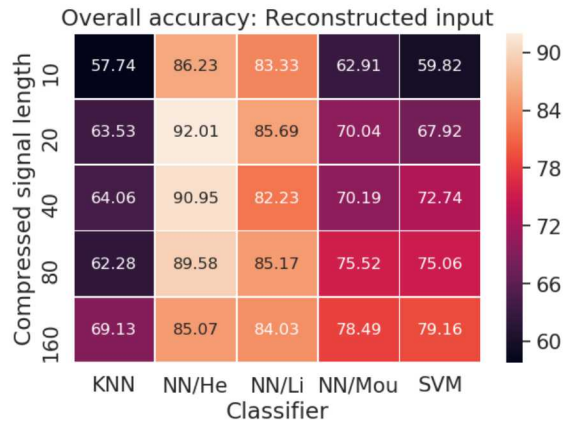
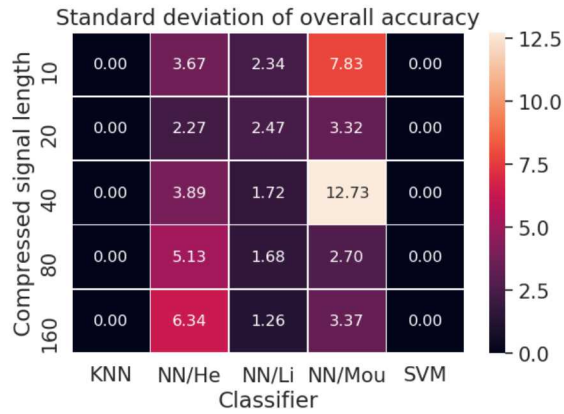


Figure 6-10. Class accuracies with varying numbers of measurements (compressed signal length) using the recurrent neural network from Mou. The inputs to the classifier are the compressed (non-reconstructed) measurements.



(a) Overall accuracy



(b) Standard deviation of the overall accuracy

Figure 6-11. Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the reconstructed hyperspectral images. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. Each model runs 20 times, with 100 epochs per run.

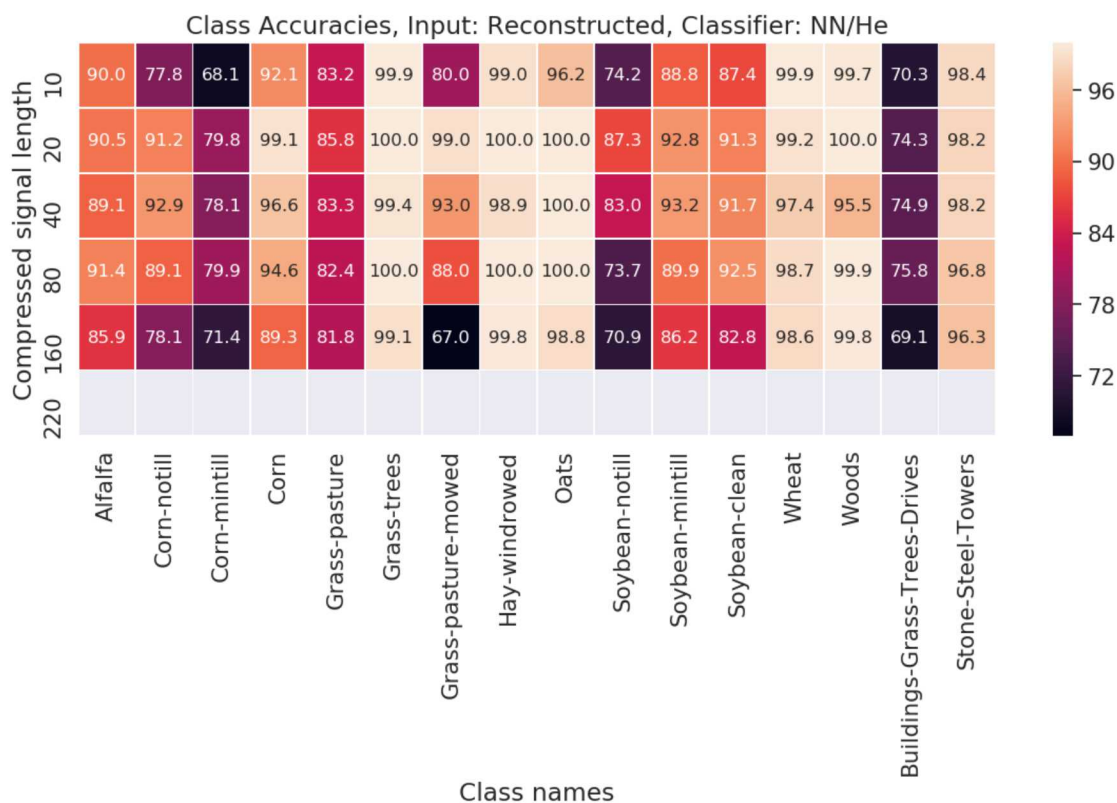


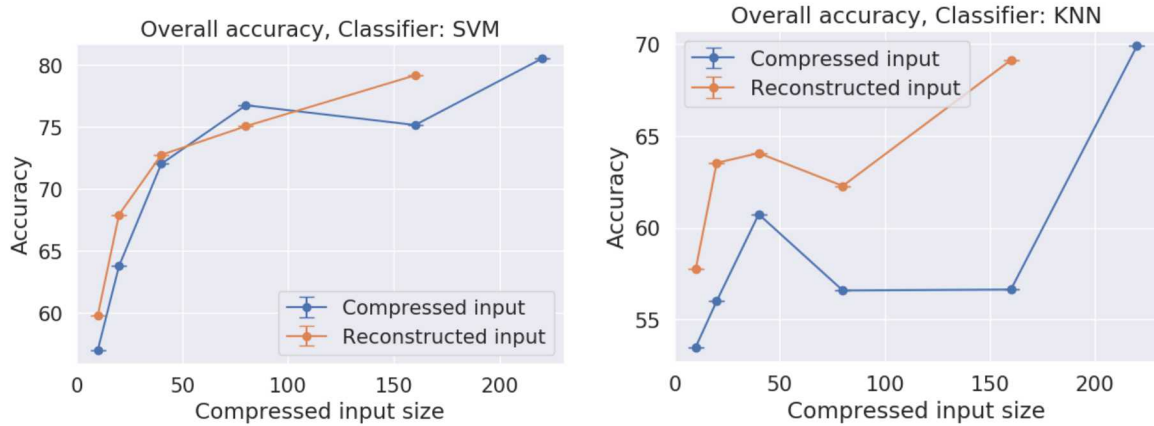
Figure 6-12. Class accuracies with varying numbers of measurements (compressed signal length) using the multiscale 3D convolutional neural network from He. The inputs to the classifiers are the reconstructed hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images, so the values at 220 are empty in the table since the original signal length is 220.

Accuracy improves on reconstructed inputs

Figure 6-13 compares the accuracy of four different classifiers: support vector machines, K nearest neighbors, the RNN from Mou, and the multiscale 3D CNN from He. Each plot compares two inputs to the classifier: the compressed (non-reconstructed) measurements, and the reconstructed spectra. Note that the reconstructed spectra has full size (220 bands), and the x-axis denotes the number of compressed measurements before reconstruction. The size of 220 corresponds to the full spectral input to the classifier. Each model runs 20 times with 100 epochs per run.

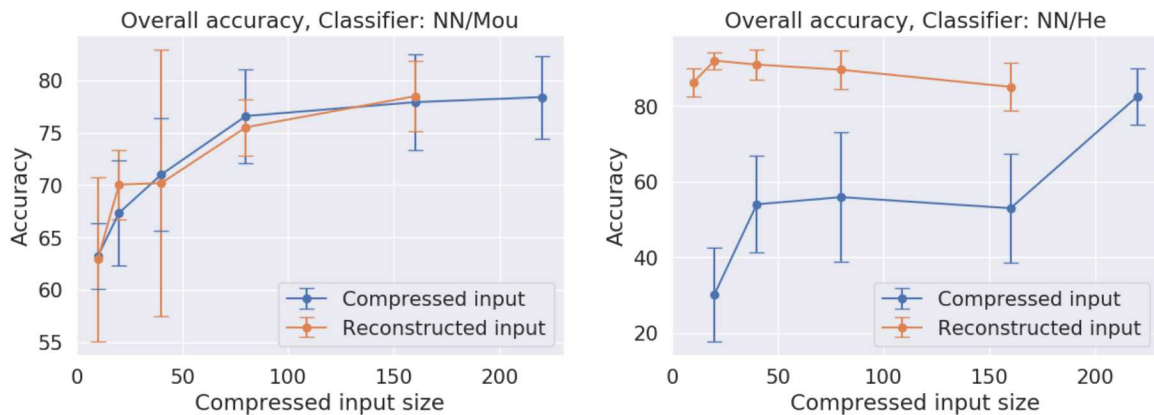
The compressed input may lose spatial context, so 3D convolutions may not be as effective. Neighboring pixels that share the same class may look different in compressed space. Compressive sensing can be an unstable, underdetermined inverse problem, where the number of measurements can be far less than the size of the signal to be reconstructed. As a result, the classifiers that do not account for spatial context perform better on compressed inputs, or the performance is comparable to reconstructed inputs. For example, support vector machines and the RNN from Mou only consider spectral correlations.

Classifiers that consider both spatial and spectral correlations, such as KNN or 3D CNNs, perform better on the reconstructed spectra, which more closely resemble physical signals. These classifiers exhibit a large divergence between reconstructed and compressed inputs, which indicates that the compressed inputs provide poor spatial context for these algorithms. The single layer perceptron from Task 1 produces fairly accurate reconstructions even with large compression ratios, boosting performance on Task 2.



(a) The classifier is the support vector machine. The size K parameters C and γ are determined by grid search for each input size, and the kernel is the radial basis function. The classifier does not consider spectral similarities between adjacent pixels, and both input types seem to perform equally well.

(b) The classifier is K nearest neighbors. The size K parameters C and γ are determined by grid search for each input size, and the kernel is the radial basis function. The classifier does not consider spectral similarities between adjacent pixels, and both input types seem to perform equally well.



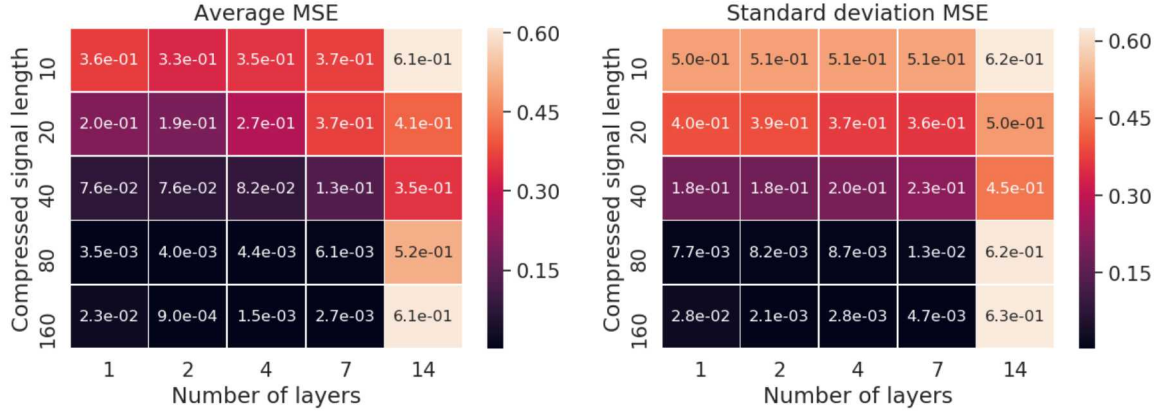
(c) The classifier is the recurrent neural network from Mou. This method predicts spectral correlations but does not consider the spatial dimension. It seems to perform best on compressed measurements.

(d) The classifier is the multiscale 3D convolutional neural network. This method does not seem to perform well on compressed measurements, as adjacent pixels may look dissimilar in compressed space, making convolutional approaches less effective. It seems to perform best on reconstructed spectra.

Figure 6-13. Accuracy comparison of four different classifiers. Each plot compares two inputs to the classifier: the compressed (non-reconstructed) measurements, and the reconstructed spectra. Note that the reconstructed spectra has full size (220 bands), and the x-axis denotes the number of compressed measurements before reconstruction. The size of 220 corresponds to the full spectral input to the classifier. Each model runs 20 times with 100 epochs per run.

6.4. Additional Reconstruction Metrics

We present other metrics, in addition to R^2 , for reconstructing hyperspectral images from compressive sensing measurements. The mean squared error (MSE) is another way to measure the difference between the reconstructed spectra and ground truth. Figure 6-14 shows the average MSE for both datasets, while Fig. 6-15 shows the average MSE for each dataset separately (Indian Pines and random spectra).



(a) Average MSE over both datasets (Indian Pines and random spectra). (b) Standard deviation of the MSE for both datasets.

Figure 6-14. Mean squared error (MSE) in the reconstruction with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.

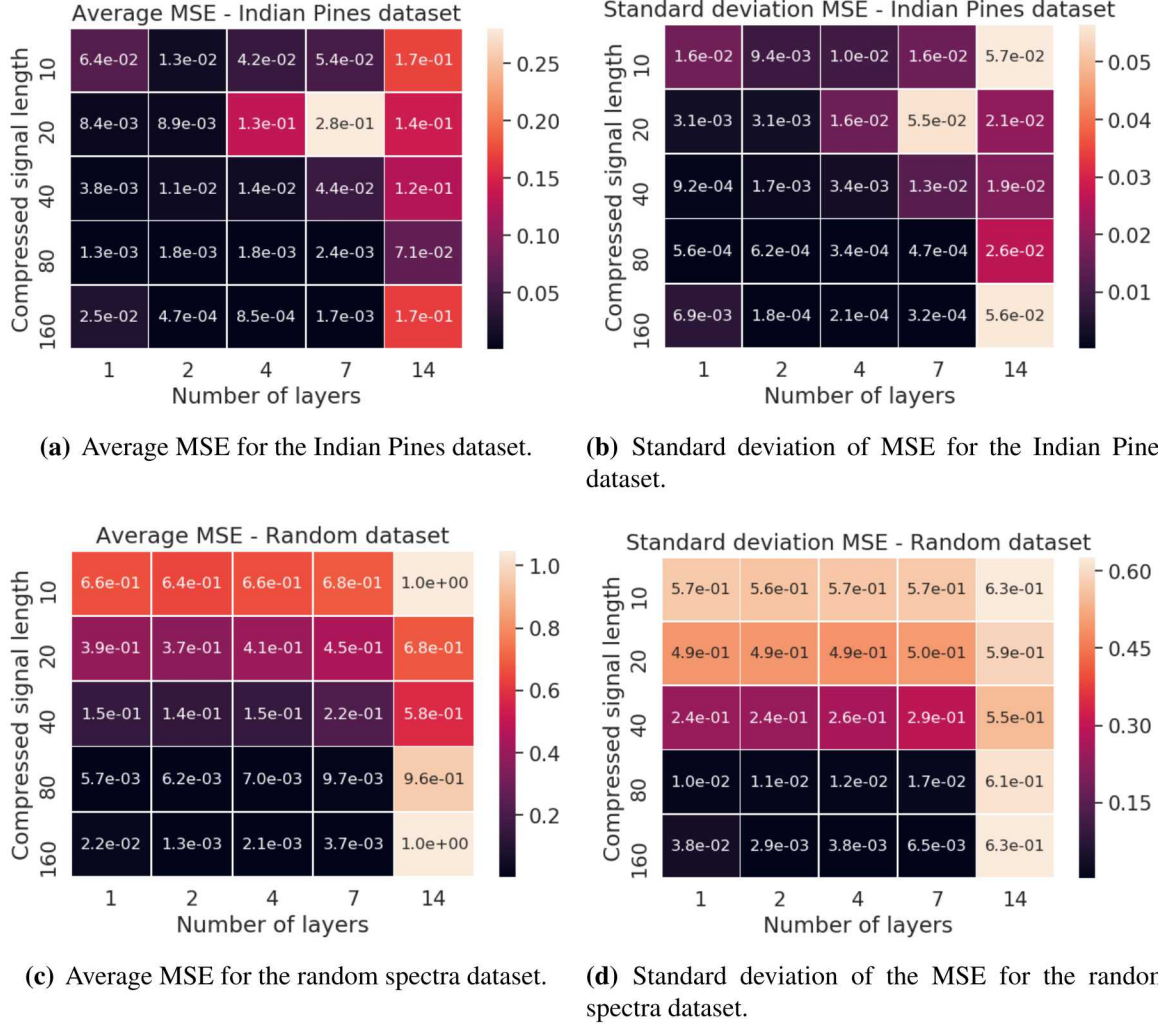
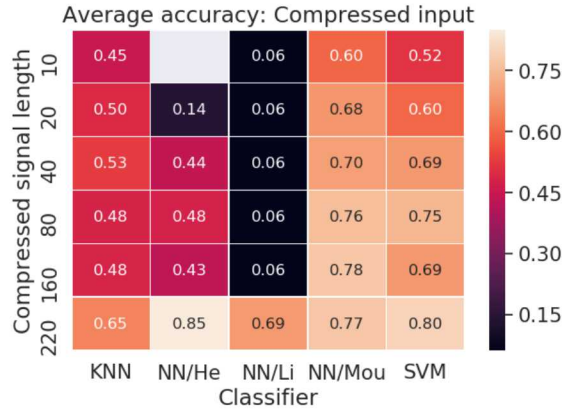


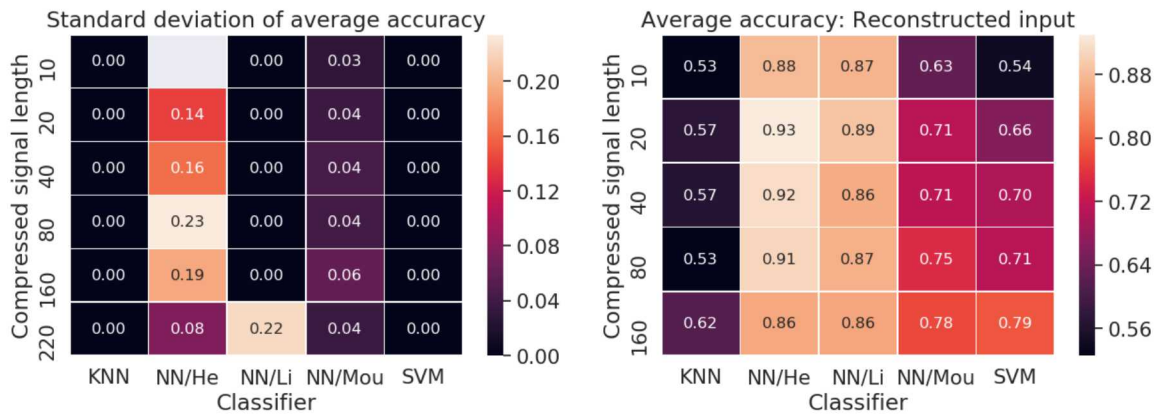
Figure 6-15. Mean squared error (MSE) for each dataset with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.

6.5. Additional Classification Metrics

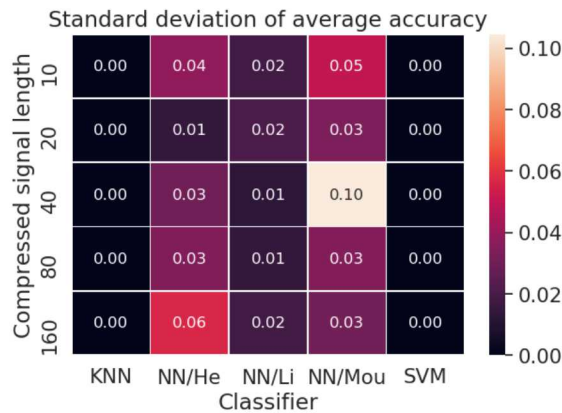
We present other classification metrics, in addition to overall accuracy, for classifying compressive hyperspectral images. The average accuracy is an average of the class accuracies. Figure 6-16 shows the average accuracy from two types of inputs: either compressed measurements or reconstructed spectra. Figure 6-17 shows the average kappa for these two types of inputs. Both figures compare different classifiers over a varying number of measurements.



(a) Average accuracy from classifying compressed measurements.

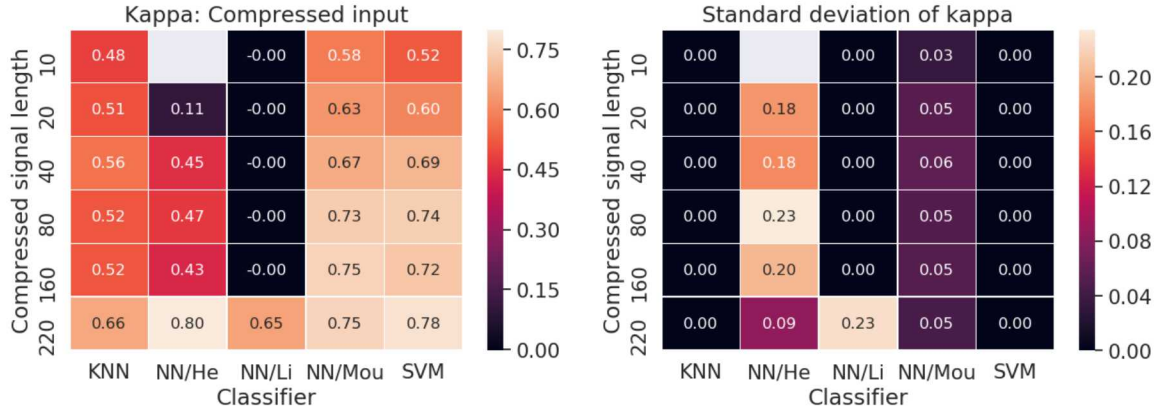


(b) Standard deviation of average accuracy from classifying compressed measurements. (c) Average accuracy from classifying reconstructed spectra.

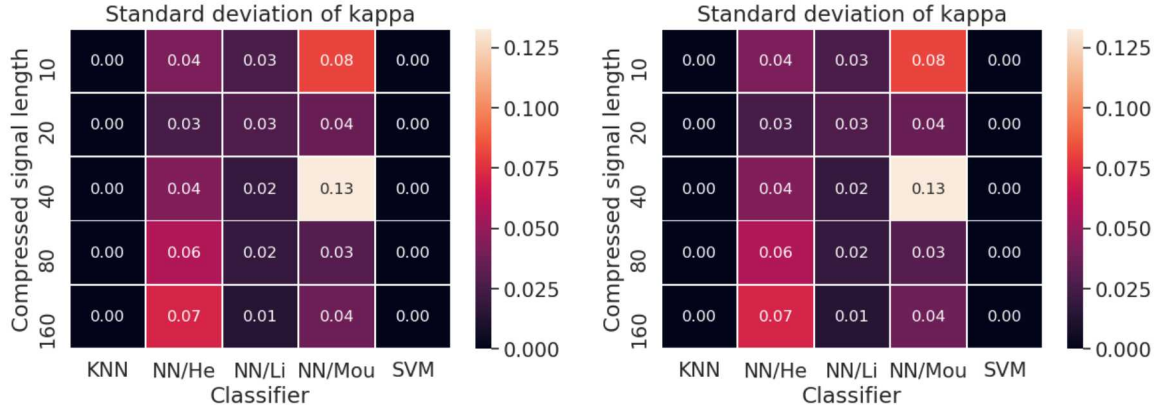


(d) Standard deviation of average accuracy from classifying reconstructed spectra.

Figure 6-16. Average accuracy of classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.



(a) Average kappa from classifying compressed mea- (b) Standard deviation of average kappa from classifying-
surements. ing compressed measurements.



(c) Average kappa from classifying reconstructed spec- (d) Standard deviation of kappa from classifying re-
tra. constructed spectra.

Figure 6-17. Average kappa from classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.

6.6. Conclusion

Neural networks can reconstruct and classify compressive hyperspectral images

We have demonstrated a two step process for hyperspectral image classification using compressive sensing measurements. The first step is to reconstruct the hyperspectral image from compressive sensing measurements. We investigated varying the number of layers in a multiayer perceptron and found that a single layer minimizes overfitting. The second step is to classify the reconstructed image. We compared support vector machines, K nearest neighbors, and three neural networks (3D CNNs, multiscale 3D CNNs, RNNs). Classifier accuracy improves using reconstructed spectra compared to raw compressed measurements. This work shows how neural networks can reconstruct and classify compressive hyperspectral images, which reduces the number of measurements and speeds acquisition time.

7. MATRIX COMPLETION FOR COMPRESSIVE SENSING USING CONSENSUS EQUILIBRIUM

7.1. Introduction

Cameras are now able to capture data exceeding > 1 gigapixel, $> 10,000$ frames per second, and > 100 color channels. Data rates can potentially reach 1×10^{15} pixels per second. Such high data rates make real-time processing difficult.

Efficient sensing takes advantage of low-dimensional structure in signals. Two areas of research explore prior structured knowledge: compressive sensing and matrix completion. Compressive sensing permits few linear measurements of a signal with nearly exact reconstruction. Matrix completion aims to estimate incomplete observations living in a low rank subspace.

We propose to recover an uncompressed tensor from incomplete compressive measurements by combining these techniques [43]. Consensus equilibrium breaks the reconstruction problem into subproblems to solve for the high-dimensional tensor. This framework allows us to apply two constraints on the statistical inversion problem. First, matrix completion enforces a low rank constraint on the compressed data. Second, the compressed tensor should be consistent with the uncompressed tensor when it is projected into the lower dimensional space. We validate our method on the Indian Pines hyperspectral dataset with varying amounts of missing data. This work opens up new possibilities for data reduction, compression, and reconstruction. This section introduces some key concepts used in this paper.

Compressive sensing exploits the inherent structure and redundancy within an acquired signal. Image and video compression algorithms operating on commercial 10 megapixel cameras can achieve compression ratios of 100:1 or higher for visualization or classification tasks, illustrating the redundancy in natural images.

Suppose a vector has N components, and $S < N$ components are non-zero. We wish to take M measurements using a random sensing strategy. Compressive sensing theory says that if M is approximately greater than $S \log N$, we can recover the N components with high probability [66]. The signal may be S -sparse under a transform, such as the Discrete Cosine Transform or wavelet transform [47]. Applications include digital holography [44], video coding [53], optical polarimetry [41, 48], image processing [28], and communications.

A key notion in compressive sensing is the restricted isometry property (RIP) [16]. If a matrix A satisfies the RIP, then A approximately preserves the Euclidean length of S -sparse signals. This implies that all pairwise distances between S -sparse signals must be well preserved in measurement space. In other words, the distance between two low-dimensional subspaces remain almost unchanged after projection by a random matrix with overwhelming probability. We

hypothesize that low-rank matrices remain low-rank after projection. Under this hypothesis, we can perform matrix completion on the compressive measurements.

Matrix completion finds applications in signal processing, computer vision, and control theory. It aims to generate a completed matrix from missing entries. Observed entries may be corrupted by erasures and transmission errors, scene occlusions, deliberate omissions, or bad pixels [64].

Candes and Tao provide a theoretical guarantee for exact matrix retrieval under nuclear norm convex relaxation [17]. Other formulations factor a matrix $A = UV$ with gradient updates for U and V [36]. In this work, we are not directly interested in recovering the missing entries from the compressive measurements. Rather, the end goal is to reconstruct the uncompressed tensor from incomplete compressive measurements.

Statistical inversion techniques minimize an objective involving data fidelity and regularization terms using the maximum a posteriori estimate [46]. Efficient minimization methods such as alternating direction method of multipliers (ADMM) can apply different proximal maps in sequence [10, 49]. However, many inverse problems cannot be framed in terms of an explicit objective [12, 42, 50, 45]. Consensus equilibrium is an optimization-free generalization of regularized inversion that can fuse multiple sources of information [13]. Applications include tomography and denoising [68]. Here we demonstrate a novel problem of reconstruction from incomplete compressive measurements.

7.2. Theory

7.2.1. Notation

Throughout this paper, we follow standard tensor notation from Kolda [38]. Let $\mathcal{X} \in \mathbb{R}^{M \times N \times B}$ denote a three-way tensor with dimensions $M \times N \times B$. The mode-3 unfolding of this tensor is

$$X := X_{(3)} \in \mathbb{R}^{B \times (M \times N)}. \quad (7.1)$$

Let $A \in \mathbb{R}^{R \times B}$ denote a matrix that encodes the measurement of $\mathcal{X}$. Here B represents the signal length, while R is the number of measurements, with $R < B$. Denote $\mathcal{Y} \in \mathbb{R}^{M \times N \times R}$ as the tensor of measurements. It is a compressed version of $\mathcal{X}$:

$$\mathcal{Y} = \mathcal{X} \times_3 A \quad (7.2)$$

where $\times_3$ is a tensor 3-mode product. Define Y as the mode-3 unfolding of this tensor:

$$Y := Y_{(3)} \in \mathbb{R}^{R \times (M \times N)}. \quad (7.3)$$

Here the third dimension of $\mathcal{X}$ is compressed from size B to size R . The compression ratio is

$$\kappa = \frac{R}{B}. \quad (7.4)$$

We index the complete set of measurements as

$$\Omega = \{1, 2, \dots, M \times N \times R\}. \quad (7.5)$$

Define ω as the set of observations, which may not include all of the possible measurements:

$$\omega \subset \Omega = \{1, 2, \dots, M \times N \times R\}. \quad (7.6)$$

If entry (i, j, k) is unobserved, we use the shorthand that $(i, j, k) \notin \omega$. The fraction of missing entries is

$$\mu = 1 - \frac{|\omega|}{M \times N \times R}. \quad (7.7)$$

Notice that the original tensor $\mathcal{X}$ is reduced in the number of entries through two steps:

1. Compression by A to yield $\mathcal{Y}$, and
2. Projection onto a set of observations to yield $\mathcal{P}_\omega(\mathcal{Y}) \in \mathbb{R}^{M \times N \times R}$, where the sampling operator $\mathcal{P}_\omega(\cdot)$ returns the original tensor but with the (i, j, k) entry denoted as missing if $(i, j, k) \notin \omega$.

After these steps, the total number of measurements τ becomes

$$\tau = \kappa(1 - \mu). \quad (7.8)$$

7.2.2. Compressive sensing

The goal is to recover $\mathcal{X}$ from $\mathcal{P}_\omega(\mathcal{Y})$. We consider the subproblems of reconstructing the mode-3 fibers $x_i \in \mathbb{R}^B$ for $i = 1, 2, \dots, M \times N$. Let $W \in \mathbb{R}^{B \times B}$ denote the inverse of a transform which represents a signal in terms of sparse coefficients. It may represent wavelet or Fourier transforms, for example. Write the coefficients as $\tilde{x}_i \in \mathbb{R}^B$ for $i = 1, 2, \dots, M \times N$. The corresponding measurements are the mode-3 fibers of $\mathcal{Y}$: $y_i \in \mathbb{R}^R$ for $i = 1, 2, \dots, M \times N$. A solution to this problem is provided by basis pursuit with denoising parameter σ :

$$\begin{aligned} & \underset{\tilde{x}_i}{\text{minimize}} \quad \|\tilde{x}_i\|_1 \\ & \text{subject to} \quad \|AW\tilde{x}_i - y_i\|_2 \leq \sigma. \end{aligned} \quad (7.9)$$

We adapt the spectral projected gradient algorithm presented by van den Berg and Friedlander [74]. Algorithm 2 formalizes the optimization problem to recover $\mathcal{X}$.

7.2.3. Matrix completion

The problem of matrix completion is to recover a matrix $Y \in \mathbb{R}^{R \times (M \times N)}$ of rank k with missing entries. This rank k constraint allows the matrix Y to be factored into two matrices, $U \in \mathbb{R}^{R \times k}$ and $S \in \mathbb{R}^{k \times (M \times N)}$. Here U represents a rank k subspace, and S are the k features for each pixel. We can solve the following optimization problem for $\hat{Y}$:

$$\begin{aligned} & \underset{\|S\|_2=1}{\text{minimize}} \quad \underset{U}{\text{minimize}} \quad \frac{1}{R} \left(\sum_{(i,j) \in \Omega} (\hat{Y}_{ij} - Y_{ij})^2 + \frac{1}{\gamma} \|U\|_2^2 \right) \\ & \text{subject to} \quad \hat{Y} = US. \end{aligned} \quad (7.10)$$

Algorithm 2. Basis pursuit with denoising

```
1: procedure BP( $\mathcal{Y}, A, W, \sigma$ )
2:   for  $i = 1, \dots, M \times N$  do
3:     Solve for  $\tilde{x}_i$  according to

               minimize  $\|\tilde{x}_i\|_1$ 
               subject to  $\|AW\tilde{x}_i - y_i\|_2 \leq \sigma$ 

4:      $x_i \leftarrow W \tilde{x}_i$ 
5:   end for
6:   return  $\mathcal{X}$ 
7: end procedure
```

Various solutions to the matrix completion problem have been proposed. We will utilize the projected gradient descent implementation presented by Bertsimas and Li [6]. Algorithm 3 formalizes the optimization problem to recover $\hat{Y}$.

Algorithm 3. Matrix completion

```
1: procedure MC( $\mathcal{Y}, k, \gamma, \Omega, \omega$ )
2:   Solve for  $\hat{Y}$  according to

               minimize  $\|S\|_2=1$    minimize  $\frac{1}{R} \left( \sum_{(i,j) \in \Omega} (\hat{Y}_{ij} - Y_{ij})^2 + \frac{1}{\gamma} \|U\|_2^2 \right)$ 
               subject to  $\hat{Y} = US$ 

3:   return  $\hat{Y}$ 
4: end procedure
```

The measurements $\mathcal{P}_\omega(\mathcal{Y})$ contain missing entries. We can define the following constraints on the full measurement tensor $\mathcal{Y}$:

1. Consistency constraint using projections. Suppose the uncompressed tensor $\mathcal{X}_0$ is given. Then $\mathcal{Y}$ should satisfy

$$\mathcal{Y}_{\text{proj}} = \mathcal{X}_0 \times_3 A$$

or equivalently,

$$Y_{\text{proj}} = AX_0. \quad (7.11)$$

We set the missing entries as

$$\mathcal{P}_{\Omega \setminus \omega}(Y) = \mathcal{P}_{\Omega \setminus \omega}(Y_{\text{proj}}) \quad (7.12)$$

where $\Omega \setminus \omega$ is the set of missing entries. Basis pursuit provides an estimate $\mathcal{X}$ from the projected measurement tensor. The two tensors $\mathcal{X}$ and $\mathcal{X}_0$ should be consistent. Algorithm 4 describes an implementation of this consistency constraint using projections.

Algorithm 4. Consistency constraint using projections

```
1: procedure BP-PROJ( $\mathcal{X}_0; Y, A, W, \Omega, \omega$ )
2:    $Y_{\text{proj}} \leftarrow AX_0$ 
3:    $\mathcal{P}_{\Omega \setminus \omega}(Y) \leftarrow \mathcal{P}_{\Omega \setminus \omega}(Y_{\text{proj}})$ 
4:    $\mathcal{X} \leftarrow \text{BP}(\mathcal{Y}, A, W, \sigma)$ 
5:   return  $\mathcal{X}$ 
6: end procedure
```

2. Low rank constraint using matrix completion. We impose a constraint on the measurement tensor $\mathcal{Y}$ to be low rank. The matrix completion algorithm produces Y_{MC} . We set the missing entries as

$$\mathcal{P}_{\Omega \setminus \omega}(Y) = \mathcal{P}_{\Omega \setminus \omega}(Y_{\text{MC}}). \quad (7.13)$$

Basis pursuit produces the uncompressed tensor $\mathcal{X}$ from the matrix completion estimate. Algorithm 5 implements this low rank constraint based on matrix completion. The procedure BP-MC takes X_0 as a dummy argument. It will become a mathematical convenience for later use.

Algorithm 5. Low rank constraint using matrix completion

```
1: procedure BP-MC( $\mathcal{X}_0; Y, A, W, \omega, \Omega$ )
2:    $Y_{\text{MC}} \leftarrow \text{MC}(\mathcal{Y}, k, \gamma, \omega, \Omega)$ 
3:    $\mathcal{P}_{\Omega \setminus \omega}(Y) \leftarrow \mathcal{P}_{\Omega \setminus \omega}(Y_{\text{MC}})$ 
4:    $\mathcal{X} \leftarrow \text{BP}(\mathcal{Y}, A, W, \sigma)$ 
5:   return  $\mathcal{X}$ 
6: end procedure
```

7.2.4. Consensus equilibrium

Consensus equilibrium is a framework to break a reconstruction problem into subproblems that can be solved separately. We define the tensor valued maps, $F_1, F_2 : \mathbb{R}^{M \times N \times B} \rightarrow \mathbb{R}^{M \times N \times B}$:

$$F_1(\mathcal{V}_1) := \text{BP-proj}(\mathcal{V}_1; Y, A, W, \Omega, \omega) \quad (7.14)$$

and

$$F_2(\mathcal{V}_2) := \text{BP-MC}(\mathcal{V}_2; Y, A, W, k, \gamma, \Omega, \omega) \quad (7.15)$$

where $\mathcal{V}_1, \mathcal{V}_2 \in \mathbb{R}^{M \times N \times B}$ and $Y, A, W, k, \gamma, \Omega, \omega$ are given. Further, let us concatenate the tensors and maps using the following notation:

$$\mathcal{V} = \begin{bmatrix} \mathcal{V}_1 \\ \mathcal{V}_2 \end{bmatrix} \in \mathbb{R}^{M \times N \times B \times 2} \quad (7.16)$$

and

$$F(\mathcal{V}) = \begin{bmatrix} F_1(\mathcal{V}_1) \\ F_2(\mathcal{V}_2) \end{bmatrix} \in \mathbb{R}^{M \times N \times B \times 2}. \quad (7.17)$$

Define another tensor valued map $G : \mathbb{R}^{M \times N \times B \times 2} \rightarrow \mathbb{R}^{M \times N \times B \times 2}$ as

$$G(\mathcal{V}) = \begin{bmatrix} \overline{\mathcal{V}} \\ \mathcal{V} \end{bmatrix} \quad (7.18)$$

Here

$$\overline{\mathcal{V}} = \frac{1}{2}(\mathcal{V}_1 + \mathcal{V}_2) \quad (7.19)$$

where addition and multiplication are pointwise. Let

$$T = (2G - I)(2F - I). \quad (7.20)$$

We are looking for a solution $\mathcal{V}^* \in \mathbb{R}^{M \times N \times B \times 2}$ that satisfies

$$T(\mathcal{V}^*) = \mathcal{V}^*. \quad (7.21)$$

This fixed point yields the uncompressed tensor:

$$\mathcal{X}^* = \overline{\mathcal{V}}^*. \quad (7.22)$$

Mann iterations can help to evaluate the fixed point:

$$\mathcal{V}^{k+1} = (1 - \rho)\mathcal{V}^k + \rho T(\mathcal{V}^k) \quad (7.23)$$

for a fixed parameter $\rho \in (0, 1)$. Addition and multiplication are defined pointwise. Algorithm 6 outlines this procedure to apply consensus equilibrium for compressive matrix completion.

Algorithm 6. Consensus equilibrium for compressive matrix completion

- 1: Initialize $\mathcal{V}^0 \in \mathbb{R}^{M \times N \times B \times 2}$ to any value.
 - 2: $k \leftarrow 0$
 - 3: **while** not converged **do**
 - 4: $\mathcal{V}^{k+1} \leftarrow (1 - \rho)\mathcal{V}^k + \rho T(\mathcal{V}^k)$
 - 5: $k \leftarrow k + 1$
 - 6: **end while**
 - 7: $\mathcal{X} \leftarrow \overline{\mathcal{V}}^k$
-

To compare two tensors $\mathcal{U}$ and $\mathcal{V}$ of the same size $M \times N \times B$, we define the mean squared error as

$$\text{MSE}(\mathcal{U}, \mathcal{V}) = \frac{1}{M \times N \times B} \sqrt{\sum_{i,j,k}^{M,N,B} (\mathcal{U}_{i,j,k} - \mathcal{V}_{i,j,k})^2} \quad (7.24)$$

where $\mathcal{U}_{i,j,k}$ denotes the (i, j, k) entry of $\mathcal{U}$.

7.3. Experiment

The Indian Pines dataset is a hyperspectral image over the Purdue University agronomy farm in West Lafayette, Indiana [5]. It consists of a 145×145 image with $B = 220$ bands, as measured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). The uncompressed tensor $\mathcal{X}$ is reduced to a size of $5 \times 5 \times 220$ by uniformly downsampling the original dataset by a factor of 32 along the spatial dimensions. This smaller size enables faster computations while still illustrating the concept.

We take $R = 110$ measurements to encode $B = 220$ bands for a compression ratio of $\kappa = 50\%$. The sampling matrix $A \in \mathbb{R}^{R \times B}$ has uniformly distributed entries:

$$A_{ij} \sim U(0, 1). \quad (7.25)$$

According to compressive sensing theory, random matrices satisfy the restricted isometry property with high probability. This enables a solution to be recovered with convex optimization. Other matrices may be substituted for A depending on the measurement system.

For basis pursuit, the spectra are compressed in Daubechies 8 (db8) wavelet coefficients. We set the denoising parameter $\sigma = 0.01$.

For matrix completion of the compressive measurements, we fix the rank $k = 6$. The regularization parameter $\gamma = 1 \times 10^6$, and the consensus equilibrium parameter $\rho = 0.5$. All parameters are fixed in the simulations.

The amount of missing data in the compressive measurements varies from $\mu = 10\%$ to $\mu = 90\%$ in increments of 10%. Missing entries in $\mathcal{Y}$ are selected randomly with probability μ . In real systems, missing data may result from transmission errors or deliberate subsampling of data. Figure 7-1 shows spectral reconstructions of one pixel from the Indian Pines hyperspectral data. Each plot varies the amount of missing data, including $\mu = 10\%$ through $\mu = 71\%$. The compression ratio is fixed at $\kappa = 50\%$. We use the metric for mean squared error defined in Eq. (7.24). This metric measures the difference between the ground truth for the uncompressed tensor and the reconstruction. Error increases with the amount of missing data in the compressive space, as plotted in Figure 7-2. The reconstructions show reasonable agreement with ground truth even though a majority of the data has been compressed or removed.

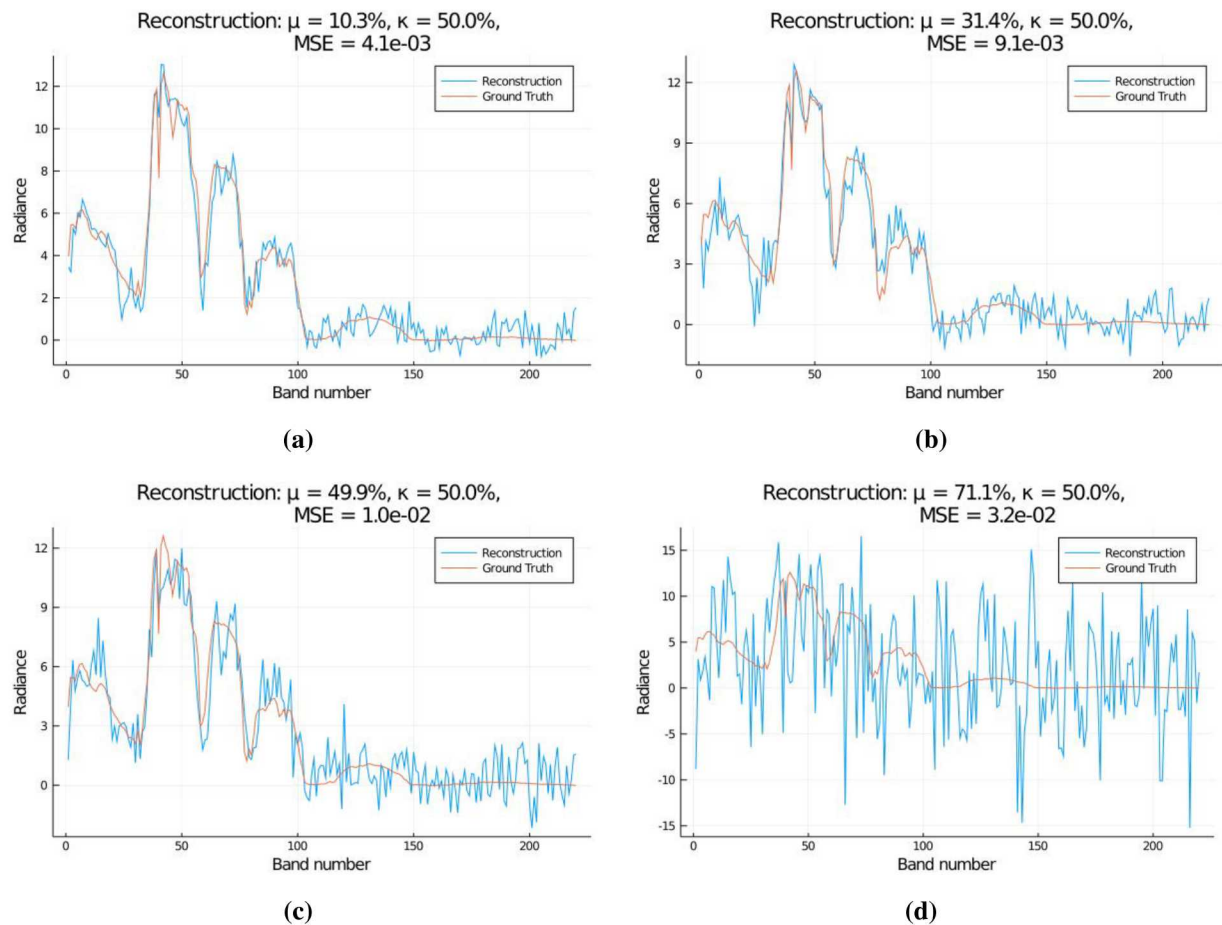


Figure 7-1. Reconstruction of compressive measurements with missing data. The plots show the spectra from one pixel of the Indian Pines hyperspectral data. μ denotes the fraction of missing data. In all simulations, the compression ratio $\kappa = 50\%$. Radiance is in units of $[\text{W cm}^{-2} \text{ nm}^{-1} \text{ sr}^{-1}]$.

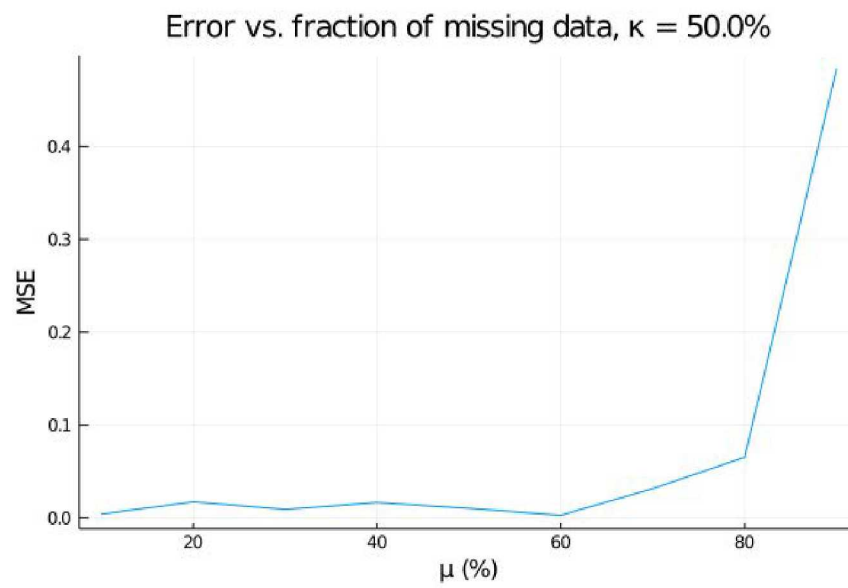


Figure 7-2. Error increases with the amount of missing data in the compressive space. In all simulations, the compression ratio $\kappa = 50\%$.

7.4. Conclusion

We have proposed a technique for reconstruction from incomplete compressive measurements. Our approach combines compressive sensing and matrix completion using the consensus equilibrium framework. The algorithm imposes two constraints on the solution. First, the compressed tensor should be consistent with the uncompressed tensor $\mathcal{X}$ when it is projected onto the low-dimensional subspace. Second, the measurements should live in a low rank subspace when the missing entries are completed. These constraints allow us to solve for the uncompressed tensor when data is missing in the compressive space. We validate our approach on the Indian Pines hyperspectral dataset. The spectra are first compressed by 50%, and then tensor elements are randomly removed. The reconstructions show reasonable agreement with ground truth even though a majority of the data has been compressed or removed. This work opens up new possibilities for data reduction, compression, and reconstruction.

REFERENCES

- [1] I. August, Y. Oiknine, M. AbuLeil, I. Abdulhalim, and A. Stern. Miniature compressive ultra-spectral imaging system utilizing a single liquid crystal phase retarder. *Scientific Reports*, 6, 2016.
- [2] Yitzhak August and Adrian Stern. Compressive sensing spectrometry based on liquid crystal devices. *Optics Letters*, 38(23):4996–4999, 2013.
- [3] W. U. Bajwa, J. D. Haypt, G. M. Raz, S. J. Wright, and R. D. Nowak. Toeplitz-structured compressed sensing matrices. *2007 IEEE/14th Workshop on Statistical Signal Processing, Vols 1 and 2*, 2007.
- [4] A. Barbero and S Sra. Modular proximal optimization for multidimensional total-variation regularization. <http://arxiv.org/abs/1411.0589>, 2017.
- [5] Larry L. Baumgardner, Marion F. Biehl and David A. Landgrebe. 220 band AVIRIS hyperspectral image data set: June 12, 1992 indian pine test site 3. <https://purrr.purdue.edu/publications/1947/1>, Sep 2015.
- [6] D. Bertsimas and M. Li. Fast exact matrix completion: A unifying optimization framework. *J. Mach. Learning Research*, 2, 2019.
- [7] J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.
- [8] J. M. Bioucas-Dias and M. A. T. Figueiredo. Two-step algorithms for linear inverse problems with non-quadratic regularization. *IEEE International Conference on Image Processing*, 2007.
- [9] M. Borgerding, P. Schniter, and S. Rangan. Amp-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.
- [10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3, 2011.
- [11] Stephen Boyd, Neal Parikh, and Eric Chu. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and trends in machine learning*, 3(1):1–122, 2011.
- [12] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

- [13] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman. Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences*, 11(3):2001–2020, 2018.
- [14] E. J. Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592, 2008.
- [15] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [16] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [17] E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
- [18] Emmanuel J. Candes and Justin K. Romberg. Signal recovery from random projections. *Proc SPIE*, 5674:76–86, 2005.
- [19] Emmanuel Candès and Justin Romberg. L1-magic.
<http://statweb.stanford.edu/~candes/l1magic/>. Accessed: Dec 12, 2018.
- [20] C. H. Chen, editor. *Compressive Sensing of Earth Observations*. CRC Press, Boca Raton, 2017.
- [21] Valerie C. Coffey. Hyperspectral imaging for safety and security. *Optics & Photonics News*, 26:26–33, 2015.
- [22] MathWorks Corporation, 2018.
- [23] M. J. DeWeert and B. P. Farm. Lensless coded-aperture imaging with separable doubly-toeplitz masks. *Optical Engineering*, 54(2), 2015.
- [24] A. Domahidi, E. Chu, and S. Boyd. Ecos: An socp solver for embedded systems. *2013 European Control Conference (ECC)*, pages 3077–3082, 2013.
- [25] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [26] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.
- [27] E. R. Dowski and W. T. Cathey. Extended depth of field through wave-front coding. *Applied Optics*, 34(11):1859–1866, 1995.
- [28] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.
- [29] Michael T. Eismann. *Hyperspectral remote sensing*. SPIE Press, Bellingham, WA, 2012.

- [30] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Trans Image Process*, 13(10):1327–44, 2004.
- [31] A. S. Fruchter and R. N. Hook. A novel image reconstruction method applied to deep hubble space telescope images. *Proc SPIE*, 3164:120–125, 1997.
- [32] Daniel Gedalin, Yaniv Oiknine, Isaac August, Dan G Blumberg, Stanley R Rotman, and Adrian Stern. Performance of target detection algorithm in compressive sensing miniature ultraspectral imaging compressed sensing system. *Optical Engineering*, 56(4):041312 1–6, 2017.
- [33] M. He, B. Li, and H. Chen. Multi-scale 3d deep convolutional neural network for hyperspectral image classification. *ICIP*, 2017.
- [34] E. Huang, Q. Ma, and Z. W. Liu. Etalon array reconstructive spectrometry. *Scientific Reports*, 7, 2017.
- [35] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. Deep fully-connected networks for video compressive sensing, 2016.
- [36] C. Jin, S. M. Kakade, and P. Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. *Advances in Neural Information Processing Systems 29 (Nips 2016)*, 29, 2016.
- [37] C. Kim, W. B. Lee, S. K. Lee, Y. T. Lee, and H. N. Lee. Fabrication of 2d thin-film filter-array for compressive sensing spectroscopy. *Optics and Lasers in Engineering*, 115:53–58, 2019.
- [38] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [39] Drew P. Kouri and Eric A. Shields. Efficient multiframe super-resolution for imagery with lateral shifts. *Applied Optics*, 53(24):F1–F9, 2014.
- [40] Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>. Accessed: Dec 12, 2018.
- [41] D. J. Lee. Compressive sensing for channeled polarimetry: Applications in spectropolarimetry and imaging polarimetry. *Proc. SPIE*, 11130, 2019.
- [42] D. J. Lee. Deep neural networks for compressive hyperspectral imaging. *Proc. SPIE*, 11130, 2019.
- [43] D. J. Lee. Matrix completion for compressive sensing using consensus equilibrium. *Proc. SPIE*, 11504, 2020.
- [44] D. J. Lee, C. A. Bouman, and A. M. Weiner. Single shot digital holography using iterative reconstruction with alternating updates of amplitude and phase. <http://www.arxiv.org/abs/1609.02978>, 2016.

- [45] D. J. Lee, K. Han, H. J. Lee, and A. M. Weiner. Synthetic aperture microscopy based on referenceless phase retrieval with an electrically tunable lens. *Applied Optics*, 54(17):5346–5352, 2015.
- [46] D. J. Lee, C. F. LaCasse, and J. M. Craven. Channeled spectropolarimetry using iterative reconstruction. *Proc SPIE*, 9853, 2016.
- [47] D. J. Lee, C. F. LaCasse, and J. M. Craven. Compressed channeled linear imaging polarimetry. *Proc. SPIE*, 10407, 2017.
- [48] D. J. Lee, C. F. LaCasse, and J. M. Craven. Compressed channeled spectropolarimetry. *Optics Express*, 25(25):32041–32063, 2017.
- [49] D. J. Lee and E. A. Shields. Compressive hyperspectral imaging using total variation minimization. *Proc. SPIE*, 10768, 2018.
- [50] D. J. Lee and A. M. Weiner. Optical phase imaging using a synthetic aperture phase retrieval technique. *Optics Express*, 22(8):9380–9394, 2014.
- [51] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3), 2007.
- [52] Y. Li, H. Zhang, and Q. Shen. Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sens.*, 9(67), 2017.
- [53] P. Llull, X. J. Liao, X. Yuan, J. B. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady. Coded aperture compressive temporal imaging. *Optics Express*, 21(9):10526–10545, 2013.
- [54] R. E. Marcia and R. M. Willett. Compressive coded aperture superresolution image reconstruction. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Vols 1-12*, pages 833–836, 2008.
- [55] R. F. Marcia, Z. T. Harmany, and R. M. Willett. Compressive coded aperture imaging. *Proc. SPIE*, 7246, 2009.
- [56] R. F. Marcia, Z. T. Harmany, and R. M. Willett. Compressive coded apertures for high-resolution imaging. *Optics, Photonics, and Digital Technologies for Multimedia Applications*, 7723, 2010.
- [57] L. Mou, P. Ghamisi, and X. Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geos. Remote Sens.*, 55(7), 2017.
- [58] Taku Nakahara, Kazunori Uruma, Tomohiro Takahashi, and Toshihiro Furukawa. Single-frame super-resolution using superpixel based dictionary. *IEEE 59th International Midwest Symposium on Circuits and Systems*, 2016.
- [59] E. D. Nelson and M. L. Fredman. Hadamard spectroscopy. *Journal of the Optical Society of America*, 60(12):1664–1669, 1970.
- [60] Yaniv Oiknine, Isaac August, Dan G Blumberg, and Adrian Stern. Compressive sensing resonator spectroscopy. *Optics Letters*, 42(1):25–28, 2017.

- [61] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.
- [62] Vishal M. Patel and Rama Cellapa. *Sparse Representations and Compressive Sensing for Imaging and Vision*. SpringerBriefs in Electrical and Computer Engineering. Springer, New York, 2014.
- [63] Pan Qi. Single-frame image super-resolution based on in-place patch match model. *5th International Conference on Digital Home*, pages 91–94, 2014.
- [64] C. L. Qiu, N. Vaswani, B. Lois, and L. Hogben. Recursive robust pca or recursive sparse recovery in large but structured noise. *IEEE Transactions on Information Theory*, 60(8):5007–5039, 2014.
- [65] Q. Ran, W. Li, Q. Du, and C. H. Yang. Hyperspectral image classification for mapping agricultural tillage practices. *Journal of Applied Remote Sensing*, 9, 2015.
- [66] J. Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):14–20, 2008.
- [67] Klaus Roth. *NMR tomography and spectroscopy in medicine: an introduction*. Springer-Verlag, Berlin, 2012.
- [68] S. Sreehari, S. V. Venkatakrisnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423, 2016.
- [69] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.
- [70] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision*, pages 839–846, 1998.
- [71] Caglayan Tuna, Gozde Unal, and Elif Sertel. Single-frame super resolution of remote-sensing images by convolutional neural networks. *International Journal of Remote Sensing*, 39(8):2463–2479, 2018.
- [72] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction. <http://www.cs.ubc.ca/labs/scl/spgl1>. Accessed: Dec 12, 2018.
- [73] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.
- [74] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.
- [75] Ewout van den Berg and Michael P Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.
- [76] R. M. Willett, R. F. Marcia, and J. M. Nichols. Compressed sensing for practical optical imaging systems: a tutorial. *Optical Engineering*, 50(7), 2011.

- [77] Chih-Yuan Yang, Jia-Bin Huang, and Ming-Hsuan Yang. Exploiting self-similarities for single frame super-resolution. *ACCV'10 Proc of the 10th Asian Conference on Computer Vision*, pages 497–510, 2010.
- [78] Kaibing Zhang, Jie Li, Haijun Wang, Xiuping Liu, and Xinbo Gao. Learning local dictionaries and similarity structures for single image super-resolution. *Signal Processing*, 142:231–243, 2018.
- [79] Y. N. Zhang and M. Q. An. Deep learning- and transfer learning-based super resolution reconstruction from single medical image. *Journal of Healthcare Engineering*, 2017.

DISTRIBUTION

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	D. Chavez, LDRD Office	1911	0359

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov



Sandia
National
Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.