DETECTION AND CLASSIFICATION IN THE COMPRESSED DOMAIN FOR MULTISPECRTRAL IMAGING

BY

YUQI LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Yoram Bresler

ABSTRACT

Various applications would benefit from rapid inference on multispectral images at the point of sensing. However, the acquisition of a full-resolution multispectral image requires advanced spectrometers and prohibitive sensing time. Also, performing the high-level vision tasks such as classification and segmentation on the multispectral data consumes more computation power than on the common RGB images. Compressed sensing (CS) circumvents this sensing process usually using a random sensing matrix to acquire fewer measurements and reconstructs the multispectral image based on a sparsity assumption. The further high-level analysis of images is performed on the reconstructed high-dimensional images. And a random sensing matrix may not be physically realizable or the best fit for extracting information pertaining to a high-level vision task. A realizable low-cost data acquisition scheme and a fast processing system that makes inference based on the acquired signal are desired for multispectral images.

In this thesis, we present a systematic way to jointly optimize the sensing scheme subject to optical realizability constraints, and make inference of the multispectral image in the compressed domain.

In the first part of the thesis, we state some open questions in compressed inference. We review the theory on inference in the compressed domain. We formulate the problem for compressed inference and state metrics to evaluate the inference performance. We then review some existing realizable optical compressed sensing imaging systems designed for multispectral images and derive the forward model of data acquisition. The feasibility of performing detection, classification and segmentation in the compressed domain directly is then discussed for the multispectral images. Using tools from detection and estimation theory, we derive the optimal decision rule to perform compressed detection, classification and segmentation in a simple data setting. Also, the feasibility of adjusting the optical acquisition

schemes jointly with the neural network is discussed. The architecture of neural networks that can achieve the performance of the optimal decision rule is proposed and the existence of optimal weights is discussed.

Next, we use a synthetic dataset to compare the performance of the proposed neural network and the optimal decision rule. Several synthetic multispectral image datasets and a clinical tumor biopsy dataset are used to verify the improvement of the obtained sensing scheme and compare the performance of the neural network with that of a known optimal decision rule.

To my family, for their love and support.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to Prof. Yoram Bresler for his insightful discussion, patient guidance, and warm encouragement throughout this MS thesis. Prof. Bresler's unwavering pursuit of excellence constantly motivates me to take one more step forward in research. He is not only a brilliant advisor but also a wise mind who teaches me beyond academic research.

I would like to thank my colleagues, collaborators and friends, Dr. Luke Pfister, Dr. Bihan Wen, Dr. Yanjun Li, Ankit Raj, Lingda Wang, Hanfei Wang, Zhinan Hu, Chao Pan, Berk Iskender, Ufuk Soylu and others. I learned a lot from working together with them. I am very grateful for their tremendous help in graduate school and in life.

Lastly, I owe my special thanks to my parents, Jinping Yu and Haoping Li, for their unconditional support and endless love. Their understanding and encouragement always back me up during this unusual Covid-19 time and all my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CS	Compressed Sensing
CSI	Compressed Sensing Imaging
DMD	Digital Micromirror Devices
DNN	Deep Neural Network
DOF	Degree of Freedom
GPU	Graphics Processing Unit
MSE	Mean Squared Error
PCA	Principle Component Analysis
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Various applications would benefit from rapid inference on multispectral images at the point of sensing. Inference tasks such as classification or segmentation for multispectral images are common in agricultural [1, 2], medical [3] and remote sensing applications [4]. A low-cost data acquisition scheme and a fast processing system that makes inference based on the acquired signal are then desired for multispectral images.

Traditionally, obtaining the high spatial-spectral-resolution images requires advanced, expensive sensors and takes much longer than a consumer-grade camera. NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) is an existing multispectral imaging system for airborne platforms that captures 224 spectral bands [5] for a single image. The size of the captured multispectral image cube is therefore hundreds times larger than a common RGB image with the same spatial resolution. The subsequent analysis and inference task of the multispectral image are performed after the whole image is obtained. Performing high-level inference tasks on this large volume 3D image, such as detection, classification and segmentation, is computationally extensive.

Several optical compressed sensing imaging (CSI) systems have been proposed [6–9] to circumvent the expensive and slow sensing process by taking fewer measurements than the number of voxels of a multispectral image using cheaper sensors. Compressed sensing (CS) theory is the pillar of these practical systems: Given enough measurements and assuming that the original signal is sparsifiable, i.e. there exists a basis under which the signal is sparse, the exact recovery of the signal from the measurements is ensured [10].

In the last decade, compressed sensing has emerged as a framework that can significantly reduce the sensing cost by taking fewer measurements than the signal dimension and recovering the signal from the measurements

computationally. The fundamental works [10–13] show that a signal $x \in \mathbb{R}^n$ can be recovered from a small set of nonadaptive, linear, and usually randomized measurements $y = Ax \in \mathbb{R}^m$, provided the signal satisfies the sparsity property. The sparsity property assumes a natural image x is sparse in some basis Ψ with a sparse code α, that is, $x = \Psi\alpha + \epsilon$, where ϵ is a small approximation error. The sparse code α is often evaluated using the ℓ_0 or its convex substitute ℓ_1 norm, e.g., α satisfies $\|\alpha\|_0 \leq s$ or $\|\alpha\|_1 \leq t$. The MAP estimate of the signal x in CS problem is expressed as:

$$\begin{aligned} \hat{\alpha} &= \arg\min_{\alpha \in \mathbb{R}^n} \|y - A\Psi\alpha\|^2 + R(\alpha) \\ \hat{x} &= \Psi\hat{\alpha} \end{aligned} \tag{1.1}$$

where $R(\alpha)$ is a sparse-promoting regularizer of α and is related to its prior distribution. For example, for Laplacian prior of α, this regularizer is $\mu\|\alpha\|_1$ (ℓ_1 minimization). Another common regularizer $\mu\|\alpha\|_0$ penalizes the number of non-zero elements in α (ℓ_0 minimization). There are many algorithms to recover the original signal assuming the sparsity property. Basis pursuit [14] and orthogonal matching pursuit [15] are heuristic algorithms to solve for the ℓ_0 minimization. LASSO [16], feature-sign [17], forward-backward splitting [18] and FISTA [19] are proposed for ℓ_1 minimization problem with convergence guarantee on the signal x.

Single-pixel camera [9] is an immediate example of a CSI system. A digital micromirror device (DMD) is used to spatially modulate the image and obtain the single-pixel measurement. The original image could be reconstructed by solving a nonlinear optimization iteratively [15, 16, 19].

But the single-pixel camera cannot directly apply to multispectral imaging systems because the key component, the DMD, has no spectral selectivity. Coded aperture snapshot spectral imaging [7] and its variations [20, 21] are proposed to perform multispectral CS imaging. It utilizes a coded aperture and a disperser lens to achieve the modulation on both spectral and spatial domain. The principle of multispectral image reconstruction is similar to that of the monotone image reconstruction. The multispectral images are also assumed to be sparse under some transformation and could be reconstructed using the sparsity property [20].

Though a random matrix is proved to be a universal sensing matrix for CS [22], a designed sensing matrix could improve the reconstruction quality

with the same number of measurements. Finding such a sensing matrix based on the data has been studied [23, 24]. But these algorithms do not yield a physically realizable sensing matrix in optical systems. To optimize the tunable optical element, i.e., the realizable sensing matrix in a CSI system for better reconstruction accuracy, some approaches use a surrogate metric such as restricted isometry property [20, 25], while others optimize the element jointly with the reconstructor [26].

When we consider the subsequent analysis or inference tasks, they are often performed in the data domain, which means the high-dimensional multispectral images are reconstructed from the measurements first. If we simply stitch the pipelines, e.g., reconstructing the images using a designed matrix and then classifying the images, then there are several shortcomings: First, the optimization of reconstruction relies on image similarity metrics such as ℓ_2 distance [26], but a reconstructed image with lower ℓ_2 error does not necessarily imply better inference performance. Second, when the number of measurements does not satisfy the sufficient condition of the perfect reconstruction, then the subsequent inference may suffer from the corruption induced by the reconstruction. For example, an end-to-end network-based reconstructor may be unstable and bring extra reconstruction error [27]. Third, performing an inference task only requires the relevant sufficient statistic, whose dimension may be much lower than that of the reconstruction. The last concern is the computational time and memory cost, which can be significantly reduced if the reconstruction step is eliminated and the inference happens in a lower dimension.

As early as 2009, the concept of compressed learning based on the compressed sensing has been proposed. The compressed learning focuses on the manipulation of the signal in the measurement domain rather than the recovery of the signal. Calderbank et al. [28] show that the soft-margin support vector machine (SVM) is able to classify in the compressed domain. Also, the generalization loss of a SVM in the compressed domain is bounded using the restricted isometry property (RIP) of the sensing matrix. Davenport et al. [29] study several signal processing problems, namely detection, classification and estimation, in the compressed domain. They provide the error bounds of several signal processing problems for a random sensing matrix A. Durrant et al. give an average-case bound on the classification error of Fisher's linear discriminant classifier in the

compressed domain, with a random sensing matrix and the full knowledge of the statistic of the signal [30]. An extended work provides sharp bounds on the generalization error of a generic linear classifier on the compressed domain [31]. A more general error probability bound using information theory is given for any decision rule and for non-sparse signals [32]. We follow the CS signal processing framework [29] to derive the theory for compressed inference with full statistics of the data, and compare this theoretical prediction with the experimental results.

Many works use handcrafted algorithms to optimize the sensing matrix in compressed sensing imaging systems, but the acquisition process has not been jointly optimized with various inference tasks. The NuMax algorithm [23] has been used to optimize the modulation process in compressed imaging systems, which improves the classification performance of the compressed domain compared to a random modulation. But this work [24] focuses on grayscale images. Linear filters and linear SVMs have been used to conduct the face classification directly in compressed domain [33]. Meanwhile, this work shows that when the compression ratio is above 100 and the reconstruction is falling apart, the classification task sacrifices less than 10% in accuracy

CS learning for multispectral images remains unexplored. It is shown that for multispectral images, the class information is redundant in the spectral domain [34, 35]. The 3D remote sensing image can be compressed via linear dimension reduction techniques such as PCA, and a SVM/CNN classifier is able to predict effectively the class label from a compressed image. The acquisition schemes mentioned above are performed in silicon and pose the technical difficulty of implementing a linear transform such as PCA in optical systems. Instead, we would like an on-board optical sensing system that performs compression at the speed of light. This sensing system is very different from the imaging system because the reconstruction of the image is no longer a priority compared to inference tasks.

At the same time, the optimization of the sensing matrix for inference quality on multispectral images is also not studied. There are recent works that optimize the optical system for the enhancement of images [36], or for a fast acquisition scheme [37]. But these works do not involve optimizing the backend CS imaging system with respect to the inference quality.

We summarize the currently unanswered questions as follows:

1. How well can a CSI system extract information for specific inference tasks, in terms of the fundamental information limits?

2. Does the availability of the full statistical model of the data affect the CS learning performance?

3. How does the constraint of the acquisition schemes affect the CS learning performance?

4. How can we find the optimal sensing matrix for CS learning under the constraint of the optical system?

5. How do the fundamental bounds of the inference performance change with the number of measurements and the complexity of the inference task?

6. If using a deep neural network to optimize the acquisition scheme and the predictor at the same time, how different is the learned acquisition scheme from the underlying optimal acquisition scheme?

7. What is the best network architecture to learn the acquisition scheme and the predictor at the same time?

In this thesis, we explore the possibility of performing high-level vision tasks directly in the compressed domain, without reconstructing the high-dimensional image. At the same time, we aim to jointly optimize the CS acquisition subject to physical constraint with the inference for a better inference accuracy. We propose an approach for learning a deep neural network (DNN) for inference on compressively sensed multispectral image data directly in the compressed domain, jointly with the optimization of the CS acquisition. We do so for two simply realizable CSI systems and another coded-aperture-based CSI system for inference tasks such as classification and semantic segmentation of multispectral images. The approach readily extends to other CSI systems and other inference tasks. For quantitative performance assessment, we compare the inference performance of the DNN in the compressed domain with the optimal decision rule on a synthetic multispectral image dataset. We apply the same approach for a medical tumor dataset and evaluate the segmentation performance in the compressed domain.

This thesis is structured as follows: Chapter 2 gives a brief review of compressed sensing and states the problem. We introduce the optimal decision rule when the full information of the data is present. Chapter 3 introduces existing optical CS systems and formulates optimization problems for compressed signal acquisition processes. Chapter 5 shows the qualitative comparison of the optimal decision rule with three simple neural networks of designed architecture, on a monochrome synthetic dataset. Chapter 6 and Chapter 7 present the experiments to validate the feasibility of optical CS inference on synthetic and real multispectral datasets, respectively. Finally, Chapter 8 concludes this thesis and discusses further research directions.

CHAPTER 2

PROBLEM STATEMENT

2.1 Notation

Table 2.1 defines the symbols and operators we use in the following chapters.

Table 2.1: Notation table.

Symbol	**Meaning**	**Symbol**	**Meaning**
x	Signal in the data domain	y	signal in the compressed domain
A	Sensing matrix	v	Additive noise
σ	Noise level	z	Inference target, such as class label
C	Number of classes	N	Side length of a square multispectral image
H	Height of a multispectral image	W	Width of a multispectral image
B	Number of spectral bands	K	Number of snapshots taken in CS systems
D	Number of samples in a training set	P_e	Error probability
f	Neural-network based predictor	Q	Q function
θ	Parameter in the neural network	w	Parameter that determines the state of sensing matrices
π	Prior probability of a class	μ	Mean of the signal in hypotheses
CELoss	Cross entropy loss	$ReLU$	Rectified linear unit
Operator	**Meaning**	**Operator**	**Meaning**
diag	Diagonalization operator	$\otimes$	Kronecker product
$\odot$	Schur-Hadamard product (Element product)	$\mathbb{1}\{\cdot\}$	Indicator function
$\star$	Convolution		

2.2 Multispectral Compressive Imaging

Consider a B band multispectral $H \times W \times B$ image cube (Figure 2.1) of height H and width W. Let $x \in \mathbb{R}^n$, with $n = HWB$, represent the vectorized version of the image cube.

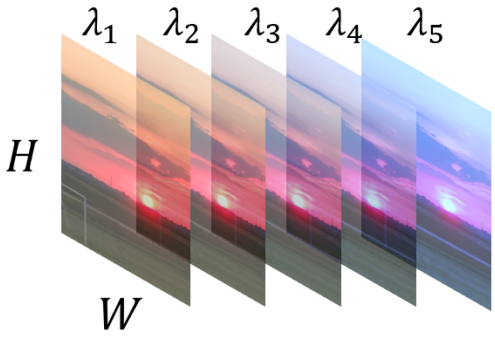


Figure 2.1: A multispectral $H \times W \times B$ image cube, $B = 5$.

We assume that the data is sensed by a linear compressive sensor represented by matrix $A \in \mathbb{R}^{m \times n}$, where $m < n$, producing the measurement

$$y = Ax \tag{2.1}$$

The measurement could include noise as well, but we choose to model all the uncertainty in the problem in the signal x itself. This allows us to focus on the effect of compressive sensing, i.e., on reducing the dimension of the measurement from n to m rather than on the reduction in SNR due to fewer measurements.

We will describe in Chapter 3 several practical CSI systems that define specific, constrained forms of A. Denoting the set of feasible sensing matrices by $\mathcal{A}$, one could optimize A subject to the constraint $A \in \mathcal{A}$ for better inference performance. However, especially with discrete—e.g., binary—constraints, the optimization could be NP hard and become infeasible for A of practical dimensions.

An alternative is to represent the tunable parameters in the CSI systems by a vector $w \in \mathbb{R}^d$, where d is the number of tunable variables, or degrees of freedom (DOF) in the acquisition system, and typically, for the optical systems we consider, $d \ll mn$. The optimization of the sensing matrix can then be reduced to the optimization of w in a much lower dimension. We

therefore represent the measurements as

$$y = A(w)x \tag{2.2}$$

As usual in the context of compressive sensing, we assume that the signal x is compressible, that is, $x \in \mathcal{X}$, where the set $\mathcal{X}$ has, in some sense, dimensionality lower than the dimension n of the ambient space. Examples of such $\mathcal{X}$ include the set of signals sparsely representable (at a fixed sparsity level) by a known or learnable dictionary [38], or the set of signals sparsifiable by a known or learnable transform [39, 40], or the set of signals that live on a lower dimensional manifold [41, 42]. When the specific form of $\mathcal{X}$ is not material, we will simply refer to the signal x as *compressible*.

2.3 Inference Problems in the Compressed Domain

As reviewed in Chapter 1, the classical formulation of compressive sensing (CS) addresses the reconstruction of $x \in \mathbb{R}^n$ from the lower-dimensional measurements $y \in \mathbb{R}^m$. Likewise, traditionally, to perform inference tasks with a CSI system one reconstructs the signal x first. Instead, for reasons discussed in Chapter 1, in this thesis we consider performing the inference tasks directly in the compressed domain.

In the following subsections, we formulate the compressed inference problems addressed in this thesis: detection, classification, and semantic segmentation.

2.3.1 Detection in the compressed domain

We consider the detection of a known compressible signal $x \in \mathbb{R}^n$ on an additive noise background, when it is compressively sensed by $A \in \mathbb{R}^{m \times n}$. The signal is absent under the null hypothesis and present under the alternative. For the sake of simplicity, we assume that the noise v is white Gaussian and has the same variance under the null and alternative hypotheses. This results in a decision problem between the following

hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: y = Av, v \sim \mathcal{N}(0, \sigma^2 I) \\ \mathcal{H}_1 &: y = A(x+v), v \sim \mathcal{N}(0, \sigma^2 I) \end{aligned} \tag{2.3}$$

If the prior probability of the null hypothesis is unknown, then, following the Neyman-Pearson formulation, we aim to maximize the detection rate $P_D \triangleq \Pr(\mathcal{H}_1 \text{ chosen when } \mathcal{H}_1 \text{ true})$ at a given false alarm rate $P_F \triangleq \Pr(\mathcal{H}_1 \text{ chosen when } \mathcal{H}_0 \text{ true})$. The performance of the detector in this case is typically evaluated using an ROC curve [43]. On the other hand, if the two hypotheses have known prior π_0 and π_1, respectively, then we aim to minimize the error probability.

$$P_e \triangleq \pi_0 P(\mathcal{H}_1 \text{ accepted} \mid \mathcal{H}_0) + \pi_1 P(\mathcal{H}_0 \text{ accepted} \mid \mathcal{H}_1) \tag{2.4}$$

which serves as the performance metric.

In the real world, the signal x in the data domain can also be a random variable. For example, we may not know whether there is a human face in an image, or when present, where the face is located in the image. We consider two scenarios in the compressed sensing detection problem:.

1. Detection of a known object with a known position
2. Detection of a known object with an unknown position

The first scenario is the known signal case considered above, with hypotheses represented more succinctly as:

$$\begin{aligned} H_0 &: y = Ax, x \sim \mathcal{N}(0, \sigma^2 I) \\ H_1 &: y = Ax, x \sim \mathcal{N}(\mu, \sigma^2 I) \end{aligned} \tag{2.5}$$

where μ is the mean of the image when a known object appears in the known position.

For the second scenario, we consider a simple case: the B band image contains a single object of size $M \times M$ pixels randomly placed in an $N \times N$ pixel background. We formulate this problem as composite binary hypothesis testing with the following two hypotheses, because the location

of the object can be treated as a latent parameter.

$$\begin{aligned} H_0 &: y = Ax, x \sim \mathcal{N}(\mu, \sigma^2 I), \mu \in \Omega_0 = \{0\}, \\ H_1 &: y = Ax, x \sim \mathcal{N}(\mu, \sigma^2 I), \mu \in \Omega_1 \\ \Omega_1 &= \text{Set of images containing the known object} \\ &\quad \text{at some position, on a zero background.} \end{aligned} \tag{2.6}$$

The latent parameter μ, the mean of the Gaussian distribution, is determined by the location of the object. The set of states Ω_1 has $(N - M + 1)^2$ elements and each element $\mu \in \Omega_1$ represents the mean of the image. For the composite hypothesis H_1, determining the object location reduces to a classification problem, given that locations are discrete. Similar to the known signal case, we also aim to minimize the error probability when the prior probabilities are known and otherwise maximize the detection rate for a given false alarm rate.

2.3.2 Classification in the compressed domain

We formulate classification in the compressed domain as a C-nary hypothesis testing problem. Under hypothesis $\mathcal{H}_i$, we have a measurement of a known compressible signal x_i on a white Gaussian noise background:

$$\mathcal{H}_i : y = A(x_i + v), v \sim \mathcal{N}(0, \sigma^2 I), \text{ for } i = 1, \ldots, C \tag{2.7}$$

With known prior probabilities, we aim to minimize the probability of misclassification, i.e., the error probability:

$$P_e \triangleq \sum_{i=1}^{C} \pi_i P(\mathcal{H}_i \text{ rejected} \mid \mathcal{H}_i) \tag{2.8}$$

2.3.3 Segmentation in the compressed domain

Image segmentation is to partition a given image into meaningful subregions that delineate the objects or scenes in the image. These subregions could be overlapping with each other or disjoint, depending on the purpose of the segmentation. This is the highest level statistical

inference task that we consider.

We consider a simple exclusive labelling scenario, i.e., one pixel can only belong to one class. For an image x of size $N \times N \times B$, where B is the number of spectral bands, semantic segmentation aims to assign one class to each pixel and yield a size $N \times N$ label map. Similar to classification, we can write a C-nary hypothesis for each pixel, resulting in C^{N^2} possible labels for the whole image.

Denoting by x_i the mean of the image corresponding to the ith label of the image, $i \in [C^{N^2}]$, the hypotheses of compressed segmentation are

$$\mathcal{H}_i : y = A(x_i + v), v \sim \mathcal{N}(0, \sigma^2 I) \tag{2.9}$$

There are several metrics to evaluate the quality of segmentation. The error probability

$$P_e \triangleq \sum_{i=1}^{C^{N^2}} \pi_i P(\mathcal{H}_i \text{ rejected} \mid \mathcal{H}_i) \tag{2.10}$$

is usually not a meaningful metric for image segmentation because it considers any imperfect segmentation as an error event, thus assigning equal weight to a segmentation that is wrong by one pixel, and is therefore visually almost correct, as to one that misclassifies all pixels, and is therefore useless. Instead, other metrics are used to evaluate image segmentation, to better correspond to their utility in applications.

Some of the more commonly used image segmentation metrics [44] are pixel-wise accuracy and mean IOU (intersection over union). Let the $z_* \in [C]^{N \times N}$ denote the ground truth segmentation, and $\widehat{z} \in [C]^{N \times N}$ denote the predicted segmentation. The pixel-wise accuracy is defined as the fraction of correctly labelled pixels,

$$\text{Pixel-wise Accuracy} \triangleq \frac{|z_* == \widehat{z}|}{N^2} \tag{2.11}$$

where == compares two matrices pixel-by-pixel and returns a Boolean matrix, and $|\cdot|$ counts the number of True elements in the Boolean matrix.

Given the predicted segmentation and the ground truth, the IOU of class $c \in [C]$ is defined as the intersection between the prediction's cth class and the groundtruth's cth class divided by their union. Let $S_c(z_*)$ and $S_c(\widehat{z})$ denote the support on which the pixel is labelled as cth class in the

groundtruth and in the predicted segmentation, respectively. The IOU of class c is then

$$\mathrm{IOU}_c(z^*, \widehat{z}) \triangleq \frac{|S_c(z_*) \bigcap S_c(\widehat{z})|}{|S_c(z_*) \bigcup S_c(\widehat{z})|}, c \in [C] \tag{2.12}$$

and the unweighted mean IOU is the mean of the IOUs of all classes

$$\mathrm{mIOU}(z^*, \widehat{z}) \triangleq \frac{1}{C} \sum_{i=1}^{C} \mathrm{IOU}_c(z^*, \widehat{z}) \tag{2.13}$$

For compressed segmentation, we aim to minimize the error probability and maximize the pixel-wise accuracy and mIOU.

2.4 Inference Using Deep Neural Networks (DNN)

In real problems, often the full statistics of the data are unknown, but we do possess D samples $\{x^{(i)}\}_{i=1}^{D}$ or compressed samples $\{y^{(i)}\}_{i=1}^{D}$ drawn from the underlying distribution $\mathcal{X}$ or from $\mathcal{Y}$ with a given sensing matrix A, and their corresponding label $\{z^{(i)}\}_{i=1}^{D}$. For example, it is easier to obtain MRI measurements for multiple subjects than construct a generative model of the data source.

We can use a deep neural network with suitable architecture to learn the inference. The goal of this neural network is to minimize the error probability of the prediction, given D pairs of training samples $\{(x^{(i)}, z^{(i)})\}_{i=1}^{D}$ or $\{(y^{(i)}, z^{(i)})\}_{i=1}^{D}$ for learning the inference in the compressed domain.

Let f_θ denote the desired neural network parameterized by θ whose input is a compressed sample y and the output could be a detection, classification, or segmentation result, which is denoted by s. To learn the network using training samples, we set a loss function $L(s, z)$ which is a differentiable surrogate of the error probability. For example, in the classification problem, the cross entropy loss is commonly used [45]. The cross entropy loss requires two inputs, a C-dimensional vector s of the predicted probability scores and the groundtruth class label $z^* \in [C]$. We use one-hot encoding to transform the groundtruth class label z^* to another C-dimension vector $s^*(z^*)$ whose only non-zero element is z^*th element

with value 1.

$$\text{CELoss}(s, z^*) \triangleq \sum_{c=1}^{C} (s^*(z^*))_c \log(s_c) \tag{2.14}$$

Letting $\widehat{\theta}$ denote the parameters of f trained using D samples

$$\widehat{\theta} = \arg\min_{\theta} \sum_{i=1}^{D} L(f_\theta(y^{(i)}), z^{(i)}) \tag{2.15}$$

the decision rule of the DNN is then:

$$\widehat{z}(y; \widehat{\theta}) = \arg\max_{i=1,\ldots,C} (f_{\widehat{\theta}}(y))_i \tag{2.16}$$

The evaluation metrics of the neural networks, error probability, pixel-wise accuracy, and mIOU are defined as follows:

$$\begin{aligned} P_e &\triangleq E_{(y,z)\sim(\mathcal{Y},\mathcal{Z})}[\mathbb{1}\{\widehat{z}(y;\widehat{\theta}) \neq z\}] \\ \text{Pixel-wise Accuracy} &\triangleq E_{(y,z)\sim(\mathcal{Y},\mathcal{Z})} \frac{|z == \widehat{z}(y;\widehat{\theta})|}{N^2} \\ \text{mIOU} &\triangleq E_{(y,z)\sim(\mathcal{Y},\mathcal{Z})} \left[\text{mIOU}(z, \widehat{z}(y;\widehat{\theta}))\right] \end{aligned} \tag{2.17}$$

Here, $\mathbb{1}\{\cdot\}$ is the indicator function and $\mathbb{1}\{s\} = \begin{cases} 1, \text{ if } s \text{ is true,} \\ 0 \text{ else.} \end{cases}$. The pixel-wise accuracy and mIOU metrics are the expectations of the quantities defined in Eq. (2.11) and Eq. (2.13), respectively. For actual evaluation of performance, these expected values are replaced by empirical means over test samples.

2.5 Optimization of the Sensing Matrix in a DNN

In the optical CS systems, we manipulate the sensing matrix A by configuring the optical elements in the system, such as altering the binary pattern in the DMDs. Rather than define a separate optimization problem to optimize the sensing matrix, one can treat A as a parameter in the neural network. We can define a differentiable loss L, and the training

target becomes

$$\widehat{\theta}, \widehat{A} = \arg\min_{\theta, A \in \mathcal{A}} \sum_{n=1}^{N} L(f_\theta(Ax^{(i)}), z^{(i)}) \tag{2.18}$$

where $\mathcal{A}$ is the set of sensing matrices that satisfy the constraints imposed by the optical system. The decision rule of the DNN in the compressed domain is then:

$$\widehat{z}(\widehat{A}x; \widehat{\theta}, \widehat{A}) = \arg\max_{i=1,\ldots,C} (f_{\widehat{\theta}}(\widehat{A}x))_i \tag{2.19}$$

For the reasons stated in Section 2.2, we instead parameterize, as shown in (2.2), the sensing matrix A by a vector w whose dimension is the same as the number of degrees of freedom of $\mathcal{A}$, and optimize over w.

$$\widehat{w}, \widehat{\theta} = \arg\min_{w, \theta} \sum_{i=1}^{D} L(f_\theta(A(w)x^{(i)}), z^{(i)}) \tag{2.20}$$

The corresponding evaluation metrics of the DNN with the tunable sensing matrix are defined as follows.

$$\begin{aligned} P_e &\triangleq E_{(x,z)\sim(\mathcal{X},\mathcal{Z})}[\widehat{z}(A(\widehat{w})x; \widehat{\theta}, A(\widehat{w})) \neq z] \\ \text{Pixel-wise Accuracy} &\triangleq E_{(x,z)\sim(\mathcal{X},\mathcal{Z})} \frac{|z == \widehat{z}(A(\widehat{w})x; \widehat{\theta}, A(\widehat{w}))|}{N^2} \\ \text{mIOU} &\triangleq E_{(x,z)\sim(\mathcal{X},\mathcal{Z})} \left[\text{mIOU}(z, \widehat{z}(A(\widehat{w})x; \widehat{\theta}, A(\widehat{w}))\right] \end{aligned} \tag{2.21}$$

2.6 Research Questions

We address the following research for the problem formulations presented in this chapter.

1. How well does a learned neural network perform compared to the optimal decision rule in terms of the performance metrics mentioned above?

2. How well does a DNN with tunable sensing matrix A learn the sensing matrix? How will the learned A compare to the optimal acquisition scheme?

3. What are suitable neural network architectures for inference in the compressed domain with a limited number of training samples?

CHAPTER 3

COMPRESSED SENSING MULTISPECTRAL IMAGING SYSTEMS

3.1 Overview

This chapter is a brief survey of optical compressed sensing imaging (CSI) systems, focusing on multispectral imaging. We describe several such systems that are studied in this thesis, and provide the sensing matrix A describing the mapping from the data x to the compressed measurement y.

The single pixel camera [9] is one of the most famous and earliest CSI systems. A single pixel camera consists of a modulation device, digital micromirror device (DMD), two relay lenses and a single photon detector (photodiode). The DMD chip has several hundred thousand microscopic mirrors arranged in a rectangular array. The mirrors can be individually rotated to an on or off state, and each tunable mirror modulates one image pixel. When a mirror is turned to the on state, light from the object is reflected into the lens making the pixel appear bright. In the off state, the light is directed elsewhere, making the pixel appear dark. This DMD can also produce grayscale states by pulse width modulation. The single pixel camera utilizes one DMD to modulate the image and uses one focusing lens to sum up the energy in all pixels so that the photodiode will receive one reading, which we treat as one snapshot. The reading corresponds to the inner product of the input image x with the DMD mask. One can alter the pixel's on/off state in the DMD to obtain multiple photodiode readings. By taking K shots, one obtains a K-dimensional measurement.

Successful reconstruction from single pixel camera's measurement [9] can be obtained using l_1-minimization mentioned in Chapter 2. Because the DMD has no spectral selectivity, this system requires a spectrometer instead of a single pixel detector to obtain spectral information. However, since this single-pixel camera only multiplexes the spatial information in

Table 3.1: Optical CS systems. DMD: digital micromirror device. CA: coded aperture.

Optical CS Systems	Measurement Dimension	Hardware Requirements	Task	Optimized Acquisition
Single pixel camera (SPC) [9, 24]	$K \times B$	DMD, single point spectrometer	Recon [9] / Classify [24]	×/✓
SD CASSI [21, 47]	$(H+B-1) \times W \times K$	DMD, dispersive elements, CCD camera	Recon	×
Colored CASSI [20]	$(H+B-1) \times W \times K$	colored CA, dispersive elements, CCD camera	Recon	✓
DD CASSI [48, 49]	$H \times W \times K$	DMD, dispersive elements, CCD camera	Recon	×
Band selection	$H \times W \times K$	filter-based spectrometer	Classify	✓
Line selection	$(HK/B) \times W \times B$	filter-based spectrometer	Classify	✓

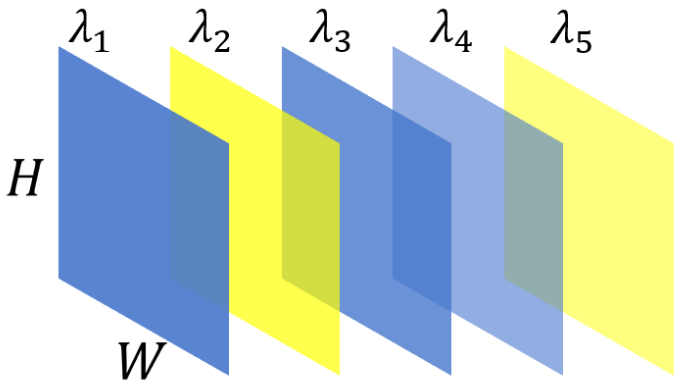
time but keeps the entire spectral domain uncompressed, such a system would require more snapshots, and therefore longer acquisition, to capture the image than a system that could multiplexes both the spatial and spectral domains.

We introduce several single shot CSI systems that overcome the limitations of the single pixel camera in the following section, with a particular focus on snapshot multispectral capture, which means that the spectral data are measured in a single exposure (shot) on the camera sensors [46].

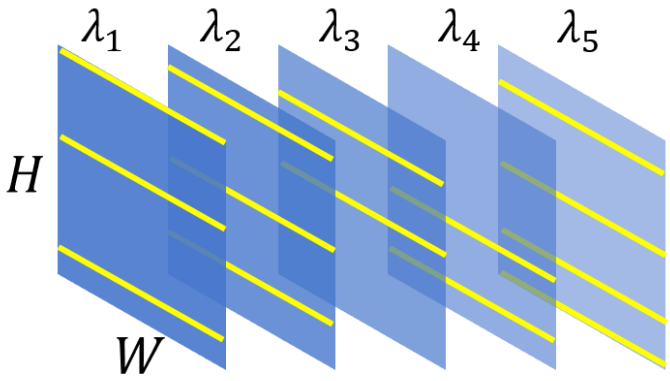
Now consider a multispectral $H \times W \times B$ image cube, where B is the number of spectral bands, and its sensing using various CSI schemes. Table 3.1 summarizes different sensing schemes and provides the measurement dimension, given $K \ll B$ physical image snapshots.

3.2 Line Selection and Band Selection in Full Sampling Systems

One approach to compressed sensing is a variation on the full-sampling scheme. There are three conventional full-sampling multispectral cameras [46]: (1) filter-based spectrometers, (2) scanning spectrometers, and (3) interferometry-based methods. The filter-based spectrometer obtains a full



(a) Band selection diagram



(b) Line selection diagram

Figure 3.1: Example of band selection and line selection in full-sampling CSI systems. Each 2D slice represents an $H \times W$ image in one spectral band. (a) Yellow or blue slices represent scanned or omitted bands. (b) Yellow or blue lines represent scanned or omitted lines.

spatial resolution (raster-scanned) image using a color filter or monochromatic light at one time and repeats this process for K different wavelengths. The scanning spectrometer captures a spectrum of a single spatial location at a time, and repeats for all HW spatial locations. The interferometry-based method (also known as Fourier transform spectral imaging) shines a beam containing multiple frequencies of light at once to obtain one full-resolution image, and repeats this process with K beams consisting of different frequency combinations.

Filter-based spectrometers and scanning spectrometers sense the 3D image cube in a parallel fashion, i.e. one can choose which optical frequency to scan or not in filter-based spectrometers without affecting the sensing of

other frequencies. In contrast, the interferometry-based methods require obtaining all the raw data before starting to process the information. This thesis does not include CSI systems based on such methods.

For example, one simple CSI scheme called "band selection", illustrated in Figure 3.1a, is to select for scanning a subset of the spectral bands and ignore the rest. Intuitively, if the image cube has strong correlation between different spectral bands or the information in different bands is redundant, then skipping several spectral bands will retain most of the information. Previous work using this scheme used mutual information or clustering [50–52] to select the bands.

Another scheme, called "line selection" [53], is illustrated in Figure 3.1b. This scheme can also be implemented using a filter-based spectrometer. For each spectral band, we select certain lines to scan. This scheme applies best to images that have both spatial and spectral correlations. The advantage of the band selection and the line selection schemes is that they do not require additional hardware to modulate the image.

The mathematical formulation of these selection schemes is straightforward. Let $\mathbf{x}_{vec} = [\vec{\mathbf{x}_1}, \ldots, \vec{\mathbf{x}_B}]^T \in \mathbb{R}^{B \times HW}$ denote the flattened multispectral image cube, where vector $\vec{\mathbf{x}_b} \in \mathbb{R}^{HW}$ represents the vectorized version of the $W \times H$ image slice in the b-th spectral band. Then the measurement in the band selection scheme is simply

$$\mathbf{y}_{BS} = P_{\Omega_K} \mathbf{x}_{vec} \in \mathbb{R}^{K \times HW} \tag{3.1}$$

where Ω_K is the set of indices of scanned bands and $P_{\Omega_K} \in \mathbb{R}^{K \times B}$ is a submatrix of the $B \times B$ identity consisting of the rows indexed by Ω_K.

For the line selection scheme, we use $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_B^T]^T \in \mathbb{R}^{BH \times W}$ to denote the multispectral image cube, where matrix $\mathbf{x}_b \in \mathbb{R}^{H \times W}$ represents the image slice corresponding to the b-th spectral band. The measurement can be written as:

$$\begin{aligned} \mathbf{y}_{LS} &= A\mathbf{x} \in \mathbb{R}^{(\sum_{i=1}^{B} |\Omega_i|) \times W} \\ \text{where } A &: \mathbb{R}^{BH \times W} \to \mathbb{R}^{(\sum_{i=1}^{B} |\Omega_i|) \times W} \\ A &= \operatorname{diag}([P_{\Omega_1}, P_{\Omega_2}, \ldots, P_{\Omega_B}]) \triangleq \begin{bmatrix} P_{\Omega_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & P_{\Omega_B} \end{bmatrix} \end{aligned} \tag{3.2}$$

where Ω_i is the set of indices of scanned lines in the i-th band, $P_{\Omega_i} \in \mathbb{R}^{|\Omega_i| \times H}$ is the submatrix of the $H \times H$ identity consisting of the rows indexed by Ω_i, and diag represents a block diagonalization operator defined above.

A variation of the line selection scheme is to scan both selected rows and columns. This variation could help if information in the image is equally correlated in both vertical and horizontal directions. To implement, this scheme requires the hardware to scan the lines in both directions, which is often possible at no increased cost - for example with a mirror galvanometer laser beam scanner.

3.3 Dual Dispersive Coded Aperture Compressive Spectral Imaging (DD-CASSI)

The DD-CASSI system [7] consists of two dispersive optical elements such as triangular prisms, a coded aperture (CA), and a focal plane array (FPA) detector such as a CCD camera. This scheme requires two dispersers placed symmetrically on the two sides of the CA.

Figure 3.2 illustrates the DD-CASSI system [46]. For better visualization, the 3D spectral image cube (x, y, λ) is shown using a 2D matrix representing both the spatial (x) domain and the spectral (λ) domain. A 3D image cube is firstly sheared by the dispersive element, which means that all spectral bands are translated in the x direction with neighboring spectral bands' translation differing by a distance of 1 pixel. Then the oblique image cube is modulated by a DMD CA of size $(H + B - 1) \times W$. The second disperser reverses the shearing and produces an unsheared spectral cube with a replicated slanted code. In the end the FPA records the sum of energies of B spectral bands, obtaining the final "projection" of size $H \times W$ as one snapshot.

With K different DMD patterns, the FPA records K different snapshots and we obtain the final measurement of size $K \times H \times W$. This CS scheme ensures that, at the cost of two dispersers and calibration, the final projection captures a partial spectrum of every spatial pixel in each snapshot.

For a mathematical formulation, let $\mathbf{x}_{vec} = [\vec{\mathbf{x}_1}^T, \ldots, \vec{\mathbf{x}_B}^T]^T \in \mathbb{R}^{B \times HW}$ denote the flattened multispectral image cube, where, as in the band

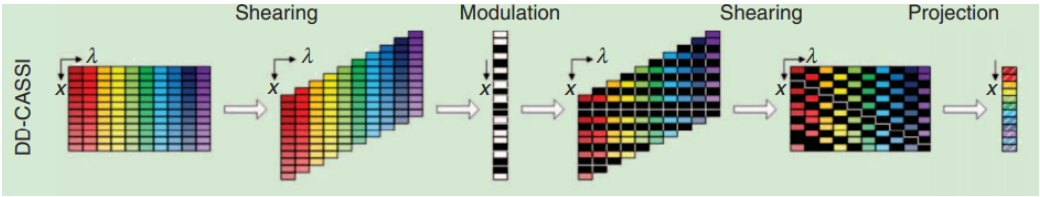


Figure 3.2: Spatial-spectral CASSI illustration.

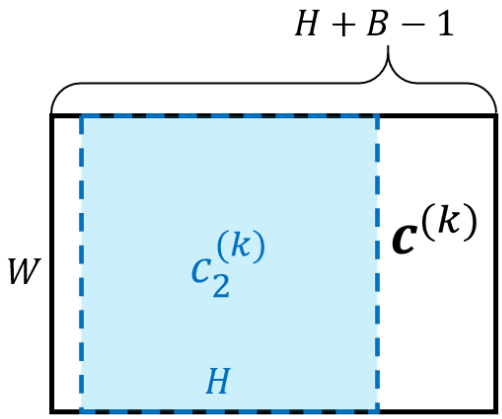


Figure 3.3: DMD pattern $\mathbf{c}^{(k)}$ (black) and DMD pattern for 2nd spectral band, $\mathbf{c}_2^{(k)}$ (blue).

selection scheme, vector $\vec{\mathbf{x}}_b \in \mathbb{R}^{HW}$ represents the vectorized b-th spectral band image slice. For the k-th snapshot, let $\mathbf{c}^{(k)} \in \{0,1\}^{(H+B-1)W}$ denote the DMD code, which is the vectorized version of the k-th 2D DMD pattern (the black solid line rectangle in Figure 3.3), and let
$Q_b : \{0,1\}^{(H+B-1)W} \to \{0,1\}^{HW}$ denote the b-th window extractor of the vectorized DMD code. That is, Q_b extracts the sub-vector with components corresponding to an $H \times W$ window from the $(H+B-1) \times W$ vectorized 2D DMD code. For example, in Figure 3.3, Q_b extracts the sub-vector with components corresponding to the blue dotted line rectangle from the vectorized version of the black solid line rectangle. Then vector

$$\mathbf{c}_b^{(k)} := Q_b \mathbf{c}^{(k)} = \mathbf{c}^{(k)}[b : b+H-1] \in \{0,1\}^{H \times W} \tag{3.3}$$

is a vectorized version of a sliding window of size $H \times W$ (the blue -shaded rectangle with broken line outline in Figure 3.3), which encodes the image's b-th spectral band. Using this notation, we can write the measurement

$\mathbf{y}_k \in \mathbb{R}^{HW}$ as:

$$\begin{aligned}\mathbf{y}_k &= \sum_{b=1}^{B} \mathbf{c}_b^{(k)} \odot \vec{\mathbf{x}_b} = \sum_{b=1}^{B} (Q_b \mathbf{c}^{(k)}) \odot \vec{\mathbf{x}_b} \\ &= [\mathbf{I}, \ldots, \mathbf{I}] \left[((Q_1 \mathbf{c}^{(k)}) \odot \vec{\mathbf{x}_1})^T, \ldots, ((Q_B \mathbf{c}^{(k)}) \odot \vec{\mathbf{x}_B})^T \right]^T \\ &= (\mathbf{1}_B^T \otimes \mathbf{I}) \text{diag}([(Q_1 \mathbf{c}^{(k)})^T, \ldots, (Q_B \mathbf{c}^{(k)})^T]^T) \mathbf{x}_{vec}\end{aligned} \tag{3.4}$$

where $\odot$ denotes the Schur-Hadamard product, or element-by-element product of vectors or matrices of the same dimension, and $\otimes$ denotes the Kronecker product.

The final measurement $\mathbf{y}$ using K different DMD patterns can be expressed as

$$\begin{aligned}\mathbf{y} &= A\mathbf{x}_{vec} \in \mathbb{R}^{K \times HW} \\ \text{where } A &: \mathbb{R}^{B \times HW} \to \mathbb{R}^{K \times HW} \\ A &= \begin{bmatrix} (\mathbf{1}_B^T \otimes \mathbf{I}) \text{diag}([(\mathbf{c}_1^{(1)})^T, \ldots, (\mathbf{c}_B^{(1)})^T]^T) \\ \cdots \\ (\mathbf{1}_B^T \otimes \mathbf{I}) \text{diag}([(\mathbf{c}_1^{(K)})^T, \ldots, (\mathbf{c}_B^{(K)})^T]^T) \end{bmatrix}\end{aligned} \tag{3.5}$$

A variation on this DD-CASSI is called the single disperser CASSI (SD-CASSI) [49], which only has one disperser and a coded aperture of smaller size. The SD-CASSI produces an oblique image cube and each snapshot has size $(H + B - 1) \times W$.

3.4 Optimization of the Acquisition Scheme

As discussed in Section 2.5, in optical systems the sensing matrix A is not entirely adjustable. One metric to evaluate the complexity of a compressed sensing systems is the number of degrees of freedom (DOF) of the acquisition scheme, which refers to the number of independent parameters that define its configuration. In general, the greater the number of degrees of freedom, the better the performance of a scheme. From this perspective, the number of DOF of the band selection scheme is only B while the line selection scheme has BH DOF. For DD-CASSI, the number of DOF for K snapshots is $KW(H + B - 1)$.

While much of the compressed sensing literature advocates and analyzes the use of randomized acquisition schemes, in which the variables controlling the DOF are chosen independently at random from some distribution, there are a few works that optimize the optical acquisition scheme for a better reconstruction performance. Some approaches use a surrogate metric such as the restricted isometry property (RIP) [20, 25]. A smaller RIP constant δ of a sensing matrix implies the inverse problem is more stable and it is easier to reconstruct the image from scarce measurements. Another work [26] optimizes the optical element jointly with a neural network-based reconstructor to achieve end-to-end optimization. However, we are not aware of any work to date that optimizes the optical sensing matrix for better inference performance. This is one of the contributions of this thesis.

CHAPTER 4

APPROACH AND ANALYSIS

In this chapter, we describe the solutions to the problems stated in Chapter 2. First in Section 4.1 we present the optimal decision rule that minimizes the error probability for compressed detection, classification, and segmentation in settings in which the data statistics are fully known. We give lower and upper bounds on error probability for the tasks of compressed detection and classification when the sensing matrix is a random matrix. Then in Section 4.2, we explain the implementation of the optimal decision rule for different optical sensing schemes. Next, in Section 4.3, a DNN framework is proposed for compressed inference. In the end, we present the detailed structure of the DNN, and an approach to the optimization of the sensing matrix by embedding its model in the DNN.

A random Gaussian matrix A of size $m \times n, m \leq n, A_{ij} \sim \mathcal{N}(0, \frac{1}{m})$ is a common sensing matrix in the compressed sensing literature [11, 12]. The Gaussian random matrix is popular because it is easy to generate and amenable to theoretical analysis. With high probability, it asymptotically satisfies restricted isometry property (RIP) with small RIP constant for sparse signals [54], which is a desirable property of a sensing matrix for exact recovery from measurements. Therefore it is useful as a benchmark to assess the overall system performance, as is also done in this chapter. However, such random Gaussian matrices often do not satisfy the constraints of the CSI systems that are considered in Chapter 3. Instead, we wish to find a sensing matrix A tailored to a particular class of signals that satisfies the optical system constraints.

4.1 Compressed Inference in a Known Data Statistics Setting and the Optimal Decision Rule

In general, we do not have the full statistics of the data in real detection, classification, or segmentation problems. Thus it is hard to evaluate the performance of a specific algorithm for these tasks. In order to examine the capability of a specific neural network in these inference problems, we propose simple data settings in which we discuss the optimal decision rule with full statistics of the data.

4.1.1 Detection in the compressed domain

Detection with unknown prior probabilities

Consider the hypotheses in (2.3), when the prior probabilities of the hypotheses are unknown. The Neyman-Pearson detector maximizes the detection rate P_D when the false alarm rate P_F is below a given level γ. This scenario is analyzed by Davenport et al. [29, 55]. The decision rule for the accepted hypothesis is a likelihood ratio test:

$$\Lambda(y) = \frac{p(y \mid \mathcal{H}_1)}{p(y \mid \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta \tag{4.1}$$

$$\begin{aligned} p\left(y|H_1\right) &= \frac{\exp(-(y-Ax)^T(\sigma^2 AA^T)^{-1}(y-Ax)/2)}{(2\pi\sigma^2)^{m/2}(|AA^T|)^{1/2}} \\ p\left(y|H_0\right) &= \frac{\exp(-y^T(\sigma^2 AA^T)^{-1}y/2)}{(2\pi\sigma^2)^{m/2}(|AA^T|)^{1/2}} \end{aligned} \tag{4.2}$$

where the threshold η is chosen such that

$$P_F = \int_{\Lambda(y)>\eta} p(y \mid \mathcal{H}_0)dy = \gamma \tag{4.3}$$

The decision rule simplifies to

$$x^T A^R y \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\|P_{A^T}x\|^2}{2} + \sigma^2 \log(\eta) \tag{4.4}$$

where $A^R \triangleq A^T(AA^T)^{-1}$ is the right inverse of A and $P_{A^T} \triangleq A^R A$ is the orthonormal projection matrix onto the row space of A.

Given the false alarm rate γ, the optimal detection rate in the compressed domain is [29, 55]:

$$P_D(\gamma) = Q\left(Q^{-1}(\gamma) - \frac{\|P_{A^T}x\|_2^2}{\sigma}\right) \tag{4.5}$$

where Q is the Q-function of standard Gaussian tail distribution.

Optimal Lossless Sensing Matrix

An optimal choice of $A \in R^{m\times n}$, for any $m \geq 1$, is one that satisfies range$(A^T) \supset x$. With this choice, we have $P_{A^T}x = x$, and the error probability (4.4) reduces to that of the uncompressed setting with $A = I$. That such lossless compressive detection is possible is, of course, to be expected, because the optimum detector for a known signal x in white noise uses the scalar sufficient statistic $x^T y$.

The limitation of the lossless choice of A described here is that it requires substantial knowledge about the signal x - in the form of an m-dimensional subspace that contains it.

Random Sensing Matrix

The other extreme of a sensing matrix is a random matrix, which is universal in the sense that its performance is, with high probability, the same for any signal x.

Suppose that the sensing matrix A is a random matrix, such that P_{A^T} is a random ortho-projector. Possible choices of A that satisfy this include a random Gaussian matrix, a random matrix with $A_{ij} \in \{-1, 0, +1\}$ [56], and various other random matrices [57]. The Johnson-Lindenstrauss lemma [56, 58] states the following:

Theorem 1 (Johnson-Lindenstrauss lemma)**.** *For arbitrary set $\mathcal{S}$ of C points, given constants $0 < \epsilon < 1, \beta > 0$, let*

$$m \geq \frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \ln(C) \tag{4.6}$$

a random orthonormal projector P_{A^T} satisfies the following with probability at least $1 - C^{-\beta}$:

$$(1-\epsilon)\sqrt{m/n} \leq \frac{\|P_{A^T}(u-v)\|_2}{\|u-v\|_2} \leq (1+\epsilon)\sqrt{m/n}, \text{ for all } u, v \in \mathcal{S} \tag{4.7}$$

The compressed detection is only concerned with two points, $\mathbf{0}$ and x.

We apply the JL lemma with $|\mathcal{S}| = C = 2, \zeta = 2^{-\beta}$. Supposing

$$m \geq \frac{4\ln(2) - 2\ln(\zeta)}{\epsilon^2/2 - \epsilon^3/3} \tag{4.8}$$

then with probability $1-\zeta$, a random orthoprojector P_{A^T} satisfies (4.7) and the term $\|P_{A^T}x\|_2^2$ concentrates around $m\|x\|^2/n$. The detection rate is

$$P_D(\gamma) \approx Q\left(Q^{-1}(\gamma) - \sqrt{SNR_m}\right) \tag{4.9}$$

where the signal-to-noise ratio in the compressed domain is

$$\text{SNR}_m \triangleq \frac{m}{n}\text{SNR} \tag{4.10}$$

and the SNR in the data domain is

$$\text{SNR} \triangleq \frac{\|x\|_2^2}{\sigma^2} \tag{4.11}$$

This detection rate is determined by the SNR and by the compression ratio n/m. In general a lower compression ratio and a higher SNR would yield a higher detection rate.

This result [29, 55] appears counter-intuitive from the perspective of compressed sensing. Suppose that the signal x is sparse at some fixed sparsity level s. Then CS theory tells us that when the number of measurements m is large enough compared to s, one can recover with high probability the exact signal from its measurement. Consider therefore an *ad-hoc detector* that first recovers the sparse signal x, and then performs the detection in the data domain. One expects the detection of this detector to be the same as for the uncompressed signal in noise, $x + v$, and not to depend on SNR reduced by the compression factor m/n.

On the other hand, the detection performance of the optimal detector using the compressed signal $y = A(x + v)$ cannot be worse than that of the *ad-hoc detector*. Therefore the factor of m/n SNR loss seen in (4.9) – (4.11) appears paradoxical.

The resolution of this apparent paradox is that the above ad-hoc detection scheme too suffers from reduced SNR, and cannot beat the optimum compressive detector. To see this, consider first what happens to the energy of any fixed signal x under the action of a random matrix

$A \in R^{m\times n}$ with orthonormal rows. (This is a good approximation to A with elements $\overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/n)$). This is equivalent to projection by a random orthoprojector, so that with high probability, $\|Ax\|^2$ concentrates around $(m/n)\|x\|_2^2$, representing a reduction of the energy of the signal by factor m/n.

Consider next what happens to the variance of the noise $v \sim N(0, \sigma^2 I_n)$ under the action of A. The covariance of Av is $\sigma^2 AA^T = \sigma^2 I_m$. Hence, the noise variance per component remains unchanged, whereas the energy of the signal is reduced by factor m/n. This fact (mentioned also in [55]) results in an inherent SNR loss by the factor m/n, and explains the appearance of this factor in all the expressions for the performance of compressive inference schemes. Because this SNR loss is inherent to the compressive scheme, it cannot, of course, be overcome by any ad-hoc scheme, including the one involving first recovering the signal.

Detection with known prior probabilities

This case is a simple extension of the case considered by Davenport et al. [29]. Suppose the priors of the null and alternative hypotheses are π_0 and π_1, respectively. The maximum a posteriori (MAP) detector yields the lowest error probability [43]. Its decision rule is a likelihood ratio test:

$$\frac{p\left(y|H_1\right)}{p\left(y|H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{p\left(H_0\right)}{p\left(H_1\right)} = \frac{\pi_0}{\pi_1} \tag{4.12}$$

It follows (see Appendix A.3) that the decision rule (4.12) can be expressed as

$$x^T A^R y \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\|P_{A^T} x\|^2}{2} + \sigma^2 \log(\frac{\pi_0}{\pi_1}) \tag{4.13}$$

We can compute the exact error probability for a given sensing matrix A:

$$\begin{aligned} P_e &= \pi_0 P(H_1 \text{ accepted} \mid H_0) + \pi_1 P(H_0 \text{ accepted} \mid H_1) \\ &= \pi_0 Q\left(\frac{\|P_{A^T} x\|_2}{2\sigma} + \frac{\sigma \log(\pi_0/\pi_1)}{\|P_{A^T} x\|_2}\right) \\ &+ \pi_1 Q\left(\frac{\|P_{A^T} x\|_2}{2\sigma} - \frac{\sigma \log(\pi_0/\pi_1)}{\|P_{A^T} x\|_2}\right) \end{aligned} \tag{4.14}$$

For orthonormal projectors P_{A^T} that satisfy (4.7), it follows that the

error probability has the upper bound:

$$P_e \leq \pi_0 \exp\left(-\left(\frac{(1-\epsilon)^2 \mathrm{SNR}_m}{8} + \frac{\log^2(\pi_0/\pi_1)}{2(1+\epsilon)^2 \mathrm{SNR}_m} + \frac{(1-\epsilon)\log(\pi_0/\pi_1)}{2(1+\epsilon)}\right)\right)$$
$$+ \pi_1 \exp\left(-\left(\frac{(1-\epsilon)^2 \mathrm{SNR}_m}{8} + \frac{\log^2(\pi_0/\pi_1)}{2(1-\epsilon)^2 \mathrm{SNR}_m} - \frac{\log(\pi_0/\pi_1)}{2}\right)\right) \tag{4.15}$$

The detailed derivation of upper bound can be found in Appendix A.3, as a simple extension to the Neyman-Pearson detector [29]. Under the uniform prior assumption ($\pi_0 = \pi_1 = 0.5$), (4.14) is shown in Appendix A.3 to yield the bounds

$$\exp\left(-\frac{(1+\epsilon)^2 \mathrm{SNR}_m}{4}\right) \leq P_e \leq \exp\left(-\frac{(1-\epsilon)^2 \mathrm{SNR}_m}{8}\right) \tag{4.16}$$

The bounds become tighter with decreasing ϵ, in which case P_e is determined by the SNR and the compression ratio of data over measurement dimension n/m. Similar to the Neyman-Pearson case, the effective SNR is reduced by the compression ratio n/m. Higher SNR and lower compression ratio n/m yield lower error probability, decaying at an exponential rate. Thanks to this exponential dependence, with sufficiently high SNR, the SNR loss by the compression factor n/m will still provide sufficiently small error probability for many applications.

Detection with composite hypotheses

Another compressed sensing problem mentioned in Section 2.3.1 considers detecting an object with unknown position and it is described in (2.6). For simplicity, we assume the prior probability of each hypothesis is the same, $\pi_0 = \pi_1 = 1/2$.

Again, the MAP decision rule minimizes the error probability. Under the

uniform prior, it reduces to the ML rule.

$$
\begin{aligned}
c^* &= \arg\max_{c\in\{0,1\}} P(H_c)P(y \mid H_c) = \arg\max_{c\in[C]} P(y \mid H_c) \\
&= \arg\max_{c\in\{0,1\}} \sum_{x\in\Omega_c} P(y, x \mid H_c) \\
&= \arg\max_{c\in\{0,1\}} \sum_{x\in\Omega_c} P(y \mid H_c, x)P(x \mid H_c) \\
&= \arg\max_{c\in\{0,1\}} P(y \mid H_c) = \arg\max_{c\in\{0,1\}} \sum_{x\in\Omega_c} P(y \mid H_c, x)P(x \mid H_c) \\
&= \arg\min_{c\in\{0,1\}} \sum_{x\in\Omega_c} \exp\left(-\frac{(y-Ax)^T(AA^T)^{-1}(y-Ax)}{2\sigma^2} + \log(P(x \mid H_c))\right)
\end{aligned}
\tag{4.17}
$$

For the detection in the data domain without compression, $A = I$, the optimal decision rule reduces to:

$$
c^* = \arg\min_{c\in\{0,1\}} \sum_{x\in\Omega_c} \exp\left(-\frac{\|y-x\|^2}{2\sigma^2} + \log(P(x \mid H_c))\right) \tag{4.18}
$$

We do not provide an upper bound of the error probability here due to its complexity. Furthermore, in the case of a designed sensing matrix A, an upper bound on the error probability in terms of the constant ϵ in (4.7) would be of limited utility. This is because, unlike the case of some random matrices, computing ϵ for an arbitrary fixed matrix A for a large set of candidate x is expensive or even intractable. Instead, we estimate the error probability using Monte Carlo simulation as described next.

Estimating the error probability by Monte Carlo simulation.

We generate independent data samples for different noise levels σ under hypotheses H_cs and use the optimal decision rule (4.17) to compute the empirical error probability. With a large enough number of samples, the empirical error probability provides a good approximation for the true error probability because the empirical P_e converges to the true P_e almost surely.

Note that in computation, the exponential terms in (4.17) may encounter numerical underflow if σ is very small. The Exp-normalize trick (also known as LogSumExp trick) avoids the numerical underflow and helps to compute the exact likelihood. For example, in (4.17), this trick computes the quantity $d \triangleq \min_{x\in\Omega_c} \|y-x\|^2/2\sigma^2$ and subsequently computes the quantity $\exp(-d)\sum_{x\in\Omega_c} \exp\left(-\left(\frac{\|y-x\|^2}{2\sigma^2} - d\right) + \log(P(x \mid H_c))\right)$ to

circumvent the numerical overflow or underflow.

4.1.2 Classification in the compressed domain

Classification with known prior probabilities

Similar to detection in the compressed domain, we denote the prior probability of each class by $\pi_i, i = 1, \ldots, C$, and for each class the mean of the signal is $x_i, i = 1, \ldots, C$. Consider the hypotheses (2.7). The maximum a posterior probability (MAP) decision rule yields the lowest error probability, or the misclassification rate. For a specific sensing matrix A and associated measurement $y = Ax$, the optimal decision rule is:

$$\begin{aligned} i^* &= \arg\max_{i=1,\ldots,C} P(H_i \mid y) \\ &= \arg\max_{i=1,\ldots,C} \pi_i P(y \mid H_i) \\ &= \arg\min_{i=1,\ldots,C} (y - Ax_i)^T \left(AA^T\right)^{-1} (y - Ax_i) - 2\sigma^2 \log(\pi_i) \end{aligned} \tag{4.19}$$

When the classification is performed in the original data domain $A = I$, the decision rule reduces to

$$i^* = \arg\min_{i=1,\ldots,C} \|y - x_i\|^2 - 2\sigma^2 \log(\pi_i) \tag{4.20}$$

The error probability upper bound of the MAP decision rule derived in Appendix A.4 is:

$$P_e \leq \sum_{t=1}^{C} \pi_t \sum_{\ell \neq t} Q\left(-\frac{\|P_{A^T}(x_t - x_\ell)\|_2}{2\sigma} + \frac{\sigma \log\left(\frac{\pi_t}{\pi_\ell}\right)}{\|P_{A^T}(x_t - x_\ell)\|_2} \right) \tag{4.21}$$

This error bound is determined mainly by the noise level σ and the projected difference $\|P_{A^T}(x_t - x_\ell)\|_2$ between different signals x_t and x_ℓ. In general, a larger average projected difference and a smaller σ yield lower error probability, if we ignore the second term inside the Q-function. Analyzing the monotonicity of this upper bound is hard due to the term $\log(\pi_t/\pi_\ell)$. Under uniform prior probability scenario, the bound reduces to

$$P_e \leq \frac{1}{C} \sum_{t=1}^{C} \sum_{\ell \neq t} Q\left(-\frac{\|P_{A^T}(x_t - x_\ell)\|_2}{2\sigma} \right) \tag{4.22}$$

Classification with unknown prior probabilities

We utilize the maximum likelihood (ML) decision rule in this case. The ML likelihood is equivalent to the MAP rule when the prior probability of the class is all equal.

$$i^* = \arg\min_i (y - Ax_i)^T \left(AA^T\right)^{-1} (y - Ax_i) \tag{4.23}$$

The conditional misclassification rate P_e can be bounded using the union bound [29]:

$$\begin{aligned} P_e|_{H_t} &= 1 - \prod_{j \neq t} \left(1 - Q\left(\frac{\|P_{A^T}(x_t - x_j))\|_2}{2\sigma} \right) \right) \\ &\leq \sum_{j \neq t} Q\left(\frac{\|P_{A^T}(x_t - x_j))\|_2}{2\sigma} \right) \end{aligned} \tag{4.24}$$

where t is the index of the underlying true hypothesis. This upper bound coincides with (4.21) under the uniform prior probabilities.

We follow the derivation by Davenport et al. [29] but utilize the JL lemma (Theorem 1) instead of the RIP property to derive an upper bound on the conditional error probability. The JL lemma does not restrict the signals to be sparse while the RIP property does. For large enough

$$m \geq \frac{4\ln(C) - 2\ln(\zeta)}{\epsilon^2/2 - \epsilon^3/3} \tag{4.25}$$

a random orthoprojector P_{A^T} satisfies (4.7) with probability at least $1 - \zeta$ simultaneously for all points x_t and x_j, $t, j \in [C]$. The upper bound, which holds simultaneously for all $t \in [C]$, is then:

$$P_e|_{H_t} \leq \sum_{j \neq t} Q\left(\frac{(1-\epsilon)\|(x_t - x_j))\|_2 \sqrt{m}}{2\sigma\sqrt{n}} \right) \tag{4.26}$$

Defining $d \triangleq \min_{i,j} \|x_i - x_j\|$, we derive an upper bound on the (unconditional) error probability using the Chernoff bound on the

Q-function $Q(z) \leq \exp(-\frac{z^2}{2})$.

$$\begin{aligned} P_e &= \sum_t P(H_t) P_e|_{H_t} \\ &\leq \sum_t P(H_t) \sum_{j \neq t} Q\left(\frac{(1-\epsilon)d\sqrt{m}}{2\sigma\sqrt{n}}\right) \\ &= \sum_{j \neq t} Q\left(\frac{(1-\epsilon)d\sqrt{m}}{2\sigma\sqrt{n}}\right) \\ &\leq (C-1)\exp\left(-\frac{d^2(1-\epsilon)^2 m}{8\sigma^2 n}\right) \\ &= (C-1)\exp\left(-\frac{(1-\epsilon)^2 \mathrm{SNR}_m}{8}\right) \end{aligned} \tag{4.27}$$

where $\mathrm{SNR}_m = m/n\mathrm{SNR}, \mathrm{SNR} = d^2/\sigma^2$ in the classification case.

Similar to the detection case, a higher SNR $\frac{d^2}{\sigma^2}$ and a lower compression ratio $\frac{n}{m}$ yield a lower error probability. This bound is linear in the number of classes C. When $C = 2$, and we set $x_1 = 0, x_2 = x, d^2 = \|x\|^2$, this bound reduces to the detection problem of signal 0 and x, as in (4.16).

This upper bound is useful when it is lower than $\frac{C-1}{C}$, the error probability of a random guess between C classes. If the number of measurements m is larger than $\frac{8\ln(C)}{(1-\epsilon)^2\mathrm{SNR}}$, then compressed classification is better than a random guess.

We use these optimal decision rules and Monte Carlo simulation to estimate the error probability P_e in compressed classification using arbitrary (not necessarily random) sensing matrices A.

4.1.3 Segmentation in the compressed domain

We consider a simple data setting and aim to minimize the error probability mentioned in Section 2.3.3: a square shift-invariant object of size $M_c \times M_c$ drawn at random from possible C classes that appears at a random position on a $N \times N$ background and is imaged for B spectral bands. The pixel values of the object are deterministic signals of its class and are known for each class, but the location of the object is unknown. This data setting is a simplified version of many image segmentation applications—for example, segmenting the cell types from a tumor biopsy

multispectral image, where the object of interest, such as tumor cells, could be round or of irregular shape and appearing in different locations, and the pixel values are correlated to each other within the same class.

The simple hypothesis test for this setting is formulated in (2.9) with $\sum_{c=1}^{C}(N - M_c + 1)^2$ hypotheses. For clarity, we use $x_{c,i}$ instead of x_i to denote the mean of the image that contains an object from cth class at the ith location Similar to the classification problem, the optimal decision rule that minimizes the error probability is:

$$c^*, i^* = \arg\min_{c,i} (y - Ax_{c,i})^T \left(AA^T\right)^{-1} (y - Ax_{c,i}) - 2\sigma^2 \log(\pi_{c,i}) \tag{4.28}$$

where $\pi_{c,i}$ denotes the prior probability of the cth object appearing in the ith location. Again, we use this optimal decision rule and Monte Carlo simulation to estimate the P_e, pixel-wise accuracy, and mean IOU.

4.2 Optimal Decision Rule in Practice

Recall the optimal decision rule for compressed segmentation (4.28). For the band selection and line selection CS scheme, we set the measurement as $A^T y = G(w) \odot x$ instead of y. This extra A^T does not affect the optimal decision rule, because the AA^T of the line or band selection schemes happens to be equal to the identity matrix. To see this, note that

$$\begin{aligned} \|A^T y - A^T A x_{c,i}\|_2^2 &= (y - Ax_{c,i})^T AA^T (y - Ax_{c,i}) \\ &= \|y - Ax_{c,i}\|_2^2 \\ &= (y - Ax_{c,i})^T (AA^T)^{-1} (y - Ax_{c,i}) \end{aligned} \tag{4.29}$$

We can then replace term $(y - Ax_{c,i})^T (AA^T)^{-1} (y - Ax_{c,i})$ in (4.19) by $\|A^T y - A^T A x_{c,i}\|_2^2$. Assuming a uniform prior of classes c and the locations i, we use

$$c^*, i^* = \arg\min_{c,i} \|A^T y - A^T A x_{c,i}\|_2^2 \tag{4.30}$$

as the optimal decision rule for compressed segmentation for line and band selection. This simplification allows further discussion of the suitable neural network architecture.

For the DD-CASSI scheme, we also use $A^T y$ as the measurement. But

computing $(y - Ax_{c,i})^T(AA^T)^{-1}(y - Ax_{c,i})$ with the whitening matrix $(AA^T)^{-1}$ can be tricky (see Appendix A.2). In our subsequent experiments, we replace the $(AA^T)^{-1}$ with the identity matrix as a suboptimal decision rule since it is easy to implement. We used (4.30) for DD-CASSI schemes as well.

4.3 Neural Network for Compressed Inference

In Chapter 2, we briefly discuss training a DNN for inference in the compressed domain and in Section 2.5 we suggest to optimize the sensing matrix within the DNN. However, choice of a suitable network architecture for compressed classification or segmentation was not addressed. In this section, we present the details of the DNN, using a compressed segmentation network as an example.

Figure 4.1 shows the diagram of the compressed segmentation network. This network consists of two parts: a linear CS operator denoted as A defines the CS scheme and a (deep) neural network denoted by f_θ maps the compressed signal to the corresponding class label denoted by z.

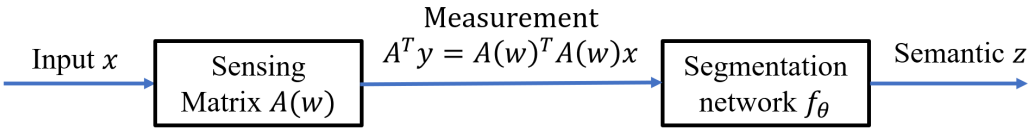


Figure 4.1: Compressed segmentation network diagram.

In Section 2.5, the manipulation of the sensing matrix A through parameterization $A(w)$ is mentioned. In addition to the loss function in (2.20), we need a penalty term, a function of the parameter w, to control the number of measurements m of the sensing matrix $A(w)$. The exact form of the penalty term depends on the CSI scheme.

4.4 Loss Function

4.4.1 Line and band selection scheme

In the band selection scheme introduced in Section 3.2, the number of DOF is $d = B$ and the parameter $w \in \mathbb{R}^B$. We can write $A^T Ax = G(w) \odot x$ where x is the image cube, and $G(w)$ is a binary mask that has the same size as x. In $G(w)$ 1 or 0 indicate scanned or skipped voxels, respectively, and $G(w) = \mathbb{1}\{w > 0\} \bigotimes \mathbf{1}\mathbf{1}^T \in \mathbb{R}^{B \times H \times W}$. Using the measurement $y = Ax$, the input to the DNN is $A^T y = A^T Ax = G(w) \odot x$.

In the line selection scheme, the number of DOF is HB and the parameter $w \in \mathbb{R}^{BH}$. The binary mask is $G(w) = \mathbb{1}\{w > 0\} \bigotimes \mathbf{1} \in \mathbb{R}^{BH \times W}$. The rest is the same as the band selection scheme.

Let $x^{(n)}, n \in [D]$ and $z^{(n)} \in [C]^{H \times W}$ denote the n-th 3D image sample out of a D-sample dataset and the corresponding segmentation label, respectively. The loss function combines the cross entropy loss, which is a common choice for image segmentation tasks [59, 60], and the cost of the number of scanned bands or lines as a penalty term.

$$\begin{aligned} L(\theta, w) &= \frac{1}{D} \sum_{n=1}^{D} \text{CELoss}(f_\theta(Ax^{(n)}), z^{(n)}) + \lambda_1 \|G(w)\|_0 \\ &= \frac{1}{D} \sum_{n=1}^{D} \text{CELoss}(f_\theta(y^{(n)}), z^{(n)}) + \lambda_1 \sum_{i,j,k} G_{i,j,k}(w) \end{aligned} \tag{4.31}$$

where CELoss denotes the sum of cross entropy loss for all pixels defined in (2.14), $\|G(w)\|_0$ counts the number of scanned lines/bands, and λ_1 is a hyperparameter that controls the sparsity of the binary mask $G(w)$.

Note that the number of scanned bands or lines is determined by the learned parameter w and it is hard to fix the number of non-zeros in $G(w)$ during the training process of the network. Hence we used a penalty term for this scheme. For DD-CASSI scheme (Section 4.4.2), we do not need a penalty term because the number of snapshots K is predetermined and the training alters the K DMD patterns only.

4.4.2 DD-CASSI scheme

For the DD-CASSI CS scheme, the number of DOF is $(H+B-1)WK$ for K snapshots, and the parameter $w \in \mathbb{R}^{(H+B-1)W\times K}$. The kth DMD pattern $\mathbf{c}^{(k)} \in \{0,1\}^{(H+B-1)W}$ mentioned in Section 3.3 is parameterized by w:

$$\mathbf{c}^{(k)} = \mathbb{1}\{w_k > 0\} \tag{4.32}$$

We can write the sensing matrix $A(w)$ using (3.5):

$$A(w) = \begin{bmatrix} (\mathbf{1}_B^T \otimes \mathbf{I})\text{diag}([(\mathbb{1}\{w_1 > 0\}_1)^T, \ldots, (\mathbb{1}\{w_1 > 0\}_B)^T]^T) \\ \cdots \\ (\mathbf{1}_B^T \otimes \mathbf{I})\text{diag}([(\mathbb{1}\{w_K > 0\}_1)^T, \ldots, (\mathbb{1}\{w_K > 0\}_B)^T]^T) \end{bmatrix} \tag{4.33}$$

Let $x^{(n)}, n \in [D]$ and $z^{(n)} \in [C]^{H\times W}$ denote the n-th 3D image sample out of a D-sample dataset and the corresponding segmentation label, respectively. The loss function for the DD-CASSI scheme is:

$$\begin{aligned} L(\theta, w) &= \frac{1}{D}\sum_{n=1}^{D} \text{CELoss}(f_\theta(y^{(n)}), z^{(n)}) \\ &= \frac{1}{D}\sum_{n=1}^{D} \text{CELoss}(f_\theta(A(w)x^{(n)}), z^{(n)}) \end{aligned} \tag{4.34}$$

where CELoss is cross-entropy loss between the predictions and labels.

4.4.3 Binarized layer

The optimization of the loss function with respect to w, the parameter of sensing matrix A, can be tricky because the DMD mask in DD-CASSI $^{(k)}$ or the selection mask $G(w)$ in the band/line selection scheme are binary-valued. One way to generate such binary values is to use the binarization function sign(w) [61], which is applied element-wise for a tensor input, yielding 1 for positive elements and 0 otherwise. The binary mask $G(w)$ is then generated from an intermediate binary parameter

$$w^b \triangleq \text{sign}(w) \tag{4.35}$$

This binarization function is easy to implement in the forward propagation. However, the derivative of the sign function is zero almost everywhere, making it apparently incompatible with back-propagation while training the network. We use the "straight-through estimator", which simply sets $\partial w^b / \partial w$ to 1 [62], treating the binary neuron during back propagation as an identity operator.

Given the gradient $\frac{\partial L}{\partial w^b}$ of the loss function L with respect to binary parameter w^b (obtained by standard back propagation), then the straight-through estimator gives the gradient of L with respect to the real-valued parameter w by

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w^b} \frac{\partial w^b}{\partial w} = \frac{\partial L}{\partial w^b} \tag{4.36}$$

This estimate of gradients with respect to binary parameters proved effective in binarized NN [26, 63].

Intuitively, with the stochastic gradient-type optimization algorithms used for optimizing the parameters of the DNN in training, for sufficiently small learning rate it is only necessary that the gradient computed per sample be correct in expected value. Although the straight-through estimator is biased, it has the right sign, which is argued to suffice under some conditions [62]. In our application the binarization layer is the first and only such layer, and this approach turns out empirically to be effective.

4.5 NN for the Optimum Decision Rule

A neural network should be able to implement, with appropriate learned parameters, the optimum decision rule. When the data distribution is unknown a neural network is able to learn the parameters in the optimal decision rule from limited samples.

Consider a special case of the compressed segmentation setting described in Section 4.1.3, with $C = 1, A = I$, and $i \in 1, ..., (N - M + 1)^2$. We can use the optimal decision rule (4.28) for compressed classification and the optimal decision rule reduces to

$$i^* = \arg\min_i \|y - x_i\|_F^2 - 2\sigma^2 \log(\pi_i) \tag{4.37}$$

where y and x_i are $N \times N$ matrices. The above decision rule is equivalent to

$$i^* = \arg\max_i \langle y, x_i \rangle - \|x_i\|_F^2/2 + \sigma^2 \log(\pi_i) \tag{4.38}$$

In this section, we describe three neural network architectures that are able to implement this optimal decision rule in the data domain.

4.5.1 Fully connected neural network

The optimal decision rule (4.38) can be implemented as a one-layer dense neural network. Let the weight matrix W of this fully connected (FC) layer have size $N \times N \times (N - M + 1)^2$, set each column of W to exactly the vectorized version of x_i, and set the ith element of bias b of this FC layer to $-\|x_i\|^2/2 + \sigma^2 \log(\pi_i)$. Then the ith element of the FC layer output is

$$\langle W_i, y \rangle - b_i = \langle y, x_i \rangle - \|x_i\|_F^2/2 + \sigma^2 \log(\pi_i) \tag{4.39}$$

The optimal decision rule hence becomes

$$i^* = \arg\max_i \langle W_i, y \rangle - b_i \tag{4.40}$$

The parameter W and b need to be learned from given data samples. Given the data sample pairs $(y^{(i)}, z^{(i)})_{i=1}^D$, $z^{(i)}$ denoting the class label of $y^{(i)}$, the loss function is defined as follows:

$$\begin{aligned} f(y)_i &\triangleq \langle W_i, y \rangle - b_i, i \in [(N - M + 1)^2] \\ L(W, b) &= \frac{1}{D} \sum_{i=1}^{D} \text{CELoss}(\text{Softmax}(f(y^{(i)})), z^{(i)}) \end{aligned} \tag{4.41}$$

The softmax function takes an input of a vector s of dimension R and normalizes it into a probability distribution consisting of R probabilities proportional to the exponentials of the input.

$$\text{Softmax}(s)_i \triangleq \frac{\exp(s_i)}{\sum_{j=1}^R \exp(s_j)}, \forall i = 1, \ldots, R \tag{4.42}$$

This network is a direct implementation of the decision rule in (4.38). However, it has a drawback: the size of the weight W has order $O(N^4)$.

Hence the sample complexity, the number of samples required to successfully train the network, is at least $O(N^4)$ [64]. It follows that this NN architecture is only suitable for small N.

4.5.2 Convolutional neural network (CNN)

In the image segmentation example mentioned in Section 4.1.3, the object is shift-invariant and the rest of the background has zero pixel values in x_i. We can therefore simplify the computation of the term $\langle y, x_i \rangle$ in the optimal decision rule (4.38) using the shift-invariance property:

$$\langle y, x_i \rangle = \langle P_i y, P_i x_i \rangle, P_i : \mathbb{R}^{N \times N} \to \mathbb{R}^{M \times M} \tag{4.43}$$

The operator P_i is a patch extractor for ith object location, $i = 1, \ldots, (N - M + 1)^2$, which takes in the image and extracts an $M \times M$ patch. For every i, $P_i x_i$ is essentially the pixel values of the shift-invariant object and we denote $\kappa \triangleq P_i x_i \in \mathbb{R}^{M \times M}$. The computation of $\langle P_i y, P_i x_i \rangle$ is equivalent to the following convolution operation performed on a 2D image.

$$\langle P_i y, P_i x_i \rangle = \langle P_i y, \kappa \rangle = vec(y \star \kappa)_i \tag{4.44}$$

Now, if the exact size of this shift-invariant object M is known, or an upper bound on the size of a bounding box in which the object can be inscribed is known, we only need one convolution layer with 1 output channel in order to compute $\langle y, x_i \rangle$ for all i. By setting the 2D convolution kernel to κ, this CNN convolves the input image y with κ, producing the feature $vec(y \star \kappa)_i, i = 1, \ldots, (N - M + 1)^2$. The next and last linear layer has an identity weight matrix $W = I \in \mathbb{R}^{(N-M+1)^2 \times (N-M+1)^2}$ and bias $b_i = -\|x_i\|_F^2/2 + \sigma^2 \log(\pi_i)$. The final step is to find the position of the maximum in the output features $Wvec(y \star \kappa) + b$ and this position is our final prediction. By setting these weights κ, W, b, this CNN network performs equivalently to the optimal decision rule:

$$\begin{aligned} i^* &= \arg\max_i \langle y, x_i \rangle - \|x_i\|_F^2/2 + \sigma^2 \log(\pi_i) \\ &= \arg\max_i vec(y \star \kappa)_i - b_i \end{aligned} \tag{4.45}$$

The convolution kernel κ and the bias b need to be learned from the data samples. Similar to (4.41), the loss function is defined as follows:

$$L(\kappa) = \frac{1}{D}\sum_{i=1}^{D} \text{CELoss}(\text{Softmax}(vec(y^{(i)} \star \kappa) - b), z^{(i)}) \tag{4.46}$$

When $M \ll N$, the number of parameters in this CNN is of order $O(M^2) + O(N^2)$, which is far less than for the fully connected network, and thus the CNN requires far fewer data samples to train.

If the object is chosen from one of C classes, then we can increase the number of channels in the convolution layer from 1 to C and implement the optimal decision similarly.

4.5.3 CNN with fixed-size kernels

From the analysis in Section 4.5.2, the kernel size of the CNN needs to be at least $\max_{c=1}^{C} M_c \times \max_{c=1}^{C} M_c$ for multiple objects. When the required kernel side length $\max_{c=1}^{C} M_c$ is close to N, the number of parameters in this CNN is then similar to that of the fully connected network.

In machine learning, it is a common practice to replace a single large filter with multiple layers of small filters that have the same receptive field. Because expressivity of a network increases exponentially with network depth [65], a deeper network has greater expressivity for a given number of free parameters. The receptive field of a layer in CNN refers to the part of the input image that contributes to the filter output at this specific layer. This receptive field increases linearly as we stack more convolutional layers. It is equivalent to the side length of the convolution of multiple layers of kernels. The receptive field of L layers of $k \times k$ filters is $((k-1)L+1) \times ((k-1)L+1)$.

In this section, we discuss a linear CNN with multiple convolution layers of a fixed convolution kernel size (e.g., 3) combined with a final linear layer. Our analysis does not include any nonlinearities in the network. Instead of assigning every parameter in this network, we prove the existence of optimal fixed kernel-size channels, with which the CNN is equivalent to the optimal decision rule in (4.38). The structure of the CNN is shown in Figure 4.2. This network has L convolution layers, each with J output

channels. The last linear layer sums up all the convolution results from the Lth layer and vectorizes the 2D matrix as a vector output.

The following theorem gives a sufficient condition on the existence of the fixed-size optimal channels, for a given number of the layers L and the number of channels J in CNN architecture.

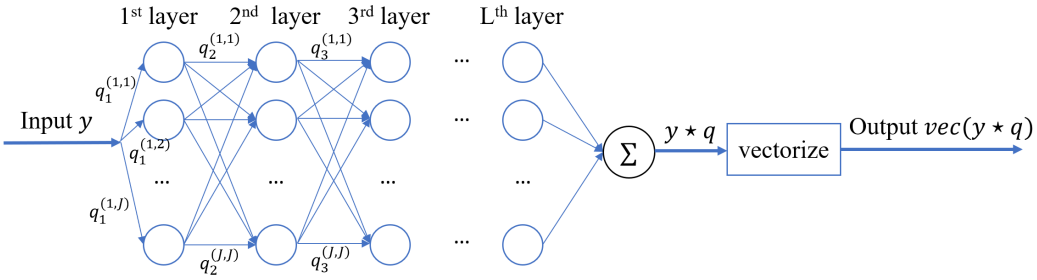


Figure 4.2: Architecture of a CNN with fixed kernel size. It has L convolution layers, each containing J convolution channels.

Theorem 2. *Consider a CNN with filter size* 3×3*, and* L *convolution layers, each layer containing* 4 *convolution channels, with an output linear layer to sum up all the channels at the end. Then this network can produce an output equal to the convolution of the input with an arbitrary* $M \times M$ *kernel if and only if* $L \geq M/2 - 1$*. Furthermore, if* $L < M/2 - 1$*, then regardless of the number of channels in each layer of the CNN, there exist* $M \times M$ *convolution kernels that the CNN cannot realize.*

Proof. Let L denote the finite number of convolution layers, and let J denote the number of convolution channels in each layer. These consecutive L convolution layers combined should act like one $M \times M$ convolution layer for an arbitrary $M \times M$ kernel. The receptive field of the Lth layer of 3×3 kernel is $(2L+1) \times (2L+1)$, which should be larger than $M \times M$, so $L \geq M/2 - 1$. This proves the necessity of the condition on L for any J. What remains to be proved is the sufficiency of the condition on L, for $J = 4$.

Let $q_\ell^{(i,j)} \in \mathbb{R}^{3\times 3}$ denote the 3×3 convolution kernel in the ℓth layer, which takes in the ith channel in the $(\ell - 1)$th layer as input and outputs j-th channel in the ℓth layer. WLOG, assume the input image y has 1

channel. Then the output of the CNN with one convolution layer is

$$\sum_{j_1=1}^{J} y_1 \star q_1^{(1,j_1)} = y_1 \star \sum_{j_1=1}^{J} q_1^{(1,j_1)} \tag{4.47}$$

Similarly, the output of the CNN with two convolution layers is:

$$\sum_{j_2=1}^{J}\sum_{j_1=1}^{J} y_1 \star q_1^{(1,j_1)} \star q_2^{(j_1,j_2)} = y_1 \star \left(\sum_{j_1=1}^{J} q_1^{(1,j_1)} \star \left(\sum_{j_2=1}^{J} q_2^{(j_1,j_2)}\right)\right) \tag{4.48}$$

and the output of the CNN with L convolution layers is

$$y_1 \star \left(\sum_{j_L=1}^{J}\sum_{j_2=1}^{J}\cdots\sum_{j_L=1}^{J} q_1^{(1,j_1)} \star q_2^{(j_1,j_2)} \star \cdots \star q_L^{(j_{L-1},j_L)}\right) \tag{4.49}$$

Let q denote the overall convolution filter

$$\begin{aligned} q &\triangleq \sum_{j_L=1}^{J}\sum_{j_2=1}^{J}\cdots\sum_{j_L=1}^{J} q_1^{(1,j_1)} \star q_2^{(j_1,j_2)} \star \cdots \star q_L^{(j_{L-1},j_L)} \\ &= \sum_{j_L=1}^{J} q_1^{(1,j_1)} \star \left(\sum_{j_2=1}^{J} q_2^{(j_1,j_2)} \star \cdots \star \left(\sum_{j_L=1}^{J} q_L^{(j_{L-1},j_L)}\right)\right) \end{aligned} \tag{4.50}$$

We would like to know the minimum J for a given kernel size M and a given number of convolution layers $L = \left\lfloor \frac{M-1}{2} \right\rfloor$. We use mathematical induction to find the minimum number J.

Consider the case of $M = 5, L = 2$ first. It is desired that for any 5×5 filter kernel κ_5, there exist $q_1^{(1,j_1)}$s and $q_1^{(j_1,j_2)}$s such that the following holds:

$$\sum_{j_1=1}^{J} q_1^{(1,j_1)} \star \left(\sum_{j_2=1}^{J} q_2^{(j_1,j_2)}\right) = \kappa_5 \in \mathbb{R}^{5\times 5} \tag{4.51}$$

Letting $q_2^{j_1} = \sum_{j_2=1}^{J} q_2^{(j_1,j_2)}$, this equation is equivalent to

$$\sum_{j_1=1}^{J} Q_1^{(1,j_1)} vec\left(q_2^{j_1}\right) = vec(\kappa_5) \in \mathbb{R}^{25} \tag{4.52}$$

where $Q_1^{(1,j_1)} \in \mathbb{R}^{25\times 9}$ is the doubly block matrix (a special case of Toeplitz matrix) corresponding to the 2D convolution of $q_1^{(1,j_1)}$. We can rewrite the

above equation as

$$\left[Q_1^{(1,1)}|Q_1^{(1,2)}|\dots|Q_1^{(1,J)}\right]\begin{bmatrix} vec(q_2^1) \\ vec(q_2^2) \\ \dots \\ vec(q_2^J) \end{bmatrix} = vec(\kappa_5) \tag{4.53}$$

For an arbitrary 5×5 filter kernel κ_5, as long as the block matrix $Q_1 \triangleq \left[Q_1^{(1,1)}|Q_1^{(1,2)}|\dots|Q_1^{(1,J)}\right]$ has full row rank, then there exists a solution of $\begin{bmatrix} vec(q_2^1) \\ vec(q_2^2) \\ \dots \\ vec(q_2^J) \end{bmatrix}$. Experimentally we found that for random $q_1^{(1,j)}$s , $J = 4$ almost surely guarantees the full row rank of Q_1.

Consider a larger number of $M > 5$, and assume we can generate any arbitrary kernel of size $(M-2) \times (M-2)$. From (4.50), it is desired that for any filter kernel κ_M of size $M \times M$

$$\sum_{j_L=1}^{J} q_1^{(1,j_1)} \star \left(\sum_{j_2=1}^{J} q_2^{(j_1,j_2)} \star \dots \star \left(\sum_{j_L=1}^{J} q_L^{(j_{L-1},j_L)}\right)\right) = \kappa_M \in \mathbb{R}^{M\times M} \tag{4.54}$$

By assumption, the above equation reduces to

$$\sum_{j_1=1}^{J} q_1^{(1,j_1)} \star \kappa_{M-2}^{(j_1)} = \kappa_M \in \mathbb{R}^{M\times M} \tag{4.55}$$

Following the analysis of 2-convolution layer CNN, we can rewrite the above equation as

$$\left[Q_1^{(1,1)}|Q_1^{(1,2)}|\dots|Q_1^{(1,J)}\right]\begin{bmatrix} vec(\kappa_{M-2}^1) \\ vec(\kappa_{M-2}^{(2)}) \\ \dots \\ vec(\kappa_{M-2}^{(J)}) \end{bmatrix} = vec(\kappa_M) \tag{4.56}$$

where $Q_1^{(1,j)} \in \mathbb{R}^{M^2\times(M-2)^2}$ doubly block matrix that is equivalent to the 2D convolution with $q_1^{(1,j)}$. As M increases, the number of convolution channels J that guarantees the full row rankness of the block matrix Q_1 approaches 2 and is lower bounded by 2.

Combining with the case of $M = 5, L = 2$, the minimum number of channels J to implement the optimal decision rule for an arbitrary M is 4. □

Remark (1)**.** *We have shown that $J = 4$ channels in each layer are sufficient for the result to hold. However, this was obtained by considering the conditions for the matrix Q_1 in* (4.52) *to have generically full row rank, which is not necessary for* (4.52) *to have a solution for every right-hand-side, because Q_1 also has learnable (tunable) parameters. Hence, we have not shown that $J = 4$ is a necessary condition. However, empirical results of training a linear CNN suggest that $J = 4$ is indeed a necessary condition. Hence we propose the following:*

Conjecture (1)**.** *For $J < 4$, the sufficient condition of Theorem 2 does not hold.*

Remark (2)**.** *The proof of Theorem 2 and our empirical results summarized in Conjecture 1 reveal the role of multiple channels in CNNs for image segmentation tasks.*

Remark (3)**.** *For the image segmentation problem with C shift-invariant objects, with each object bounded in a $M_c \times M_c$ box, we can use a similar architecture of a linear CNN to implement the optimal decision rule. We simply use $M = \arg\max_{c\in[C]} M_c$ for the theorem. The convolution layers are replicated C times, and each set of convolution layers can generate one $M_c \times M_c$ filter kernel. The final linear layer is modified to take in C channels instead of 1.*

Remark (4)**.** *If we adopt the popular CNN architecture, using a ReLU layer after each convolution layer, then the analysis of the number of channels J becomes complex. Because the ReLU only transmits one polarity, and two ReLUs would be required to implement a linear operation, more channels will be required in general to implement an arbitrary $M \times M$ convolution by a ReLU CNN than by a linear CNN. We do not discuss this scenario further theoretically and instead present the results of numerical experiments in the next chapter.*

Similar to section 4.5.2, the last linear layer has an identity weight matrix $W = I$ and the bias $b_i = -\|x_i\|_F^2/2 + \sigma^2 \log(\pi_i)$. The optimal

decision rule of this L-layer CNN is

$$\begin{aligned} q &\triangleq \sum_{j_L=1}^{J} \sum_{j_2=1}^{J} \cdots \sum_{j_L=1}^{J} q_1^{(1,j_1)} \star q_2^{(j_1,j_2)} \star \cdots \star q_L^{(j_{L-1},j_L)} \\ i^* &= \arg\max_i vec\,(y \star q)_i - b_i \end{aligned} \tag{4.57}$$

and the loss function is defined as:

$$L(q,b) = \frac{1}{D} \sum_{i=1}^{D} \text{CELoss}\left(\text{Softmax}\left(vec(y^{(i)} \star q) - b\right), z^{(i)}\right) \tag{4.58}$$

The number of parameters in this CNN network has order of $O(M) + O(N^2)$, because the number of 3×3 convolution kernels is $J^2 \lfloor \frac{M-1}{2} \rfloor = 16 \lfloor \frac{M-1}{2} \rfloor$.

4.5.4 U-Net: Empirically successful segmentation network

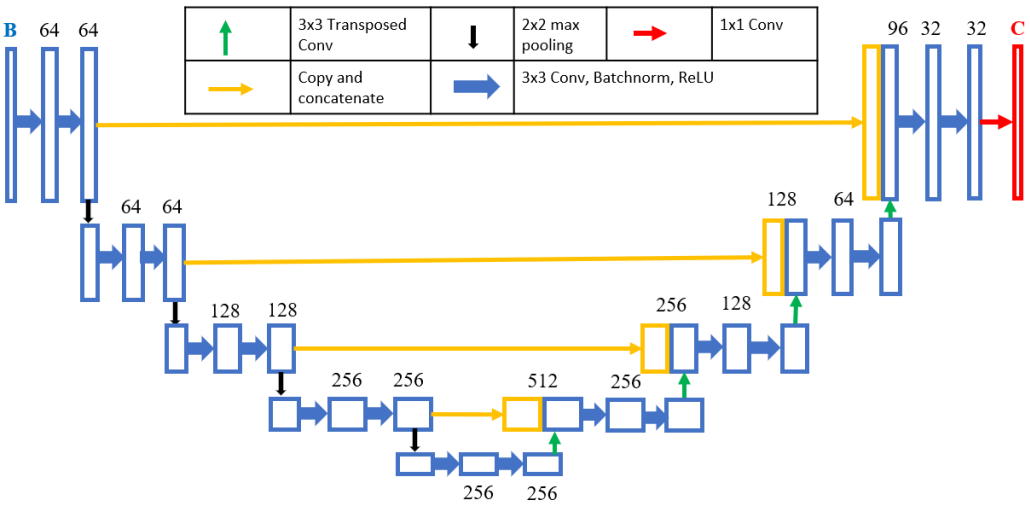


Figure 4.3: U-Net deep neural network structure and notations.

A U-Net [59] structure can be used as the segmentation network f_θ. Since the U-Net contains not only convolution layers but also BatchNorm layers and pooling layers, as well as skip connections, the analysis of implementing the optimal decision rule in a U-Net is complex and is not attempted here. The U-Net is a successful model for various medical imaging segmentation applications [66] thanks to the following architectural

aspects: (i) its encoder-decoder structure provides it with a large receptive field, enabling it to understand the full context of the image; and (ii) its skip-connections that propagate the low-level features to the final semantics enable it to be responsive to local features.

Figure 4.3 shows the U-Net architecture adapted from one used for monochrome image segmentation [59] by increasing the number of input channels from one to B. The input to this U-Net is a multispectral image with B spectral bands. This input could be a multispectral image x, in the case of segmentation in the data domain, or an image $A^T y$, derived from the CS measurement y, in the case of segmentation in the compressed domain. Assuming that A has full row rank (no redundant measurements), it follows that A^T has a left inverse, and therefore the information contents of $A^T y$ and y are the same, and $A^T y$ may be used as the input to the DNN rather than y, without loss. The multispectral feature extraction is all done in the first layer of the network.

In detail, this U-Net has 4 basic encoder blocks, 4 decoder blocks in the left and right halves of the diagram, respectively, one center block and direct copy connections between them. These 4 encoder blocks reduce the convolution feature size by half in each block and constitute the contraction branch. In turn, the 4 decoder blocks double the feature size in each block and constitute the expansion branch.

Each encoder block has 2 composite layers and a max pooling layer to reduce the feature size. Each composite layer consists of a 3×3 convolution layer, a batch normalization (BN) layer, and a Rectified linear unit (ReLU) layer. The number on top of each rectangle is the number of channels of the features. Each decoder block has two composite layers and one transposed convolution layer to increase the feature size.

The decoder block also incorporates features from previous encoder layers through direct concatenation. The center block is similar to the decoder block except it does not include the concatenation at its input. In the last layer, a 1×1 convolution layer (red arrow) produces the final C-channel tensor of size $C \times H \times W$ and applying the softmax function and the argmax function to this tensor yields $z \in [C]^{H \times W}$.

4.6 Other Inference Paradigms

In this section, we describe other network architectures to perform inference using compressed measurements and original data. We limit the discussion to the line and band selection scheme and use the U-Net as the segmentation network.

4.6.1 Segmentation on reconstructed data

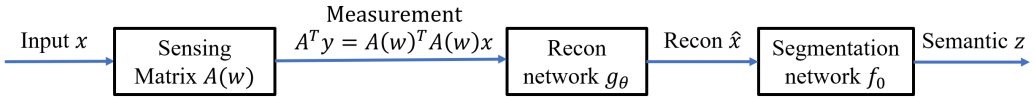


Figure 4.4: Composite segmentation network block diagram.

To compare the direct inference in compressed domain with the traditional reconstruction + inference pipeline, we use the network architecture shown in Figure 4.4, called composite segmentation network. Here, the segmentation U-Net f_0 is a pre-trained network trained on fully scanned data and is fixed in this pipeline. The reconstruction U-Net g_θ takes in $A^T y$ as input and outputs a reconstruction $\widehat{x}$. The reconstruction U-Net g_θ has the same network architecture as segmentation U-Net except that the final layer will produce a B-band image instead of C-class predictions. The reconstruction network g_θ is trained to minimize the mean squared error between the input x and the reconstructed results $\widehat{x}$ for a tunable sensing matrix $A(w)$.

$$\min_{\theta,w} \frac{1}{N} \sum_{n=1}^{N} \|g_\theta(y^{(n)}) - x^{(n)}\|_2^2 + \lambda_1 \|A(w)\|_0 \tag{4.59}$$

The final prediction of this composite network is then given by $f_0(g_\theta(y))$.

4.6.2 Multitask learning: Reconstruction and segmentation

Figure 4.5 shows the diagram of multitask learning (MTL) to reconstruct 3D data and produce semantics at the same time. The idea of multitask learning [67] is not new: by sharing representations between related tasks, a

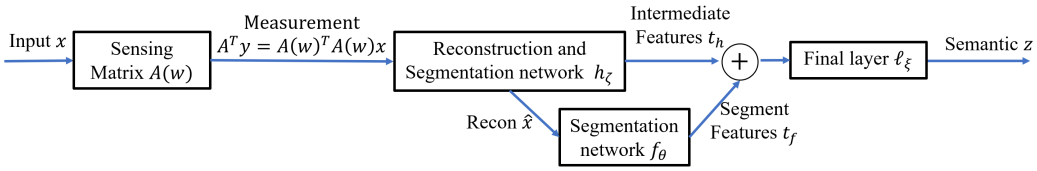


Figure 4.5: Multitask learning network block diagram.

joint model that is trained to perform multiple tasks generalizes better on a single original task. In this MTL model, we utilize eavesdropping to improve the segmentation accuracy.

The reconstruction and segmentation U-Net h_ζ takes in the $A^T y$ as input and has two outputs: the reconstruction $\widehat{x}$ and the intermediate features t_h. The U-Net h_ζ has one contraction branch composed of 4 encoder blocks, which are the same as in Section 4.5.4. The U-Net h_ζ has two expansion branches, each consisting of 4 decoder blocks. One expansion branch produces the reconstruction $\widehat{x}$ while the other branch produces intermediate features t_h. Another segmentation U-Net f_θ takes in the reconstruction $\widehat{x}$ and generates the segmentation features t_f. The final convolution layer ℓ_ξ of this joint network combines the intermediate features t_h and segmentation features t_f by concatenating them together and producing the final semantics z.

The final decision of the segmentation is based on the reconstructed $\widehat{x}$ and the intermediate feature t_h. The eavesdropping of the multi-task learning happens in the contraction branch of h_ζ.

We set the loss function of this MTL network to:

$$\begin{aligned} L(\theta, \zeta, \xi, w) = & \frac{1}{N} \sum_{n=1}^{N} \text{CELoss}(\ell_\xi(t_h, t_f), z^{(n)}) \\ & + \lambda_{MSE} \frac{1}{N} \sum_{n=1}^{N} \|x^{(n)} - \widehat{x}\|_2^2 \ + \lambda_1 \|A(w)\|_0 \end{aligned} \tag{4.60}$$

where t_h can be written as a function $t_h(h_\zeta, A(w)x^{(n)})$, t_f as a function $t_f(f_\theta, h_\zeta, A(w)x^{(n)}))$ and $\widehat{x}$ as $\widehat{x}(h_\zeta, A(w)x^{(n)})$. The final prediction is given by $\ell_\xi(t_h, t_f)$. This network is trained to minimize the loss function defined in (4.60)

$$\min_{\theta, \zeta, \xi, w} L(\theta, \zeta, \xi, w) \tag{4.61}$$

CHAPTER 5

SYNTHETIC DATA NUMERICAL EXPERIMENTS: NEURAL NETWORKS VS OPTIMUM CLASSIFICATION

In this chapter, we compare the performance of a neural network of a simple architecture versus that of the optimal decision rule in the data domain. These experiments show the neural network's capability of choosing optimal parameters to implement the optimal decision rule for a given number of training samples and training epochs, which verifies the analysis in Section 4.5.

5.1 Synthetic Dataset for Classification Task

In Section 4.5, we have listed three neural network architectures, namely, fully connected neural network (section 4.5.1), CNN (section 4.5.2), and CNN with fixed size filters (section 4.5.3). Given the correct parameters, these neural networks can implement the optimal decision rule in the data domain. We also briefly mentioned their sample complexity. In this section, we generate a synthetic monochromatic dataset to experimentally verify the number of samples these networks need to learn the optimal parameters.

We simulate a shift-invariant object of size $M \times M$ randomly appearing in a $N \times N$ background. Figure 5.1 shows two sample images with negligible noise, where the object of fixed pixel mean appears in the different locations. The goal of this classification task is to determine the location of the object using the input image.

The hypothesis of this classification task is formulated in (2.9), with $(N - M + 1)^2$ hypotheses, and in the data domain, the hypothesis reduces to:

$$y = x_i + v, v \sim \mathcal{N}(0, \sigma^2 I) \tag{5.1}$$

where x_i represents the mean of the image. The mean of the image is a random sample from Uniform$[0, 1]^{M \times M}$, and we fix the x_i in these

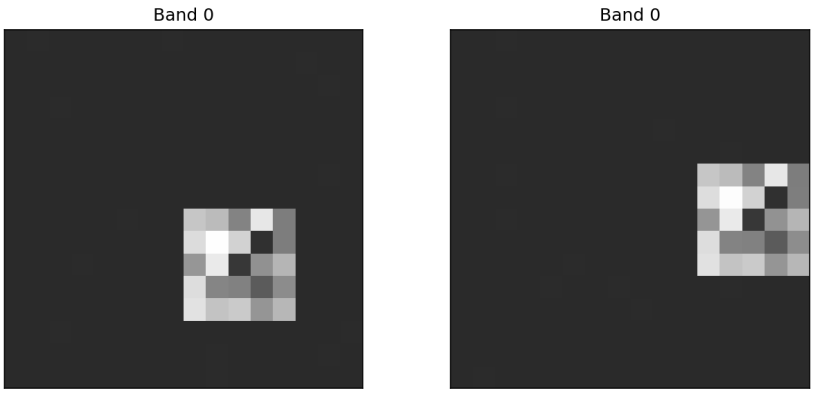


Figure 5.1: Sample image from synthetic dataset for classification.

numerical experiments. The optimal decision rule reduces to

$$i^* = \arg\min_i \|y - x_i\|^2 - 2\sigma^2 \log(\pi_i) \tag{5.2}$$

where π_i represents the prior probability of the ith location. We define two different spatial distributions of the object:

$$\text{Uniform spatial prior: } \pi_i = \frac{1}{(N-M+1)^2}, \forall i \tag{5.3}$$

In the Gaussian spatial prior, letting (s,t) denote the top left coordinate of the object, then

$$\begin{aligned} s' &\sim \mathcal{N}\left(\frac{N-M}{2}, \frac{N-M}{8}\right), s = clip(s', 0, N-M-1) \\ t' &\sim \mathcal{N}\left(\frac{N-M}{2}, \frac{N-M}{8}\right), t = clip(t', 0, N-M-1) \end{aligned} \tag{5.4}$$

where $clip(x,a,b) = \begin{cases} a, & \text{if } x < a, \\ x, & \text{if } a \le x \le b, \\ b, & \text{if } x > b. \end{cases}$

The object appears more often around the center of the image under the Gaussian spatial prior.

In this experiment, we set $N = 16, M = 5, C = 144$. We generate two datasets under Gaussian and uniform spatial distributions. Each dataset has 10,000 training images and 20,000 test images. Using different network

architectures described in Section 4.5.1, 4.5.2 and in 4.5.3, we train the classification networks using the Adam optimizer with learning rate 0.001 for 60 epochs.

5.2 Simple Networks to Simulate Optimal Decision Rule

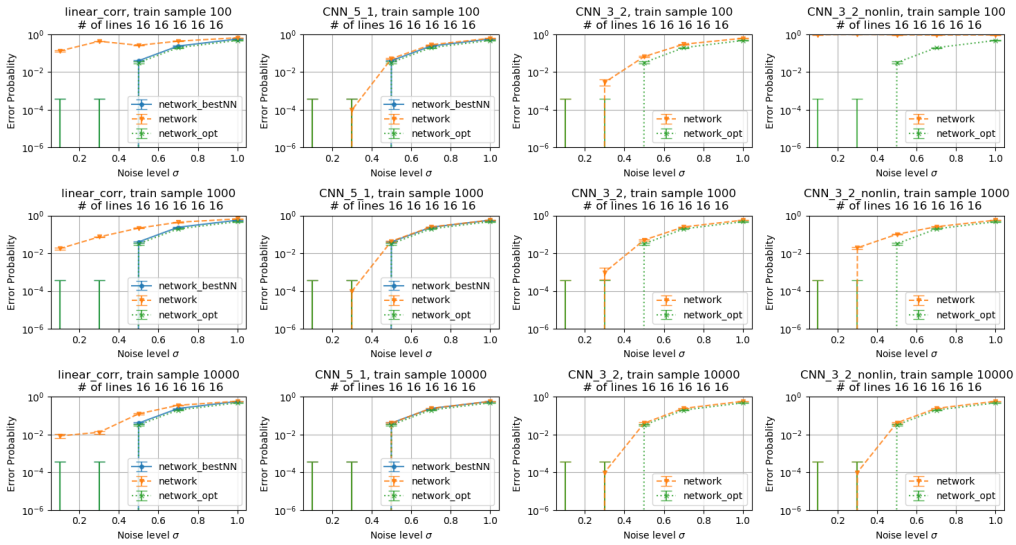
Figure 5.2 shows the error probability versus the noise level σ in the synthetic data setting using different classifier network architectures. The networks is trained using the Adam optimizer with learning rate 0.01 for 60 epochs. The number of training samples varies between 100, 1,000, and 10,000 and the training samples are uniformly randomly selected from the training set. The error probability of the networks, defined in (2.17), is calculated over 20,000 independent test samples. The error bar in each figure indicates 95% confidence level.

In each subfigure, there are two or three legends depending on the network architecture:

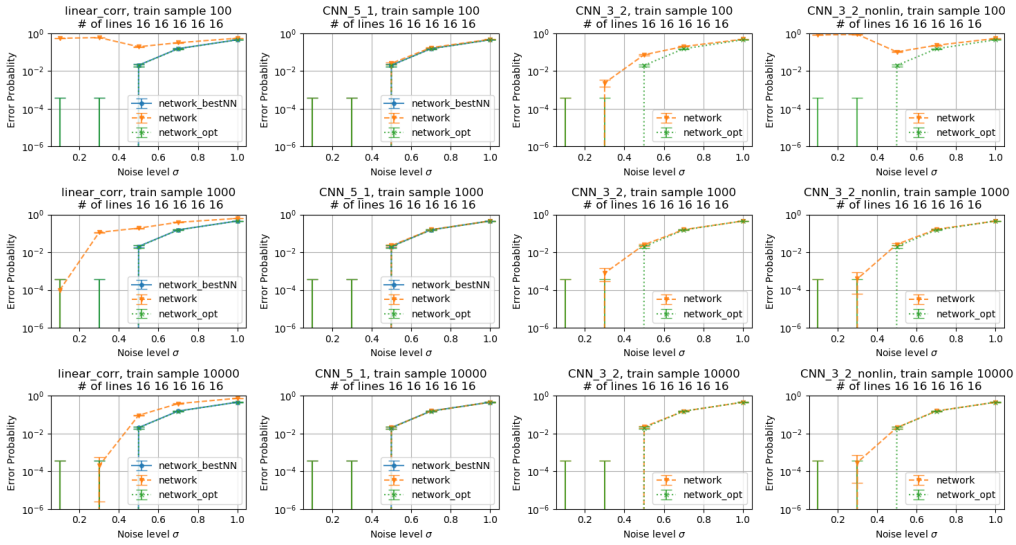
1. `network_opt`: The optimal decision rule.
2. `network`: The trained neural network with given number of training samples.
3. `network_bestNN`: The neural network with preassigned best weights.

The title of each subfigure indicates the name and the configuration of the classifier network.

1. `linear_ corr` represents the fully connected neural network.
2. `CNN_5_1` represents a 5×5-kernel CNN. It consists of a single channel convolution layer and a linear layer.
3. `CNN_3_2` represents a 3×3-kernel CNN. It consist of 2 convolution layers with 2 channels and a linear layer.
4. `CNN_3_2_nonlin` represents a 3×3-kernel CNN. It consist of 2 convolution layers with 2 channels, 2 ReLU layers and a linear layer.



Gaussian spatial prior



Uniform spatial prior

Figure 5.2: Error probability versus noise level using different network architectures. Top: the random object is generated under Gaussian spatial prior. Bottom: the random object is generated under uniform spatial prior.

The title also contains the number of training samples used to train the network, selecting from 100, 1000, and 10000.

With both Gaussian and uniform spatial prior, the networks with preassigned weights have the same performance as the optimal decision rule, which verifies the analysis in Section 4.5. For the trained networks, they all perform the same or worse than their optimal decision rule counterpart,

which is expected. As the number of training samples increases, an improvement of the network performance is observed. Empirically, with 10,000 training samples, the CNN achieves performance similar to that of the optimal decision rule evaluated by 10,000 test samples, but the fully connected network requires more training samples. This result corroborates the aforementioned sample complexities of each network architecture.

5.3 Summary

In this chapter, we compare the performance of the neural networks with the optimal decision rule in the data domain. We experimentally show the neural networks' capability of selecting optimal parameters when given enough training samples.

CHAPTER 6

SYNTHETIC DATA NUMERICAL EXPERIMENTS: OPTIMIZING THE ACQUISITION

In this chapter, we compare the neural network's performance for inference tasks in the compressed domain, using different sensing schemes. We show that the sensing matrix could be optimized as a parameter in the neural network.

6.1 Compressed Classification Dataset

We build a synthetic multispectral dataset similar to the monochromatic dataset. We again simulate a shift-invariant object of size $M \times M$ randomly appearing in an $N \times N$ background, which is imaged for B spectral bands. Let $p \in \mathbb{R}^{M \times M \times B}$ denote the pixel values of the object in all spectral bands. Each element $p_{i,j,b}$ can be set arbitrarily to imitate the correlation between spectral bands, where i, j are spatial indices and b is the indices of spectral bands. We set different pixel values of the object in Section 6.1.3 and Section 6.1.4. The SNR for each pixel inside the object can then be calculated, $\mathrm{SNR}_{i,j,b} := 10 \log(p_{i,j,b}^2/\sigma^2)$. We use $\mathrm{SNR}_b := 10 \log\left(\sum_{i,j} p_{i,j,b}^2/M^2\sigma^2\right)$ to denote the SNR of b-th spectral band.

In this experiment, we set $N = 16, M = 5, B = 8$. We generate two datasets under Gaussian or uniform distribution for the object location, each having 10,000 training images and 5,000 test images. These two datasets only differ in the distribution of the object position.

6.1.1 Compressed classification network training

We apply the line selection and band selection compressed sensing schemes (illustrated in Section 3.2) and the DD-CASSI (illustrated in Section 3.3) to this synthetic data and obtain the measurements $y = Ax$. We use the

compressed classification network architecture shown in Figure 4.1, and use the U-Net (Section 4.5.4) as the classifier network f_θ. We use (4.31) and (4.34) as the loss function for the line/band selection and DD-CASSI schemes, respectively. We empirically choose λ_1, the hyper-parameter in (4.31) that controls the sparsity of the final scanning mask $G(w)$, to range from 0.001 to 0.1, so that the sparsity of $G(w)$ ranges from 0.35 to 1.

We train the compressed classification network using the Adam optimizer with learning rate 0.001. For a network with a tunable CS operator A, we adopt a 2-stage training strategy: initialize the binary parameter w^b randomly such that its sparsity (number of 1s / total elements) is 0.5, train the segmentation network f_θ jointly with the parameter w for 60 epochs, then fix w and train the U-Net f_θ for another 60 epochs.

Note that in the line selection or band selection scheme, we do not have a closed-form relationship between the sparsity of $G(w)$ and the ℓ_0 penalty coefficient λ_1. Instead, to enable a fair comparison between the learned and a random mask we generate for each λ_1 a random mask that has the same number of non-zeros as the learned mask.

6.1.2 Optimum decision rule benchmark

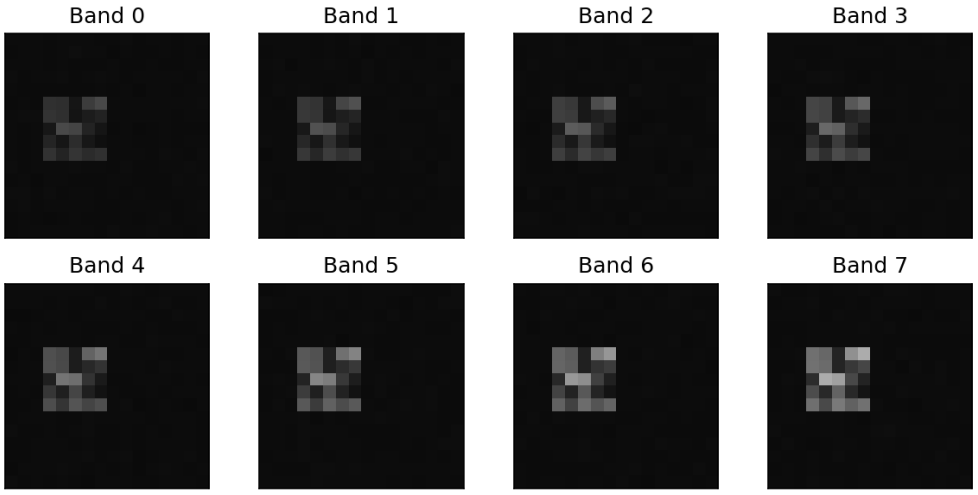
We use the optimal decision rule mentioned in Section 4.2 as a benchmark, and compare the performance of the DNN to that of the optimal decision rule. In order to examine the acquisition scheme learned in the DNN, we use the learned mask $A(\widehat{w})$ obtained from the trained neural network to compress the data and apply the optimal decision rule to the measurements, and compare the results against a random mask $A(w)$.

6.1.3 Homogeneous dataset

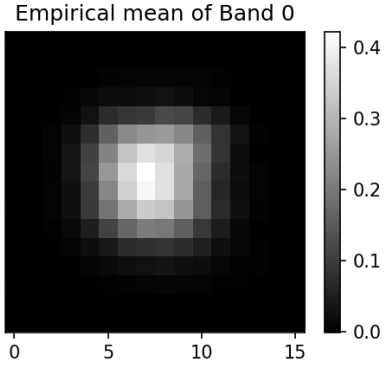
Experimental Setup

In multispectral imaging, different spectral bands often present a similar silhouette but differ in details. In the first experiment, we imitate this setting and vary the intensity of each spectral band. This is implemented by choosing one random sample $p_{:,:,0} \in \mathbb{R}^{M\times M}$ from the uniform distribution,

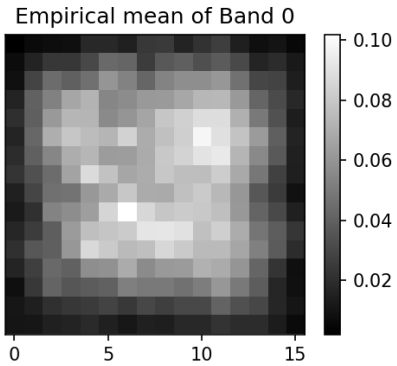
$$p_{i,j,0} \sim \text{Uniform}[0,1], \forall i \in [M], \forall j \in [M] \tag{6.1}$$



(a) Sample image with 8 spectral bands



(b) Empirical mean of 0-th band under Gaussian spatial prior.



(c) Empirical mean of 0-th band under Gaussian spatial prior.

Figure 6.1: (a): One sample image input of the first compressive classification experiment, for $N = 16, B = 8, M = 5, \sigma = 0.01$ and under Gaussian spatial distribution. The object's shape is the same across spectral bands but its intensity varies.

and setting all other spectral bands using

$$p_{i,j,b} = \alpha^b p_{i,j,0}, \alpha > 1, \text{ for } b = 1, \ldots, B. \tag{6.2}$$

The increasing intensity of spectral bands indicates the last spectral band is least noisy. We set the coefficient $\alpha = 1.15$ so that the average SNR of the spectral bands differs by 9 dB between Band 0 and Band 7, with smaller differences between other bands. As a consequence, the band selection scheme under this setting should select the spectral bands with higher intensity.

We generate two synthetic datasets under the Gaussian or uniform spatial distribution. Figure 6.1a shows one sample of a synthetic multispectral image x where spectral bands have varying intensities. Figure 6.1b shows the empirical mean of the 0th spectral band of 100 samples under the Gaussian spatial prior and Figure 6.1b shows the same except under the uniform spatial prior. Under the Gaussian prior, the object tends to appear around the center more than under the uniform prior. As a consequence, under the Gaussian spatial prior, we expect the line selection scheme to select lines around the center.

Results and Discussion

Figure 6.2 and Figure 6.3 show the binary line pattern of the line selection schemes under Gaussian and uniform spatial prior, respectively. Each subfigure has size $B \times N$, and the x-axis represents the horizontal line position while the y-axis represents spectral bands. The red and white elements represent scanned lines and skipped lines, respectively. For example, in the top left subfigure of Figure 6.2, the bottom and top two lines are all skipped. These subfigures are arranged according to the number of scanned lines (the number of lines, noise level σ and the sparsity penalty coefficient λ_1 are labelled in the title), except for the last row. The last row of subfigures shows the heatmaps of the line patterns. In general, under the Gaussian spatial prior, the selected lines appear more frequently in the center. This validates the efficacy of the line selection scheme.

For both spatial priors, the learned line patterns allow line gaps within the same spectral band (indicated by the white horizontal bars), but the gaps rarely exceed the object side length $M = 5$. This ensures that at least one line per band "hits" the object and helps identify its position. The gap

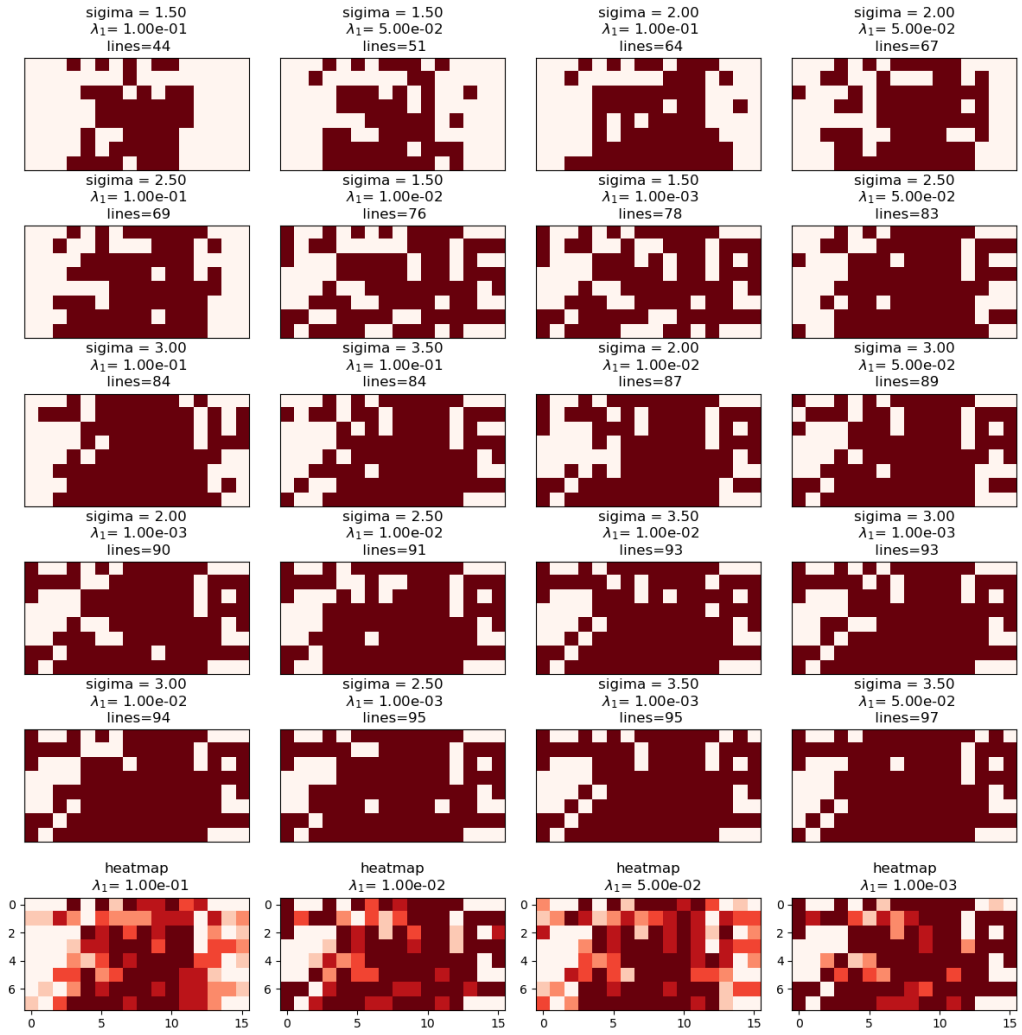


Figure 6.2: The binary masks of the learned line selection scheme under Gaussian distribution with different noise levels and λ_1s. Last row: the heatmap of selected lines.

arrangement also indicates that scanning interlacing lines between the spectral bands is a good choice for the line selection scheme. For example, scan the 1st, 5th, and 9th for the first spectral band, scan the 2nd, 6th, and 10th for the second spectral band, etc. This increases the probability that multiple spectral bands are sensed per object location.

Figure 6.4 shows the error probability versus the noise level σ of the line selection scheme under different spatial priors. There are two pairs of legends in each subfigure.

1. `random_line` or `learned_line`: Prediction of the DNN using a random line selection scheme or the learned line selection scheme.

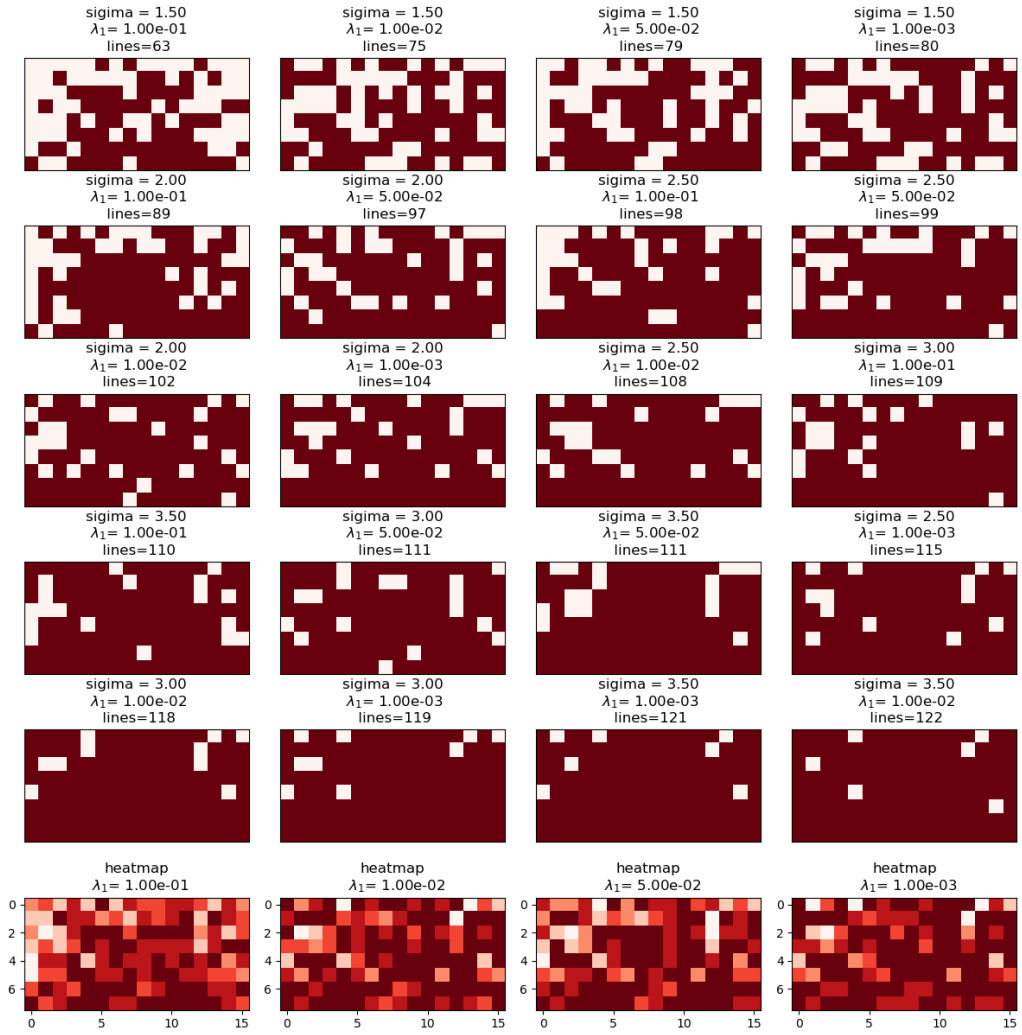
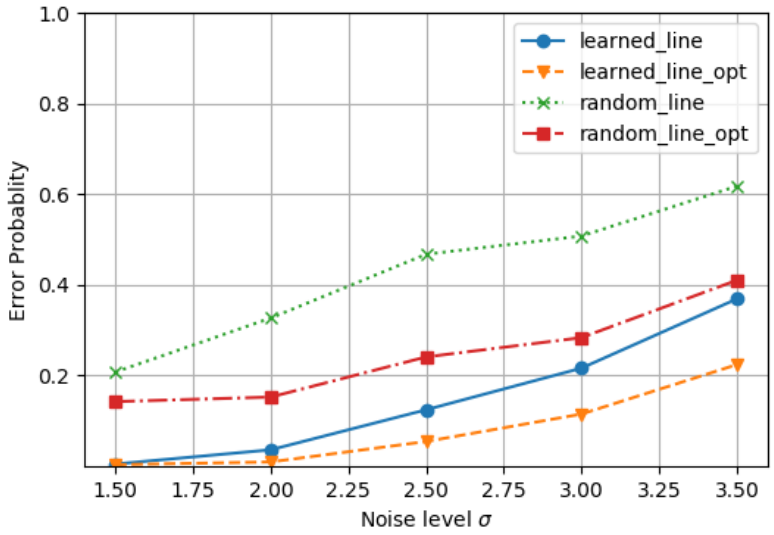


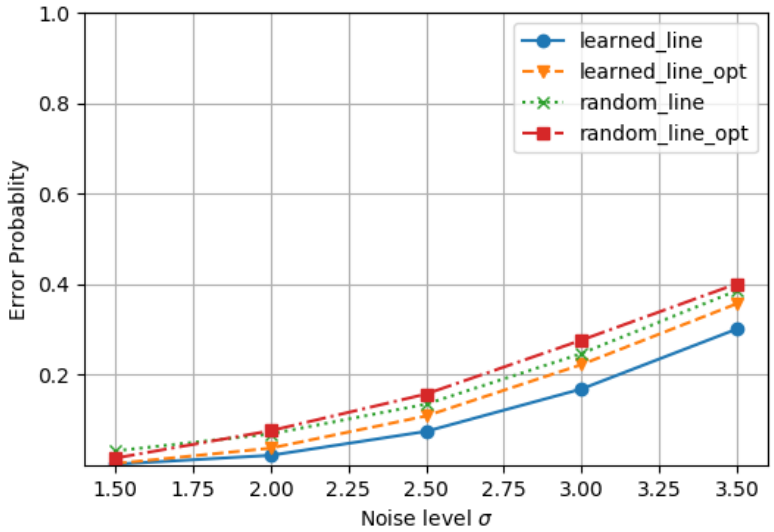
Figure 6.3: The binary masks of the learned line selection scheme under Gaussian distribution with different noise levels and λ_1s. Last row: the heatmap of selected lines.

2. `random_line_opt` or `learned_line_opt`: Prediction of the optimal decision rule, where the compressed data is obtained using a random or the learned line selection's mask.

The top and bottom subfigures in Figure 6.4 show the error probability versus the noise levels under Gaussian spatial prior and uniform spatial prior, respectively. In both Gaussian case and uniform case, the DNN classifies better using the learned line selection mask than a random mask at all noise levels. Also, the optimal decision rule with the same learned lines selection's mask $G(w)$ yields lower error probability compared to a random line mask. This suggests that a learned scheme does extract more



Gaussian spatial prior



Uniform spatial prior

Figure 6.4: Error probability versus noise level using line selection as CS scheme and different inference methods. In general, the number of selected lines increases with the noise levels. Top: the object location follows Gaussian spatial prior. Bottom: the object location follows uniform spatial prior.

useful lines, rather than the effect being only due to a deeper DNN. It is evident that the impact of learning the mask is greater for the Gaussian distribution of the object position - perhaps because there is more to learn

in this instance.

Surprisingly, in the case of a uniform distribution of the object position, the DNN's performance is even better than an optimal decision rule. Because no method can outperform the optimal decision rule in terms the error probability, for which it is optimal, we explored possible reasons for this discrepancy. A computation of the empirical distribution of object location in the numerical experiment revealed that the relative frequency of the object location at various positions deviated by up to $\pm 29\%$ from the uniform probability distribution of $1/(N-M+1)^2$ per location. Because the DNN was trained on the empirical probability, whereas the optimum decision rule assumed a uniform distribution, the DNN gained "an unfair advantage" over the optimum decision rule, thus enabling the DNN to overperform at the higher noise levels, when the prior plays a greater role in the decision.

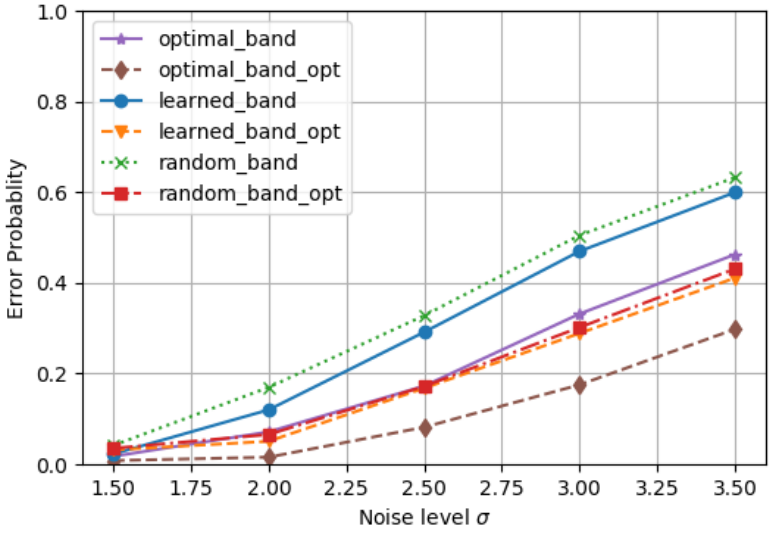
Figure 6.5 shows the error probability of the band selection scheme. There are three pairs of legends in each subfigure.

1. `random_band` or `learned_band`: Prediction of the DNN using a random band selection or the learned band selection.

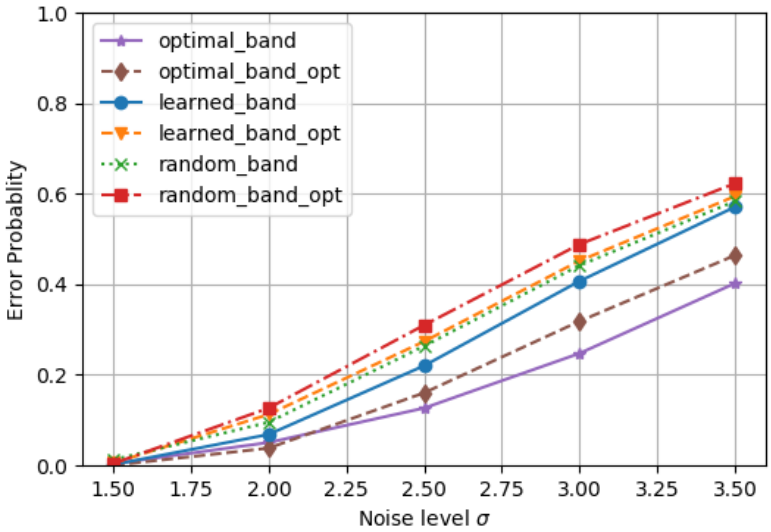
2. `random_band_opt` or `learned_band_opt`: Prediction of the optimal decision rule (choose $m = \arg\max_m p(H_m|x)$), where the compressed data is obtained using a random band selection's mask or the learned band selection's mask.

3. `optimal_band` or `optimal_band_opt`: Prediction of the DNN using the optimal spectral bands or prediction of the optimal decision rule using optimal spectral bands. The 'optimal' bands are defined as the spectral bands that have higher SNRs.

The top and bottom subfigures in Figure 6.5 show the error probability versus different noise levels under Gaussian spatial prior and uniform spatial prior, respectively. Again, for both priors, the learned CS scheme outperforms a random CS scheme. However, error probabilities with the learned band selection mask are higher than with the optimal bands, because the DNN does not always learn the optimal bands. This implies that the binary layer in the DNN is harder to train than the rest of the



Gaussian spatial prior



Uniform spatial prior

Figure 6.5: Error probability versus noise level under different CS schemes and different inference methods. In general, the number of selected bands increases with the noise level. Top: the random object is generated under Gaussian spatial prior. Bottom: the random object is generated under uniform spatial prior.

continuous classifier CNN and we cannot guarantee that the optimal bands get selected every time.

We again observe, under the uniform spatial prior, that the DNN does

even better than the optimal decision rule at the higher noise levels. Our explanation is the same as that given for Figure 6.4.

From Figure 6.5 and Figure 6.4, we can see that the line selection scheme generally performs better than the band selection scheme because the former has more degrees of freedom (DOF).

6.1.4 Inhomogeneous dataset

Experimental Setup

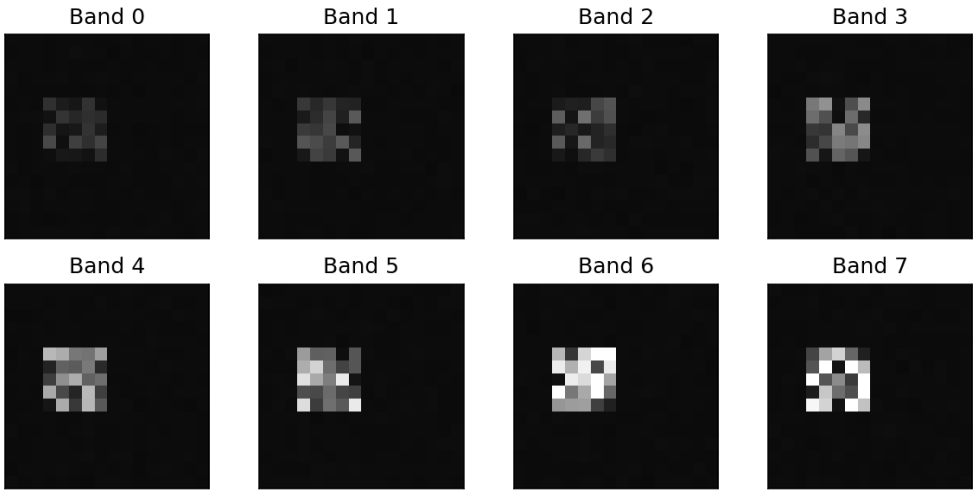
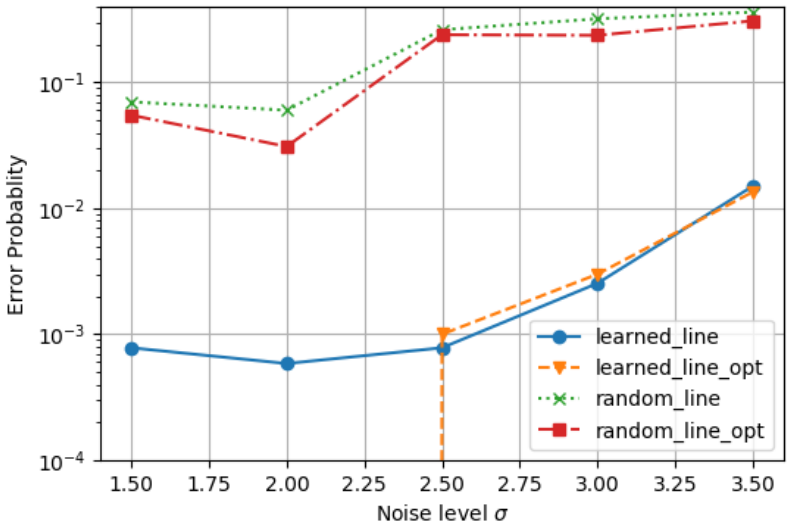


Figure 6.6: One sample image input of the second compressive classification experiment, when $N = 16, B = 8, M = 5, \sigma = 0.01, \alpha = 1.3$.

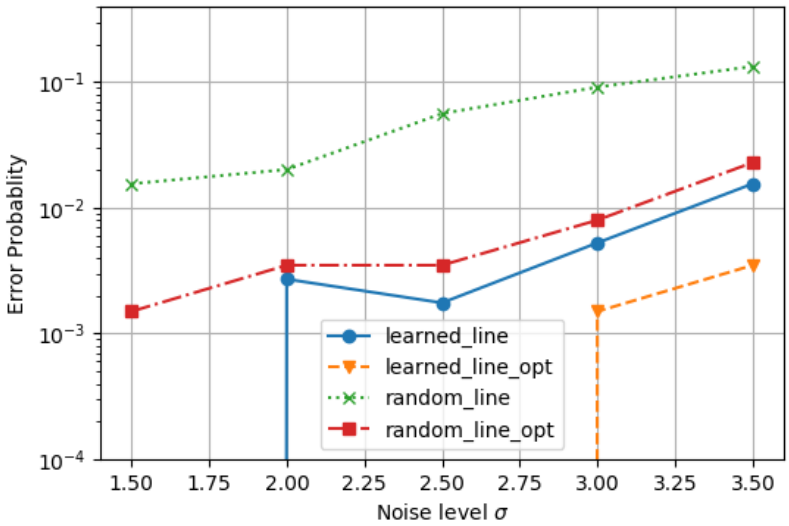
In the second experiment, we keep the dimension $N = 16, M = 5, B = 8$ the same and only change the pixel values of the object $p_{i,j,b}$. In medical imaging applications, one "spectral band" could represent one chemically stained image, which presents features distinct from another band, differently stained. To imitate these uncorrelated spectral bands, we set $p_{i,j,b} \sim \alpha^b \text{Uniform}[0, 1]$ independently, where $\alpha = 1.3$. Figure 6.6 shows one sample image in this setting. As before, the pixel's intensity increases with the spectral bands but the contrast between spectral bands is stronger than in the previous homogeneous dataset.

Results and Discussion

Figure 6.7 shows the error probability in log scale versus noise levels of the line selection scheme under different spatial priors. The legends are the



Gaussian spatial prior

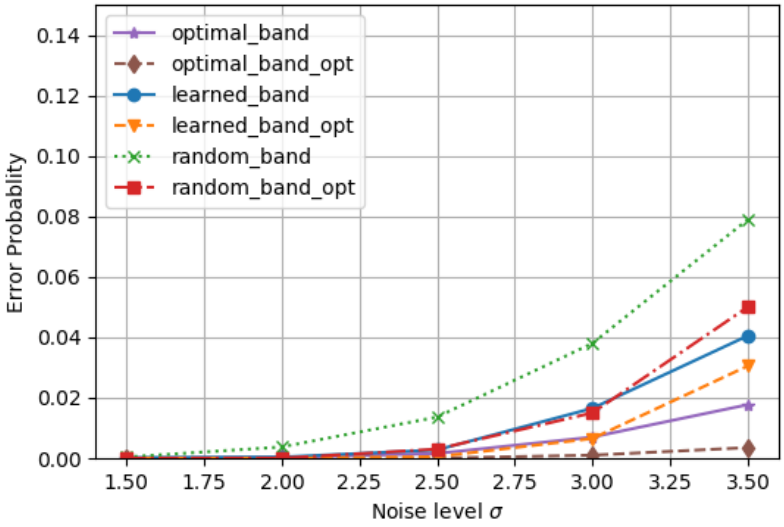


Uniform spatial prior

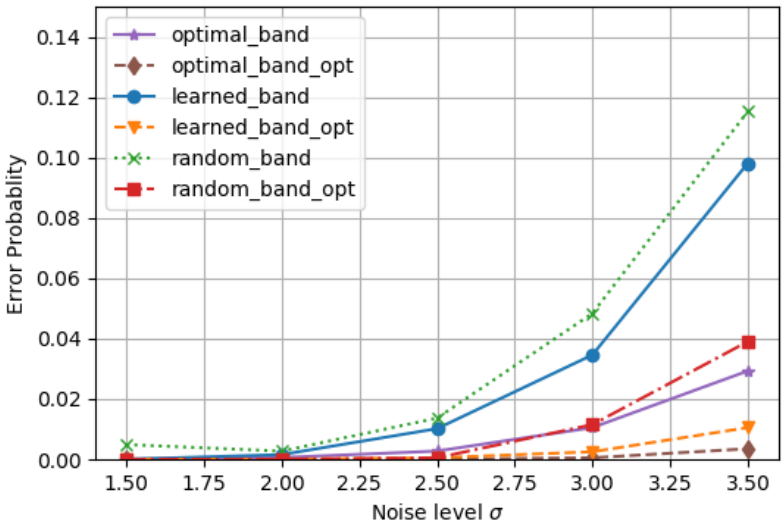
Figure 6.7: Error probability (in log scale) versus noise level using line selection as CS scheme and different inference methods. In general, the number of selected lines increases with the noise level. Top: the object location follows Gaussian spatial prior. Bottom: the object location follows uniform spatial prior.

same as in Figure 6.4. The learned line selection mask outperforms a random mask at all noise levels.

Figure 6.8 shows the error probability versus noise levels of the band



Gaussian spatial prior



Uniform spatial prior

Figure 6.8: Error probability versus noise level under different CS schemes and different inference methods. In general, the number of selected lines increases with the noise level. Top: the random object is generated under Gaussian spatial prior. Bottom: the random object is generated under uniform spatial prior.

selection scheme. We select optimal bands according to the SNR of each band. The legends in each subfigure are the same as in Figure 6.5. Figure 6.8 again shows that the learned band selection mask is superior to a

random mask. In the uniform spatial prior case, the performance of the DNN with the learned band selection is closer to that of a random band selection. But the performance of optimal decision rule with the learned band selection is almost as good as that of the optimal bands. This implies that the DNN is indeed learning which bands are better to scan with, and the "selection" becomes easier as the coefficient α increases and the SNR of each band has larger deviation.

Figures 6.9 and 6.10 further corroborate that the learned band mask in the DNN approaches the optimal selection. There are 8 subfigures in Figure 6.9 and the legends inside each subfigure are similar. For the top left subfigure, the two legends represent

1. `DNN: learned − optimal`: The difference between the prediction error rates of different DNNs trained using the learned band selection mask vs. using the optimal bands.

2. `DNN: random − learned`: The difference between the prediction error rates of different DNNs trained using a random band selection mask vs. using the learned bands.

In the top left subfigure of Figure 6.9, as the noise level σ increases, the difference between the error rates with a learned mask vs. a random mask increases, which implies that the DNN is able to learn a relatively effective band selection even in the high noise regime. At the same time, the difference between the error rates the learned mask vs. the optimal mask also increases, which means that in the high noise regime, the learned selection of spectral bands is less likely to approach the optimal band selection.

For the top right subfigure of Figure 6.9, the two legends are

1. `opt rule: learned − optimal`: The difference between the prediction error rates of the optimal decision rule using the learned band selection mask vs. the optimal bands.

2. `opt rule: random − learned`: The difference between the prediction error rates of the optimal decision rule using a random band selection mask vs. the optimal bands.

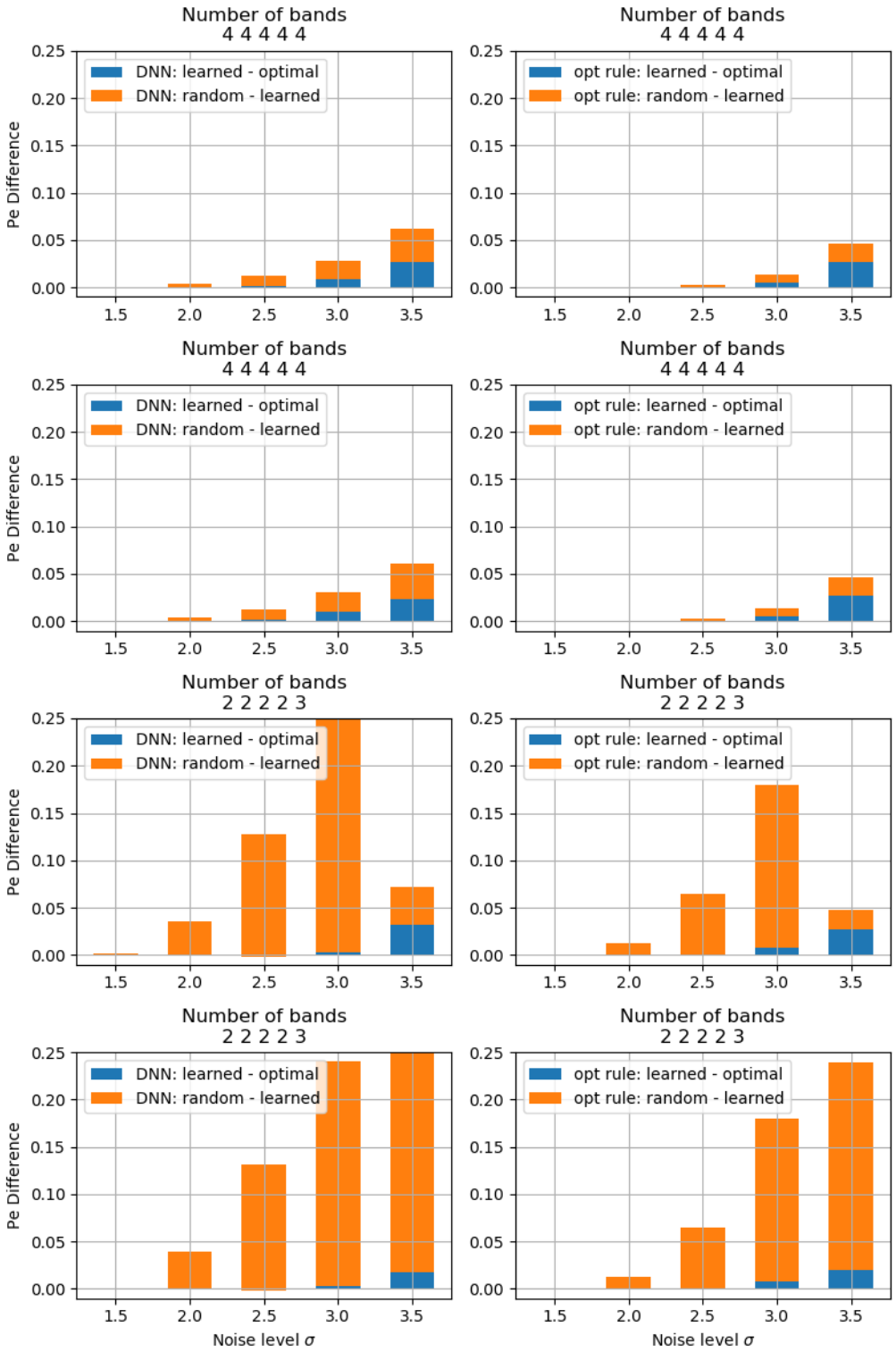


Figure 6.9: Difference of error probability using different classification methods and number of selected bands, under Gaussian spatial prior.

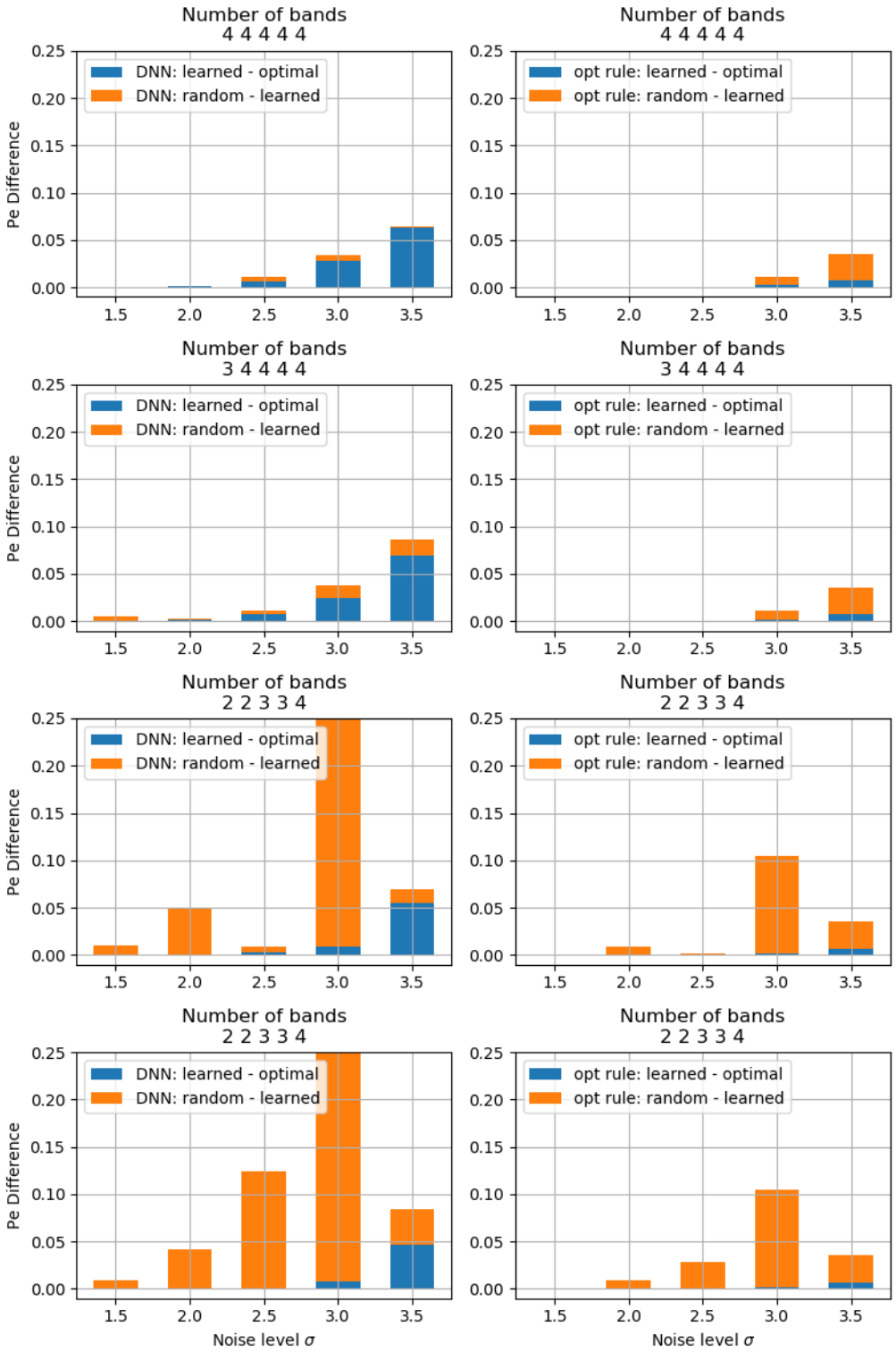


Figure 6.10: Difference of error probability using different classification methods and number of selected bands, under uniform spatial prior.

The trends of stacked bars are similar to the top left subfigure of Figure 6.9 and corroborate the effectiveness of the learned mask when used in conjunction with the optimal decision rule.

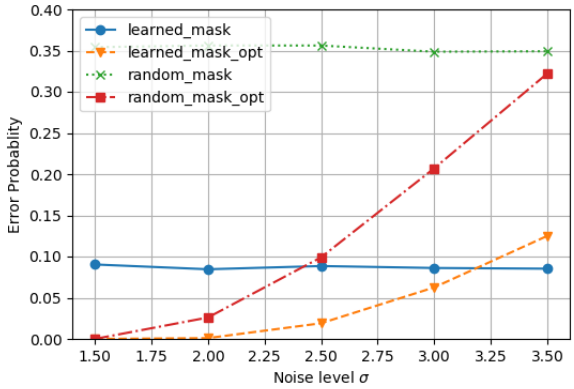
Each row of subfigures in Figure 6.9 represents the results obtained with the same setting for the hyperparameter λ_1 that controls the sparsity of the scanned bands, and hence they have the same number (listed in the title of the subfigure) of selected bands. In the left column of subfigures in Figure 6.9, the orange bar represents the prediction error rates difference between DNNs using a random and the learned selection mask, while the blue bar represents the difference between DNN using a learned and the optimal selection mask. The right column subfigures shows the corresponding differences in prediction error rates of the optimal decision rule that is adapted to the corresponding band selection mask.

Comparison of the four subfigures in the left column of Figure 6.9 reveals that as the number of selected bands decreases, the ratio of blue bar over the total bar (call it "blue ratio") often decreases under the same noise level. A lower blue ratio indicates that a learned band selection scheme is closer to the optimal bands. When the number of selected bands decreases, the orange bar increases dramatically and the blue ratio decreases accordingly, indicating that the DNN is selecting the spectral bands more effectively when the required number of measurements is smaller. This agrees with our intuition: When the budget of band scanning is small, a carefully chosen measurement based on the data distribution is favorable. Conversely, when the budget, or the number of measurements, is not limited, then the theory of CS guarantees perfect recovery with high probability.

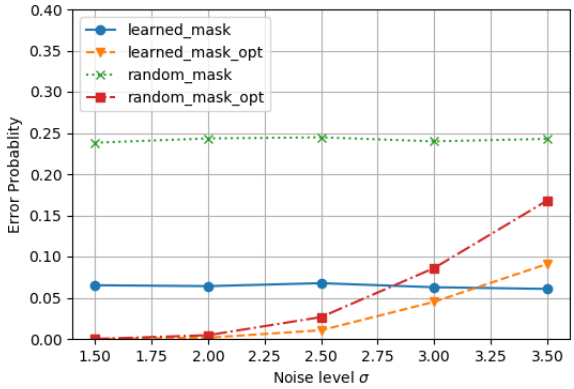
Figure 6.10 shows the corresponding results for the case that the object location follows the uniform distribution. In the top left subfigure, we can see that the blue ratio is very high, but in the top right subfigure, the optimal decision rule's error rate using learned bands is still very close to that using optimal bands. The trend of orange bars with respect to number of selected bands using the optimal decision rule is still the same.

Compressed Classification with the DD-CASSI CSI system

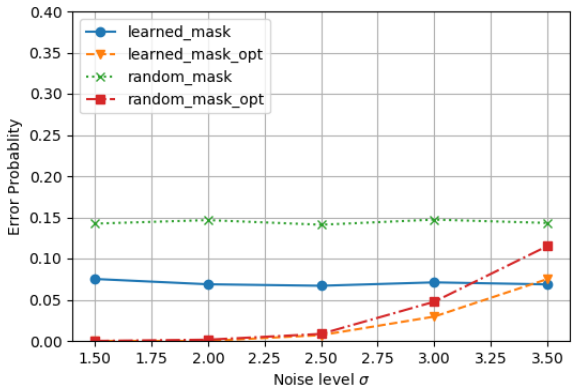
As an alternative to the simple band and line selection schemes, we use the DD-CASSI scheme (described in Section 3.3) to sense the image. We wish to determine whether the increased number of DOF available in the



$m = 1$



$m = 2$



$m = 4$

Figure 6.11: Error probability versus noise levels under DD-CASSI CS with a random mask or the learned masks. Top: number of snapshot $m = 1$. Middle: number of snapshot $m = 2$. Bottom: number of snapshots $m = 4$.

DD-CASSI scheme offers improved error rates. We train the DNN jointly with the linear CS layer of the DD-CASSI.

Recall (Section 4.2) that for DD-CASSI we do not implement the exact optimal decision rule, because we replace the matrix $(AA^T)^{-1}$ by the identity. We therefore refer to this decision rule as the suboptimal decision rule.

Figure 6.11 shows the error rates of the DNN and of the suboptimal decision rule versus noise levels using the DD-CASSI scheme with a random or the learned mask. Each subfigure shows the error rates with a different number of snapshot m. There are four legends in each subfigure.

1. `random_mask` or `learned_mask` : Prediction of the DNN with a random mask or with the learned mask.

2. `random_mask_opt` or `learned_mask_opt` : Prediction of the suboptimal decision rule with a random mask or the mask learned with the DNN.

In Figure 6.11, the mask learned with the DNN provides a substantial improvement over the random mask in the prediction of both the DNN and the suboptimal decision rule at all noise levels. This suggests that the DNN's output can be used as a metric for the mask's ability to extract classification information not only for this DNN, but also more generally. As could be expected, as the number of snapshots increases, the gap between the learned mask and a random mask in both the DNN and the optimal decision rule shrinks, implying that the higher the data compression ratio, the more effective the learned mask will be compared to a random mask.

Less expected is the near invariance of the error rates of the DNN to the noise level with both a random mask and the learned mask. One possible explanation might be that the architecture of the U-Net is not suitable to resolve the image compressed by the DD-CASSI system.

6.2 Compressed Segmentation

6.2.1 Segmentation for size varying object

We generate a synthetic segmentation dataset, where every multispectral image contains a size-varying object. Similar to the aforementioned

compressed classification dataset, each $N \times N$ pixel B-band multispectral image has a square object whose pixels are Gaussian independently distributed with the same per-band variance σ^2. The mean of square object are randomly chosen with equal probability $1/C$ from one of C fixed means, corresponding to the different classes and each square has size $M_c \times M_c$. The background pixel values are Gaussian independently distributed with the same per band variance σ^2, and correspond to the $(C+1)$th class. The object is placed at random so as to be fully contained in the background, in one of the $(N - M_c + 1)^2$ possible locations. Image pixels are therefore drawn from one of $C + 1$ classes. The segmentation task is to separate the object from the background and assign a label to each of the N^2 pixels to indicate their class.

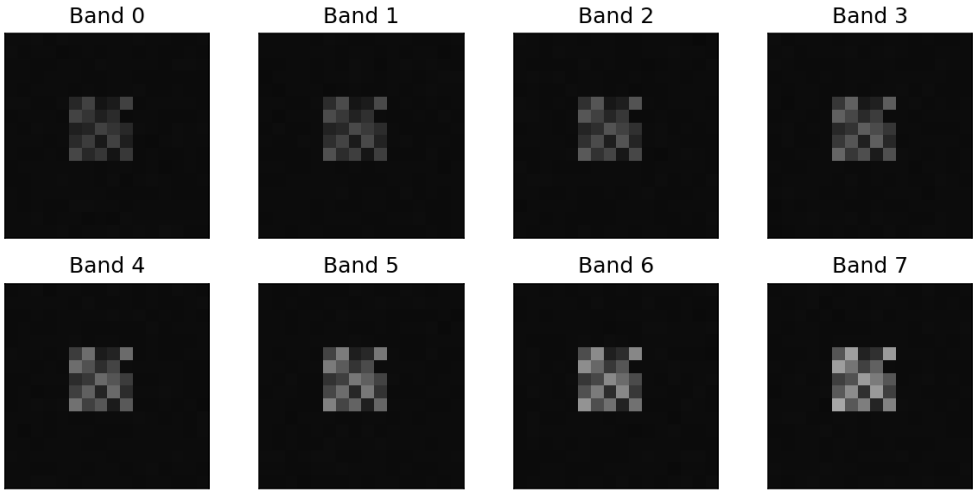
In this experiment, we set $N = 16, B = 8, C = 6, M_c$ varying from 5 to 10. We set the shape of the object the same in all spectral bands, but the intensity differs. We choose C random samples from $p_{i,j,0} \sim \text{Uniform}[0,1]^{M_c \times M_c}$, and set the other spectral bands as $p_{i,j,k} = \alpha^k p_{i,j,0}$ with $\alpha = 1.15$. This fixed set of C means is used for all data in all the experiments. Figure 6.12 shows two sample images with 8 spectral bands in the low noise regime.

Similar to compressed classification, we generate two datasets under Gaussian or uniform spatial distribution for the object location, each having 20,000 training images and 2,000 test images. We train the compressed segmentation network (Figure 4.1) using the Adam optimizer with learning rate 0.001. For a network with tunable CS operator A, we adopt a 2-stage training strategy as in the compressed classification network.

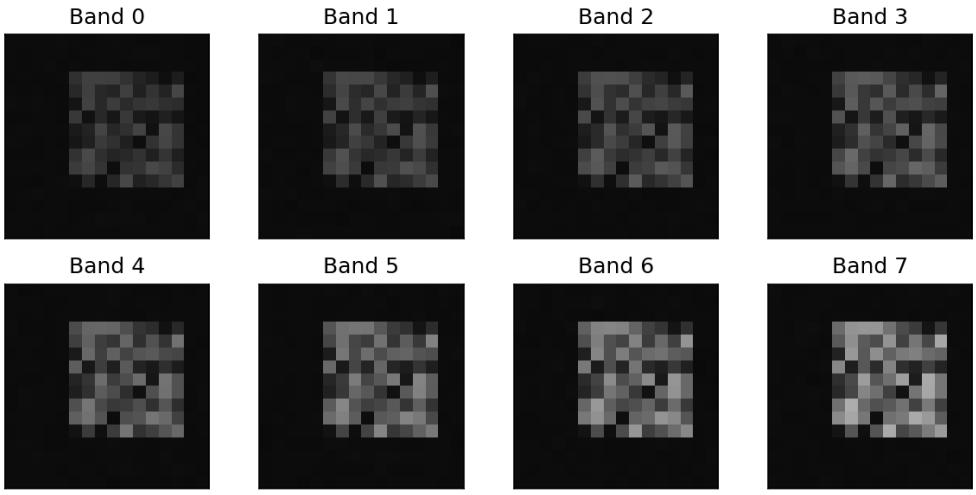
Results and Discussion

Figure 6.13 and Figure 6.14 shows the learned line selection masks similar to Figure 6.2, 6.3. Again, the selected lines are clustered in the center under the Gaussian distribution. The line gaps in the learned masks are slightly larger compared to the previous two figures, because the object size from some class are varying and the largest object side length in this dataset is 10.

We use three metrics defined in section 2.5 to evaluate the performance of the DNN and the optimal decision rule, namely, mean Intersection-Over-Union (mIOU), pixel-wise accuracy, and error probability. The third metric is not commonly used in segmentation tasks, but we use



(a) Sample image with 8 spectral bands



(b) Sample image with 8 spectral bands

Figure 6.12: Top: One sample image input of the compressive segmentation experiment, when $N = 16, B = 8, C = 5, M_c = 5, \sigma = 0.01$. Bottom: another sample image input $M_c = 9$.

this metric as a reference, since the optimal decision rule is supposed to yield a lower error probability compared to any other methods.

Figure 6.15 shows the mIOU, accuracy and error probability versus noise level σ using the line selection scheme. The legends in these figures are.

1. `random_line` or `learned_line`: Prediction of DNN using a random line selection scheme or the learned line selection scheme.

2. `random_line_opt` or `learned_line_opt`: Prediction of the optimal

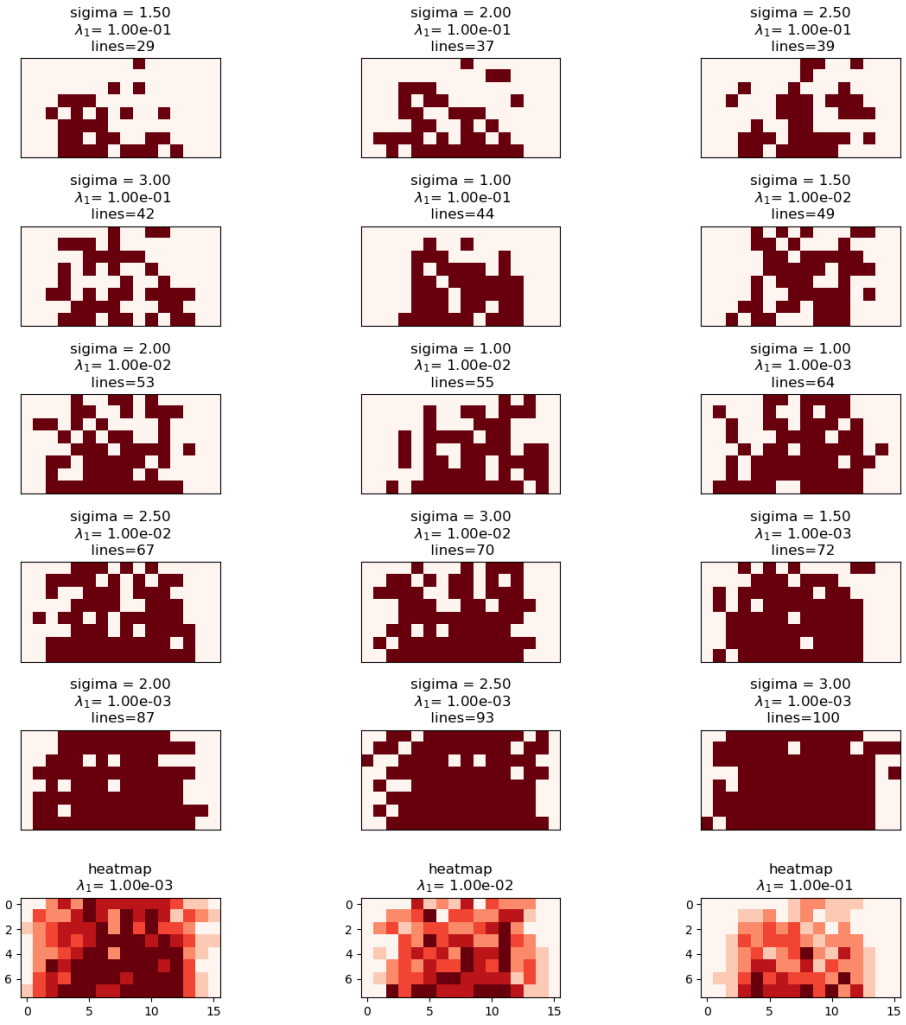


Figure 6.13: The binary masks of the learned line selection scheme under Gaussian distribution with different noise levels and λ_1s. Last row: the heatmap of selected lines.

decision rule, where the compressed data is obtained using a random line selection's mask or the learned line selection's mask.

The top and bottom row of subfigures of Figure 6.15 show results under Gaussian spatial prior and uniform spatial prior, respectively. Comparison of the top and bottom row reveals that the learned line selection scheme is able to improve the DNN's performance significantly on segmentation tasks, and this improvement is also observed in the optimal decision rule. Similar to classification, the difference between learned and random lines is larger under Gaussian spatial prior than uniform.

Figure 6.16 shows the mIOU, accuracy and error probability using the

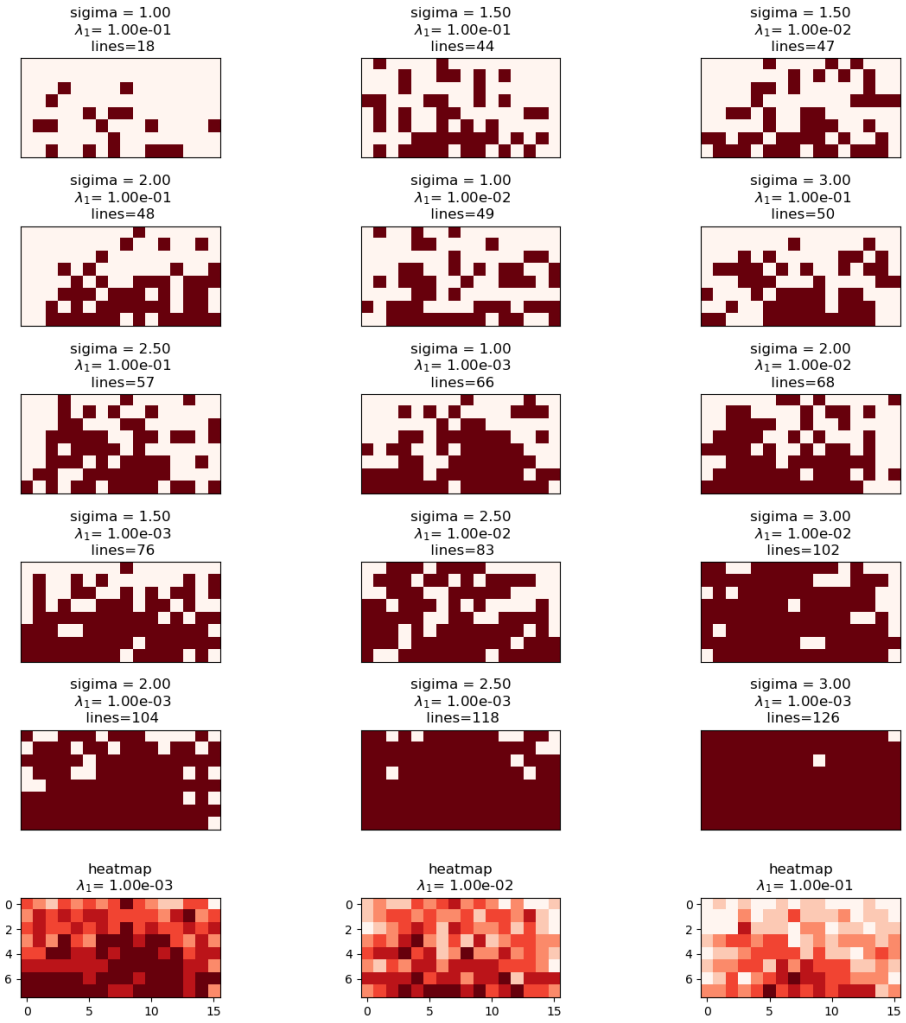


Figure 6.14: The binary masks of the learned line selection scheme under Gaussian distribution with different noise levels and λ_1s. Last row: the heatmap of selected lines.

band selection scheme. The legends in each subfigure are:

1. `random_band` or `learned_band`: Prediction of DNN using a random band selection scheme or the learned band selection scheme.

2. `random_band_opt` or `learned_band_opt`: Prediction of the optimal decision rule, where the compressed data is obtained using a random band selection's mask or the learned band selection's mask.

3. `optimal_band` or `optimal_band_opt`: Prediction of DNN or the prediction of the optimal decision rule using the optimal bands. The

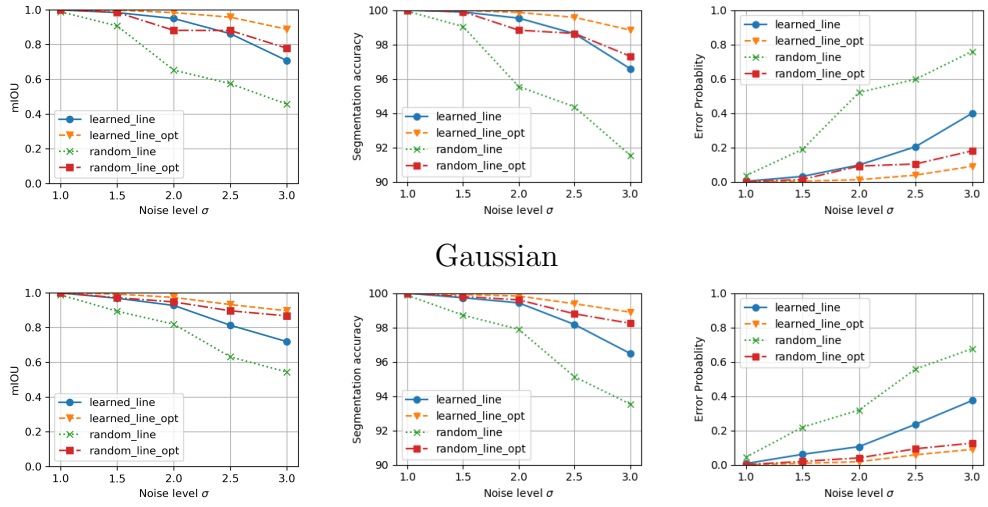


Figure 6.15: mIOU, accuracy and error probability versus noise levels using the line selection scheme when $\lambda_1 = 0.01$. Top row: mIOU, accuracy and error probability under Gaussian spatial prior. Bottom row: the same except under uniform spatial prior.

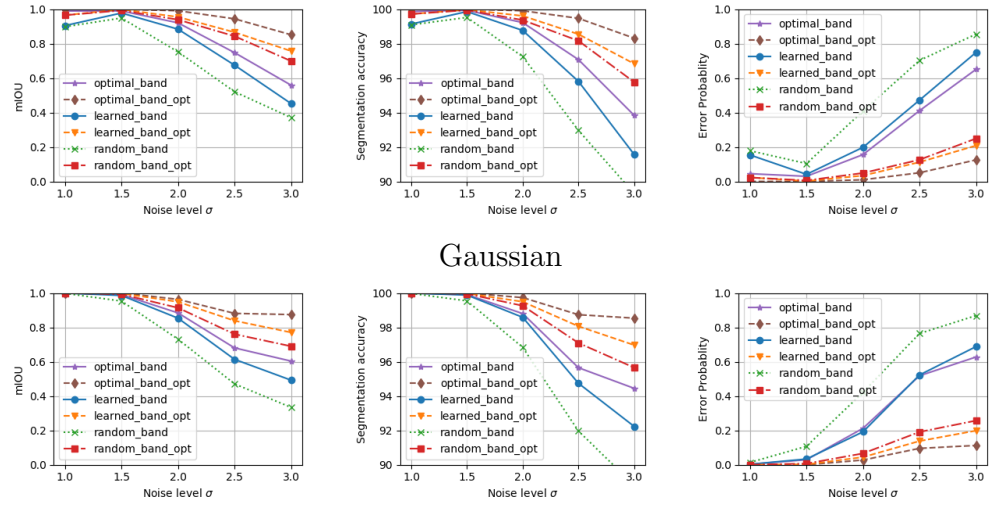


Figure 6.16: mIOU, accuracy and error probability versus noise levels using the band selection when $\lambda_1 = 0.01$. Top row: results under Gaussian spatial prior. Bottom row: results under uniform spatial prior.

'optimal' bands are defined as the spectral bands that have higher SNRs.

The top and bottom rows show the results under Gaussian spatial prior and

uniform spatial prior, respectively. Again, a learned CS scheme outperforms a random CS scheme. In terms of the error probability, the performance of a DNN combined with the learned band selection is very close to that of a DNN with the optimal bands, in both Gaussian and uniform spatial prior. This indicates that a learned band mask pursued by a DNN is indeed optimizing the bands using the deep segmentation network as a final metric.

6.2.2 Segmentation for simple squares

We generate another dataset where the images only contain a simple square. Each $N \times N$ pixel B-band multispectral image has an $M \times M$ ($M \ll N$) pixel square object whose pixels are Gaussian independently distributed with the same per-band variance σ^2 and a **common mean spectrum** $s_c \in \mathbb{R}^B$ drawn with equal probability $1/C$ from one of C fixed means, corresponding to the different classes. The background pixels are Gaussian independently distributed with the same per band variance σ^2, and correspond to a separate class. The object is placed at random so as to be fully contained in the background, in one of the $(N - M + 1)^2$ possible locations. Image pixels are therefore drawn from one of $C + 1$ classes.

Let c denote the class of the object, m denote the spatial location of the object, and $R_m \in \{0, 1\}^{N \times N}$ denote the coverage of the object, where 1 indicates the object and 0 indicates the background. Then the mathematical formulation of the synthetic image is:

$$\mathbf{x} = R_m \otimes s_c \sim P(m) \otimes \mathcal{N}(s_c, \sigma^2 I) \tag{6.3}$$

This problem can be treated as a classification problem, because there are in total $C(N - M + 1)^2$ possible images. Equation (4.19), the optimal decision rule of compressed classification, now becomes:

$$\begin{aligned} c^*, m^* = &\arg\min_{c,m} -2\sigma^2 \log(P(m)) \\ &+ (A\mathbf{x} - A(R_m \otimes s_c))^T (AA^T)^{-1} (A\mathbf{x} - A(R_m \otimes sc)) \end{aligned} \tag{6.4}$$

In this experiment, we set $N = 16, B = 8, C = 6, M = 6$ and use the same three metrics to evaluate the performance of the DNN and the optimal decision rule.

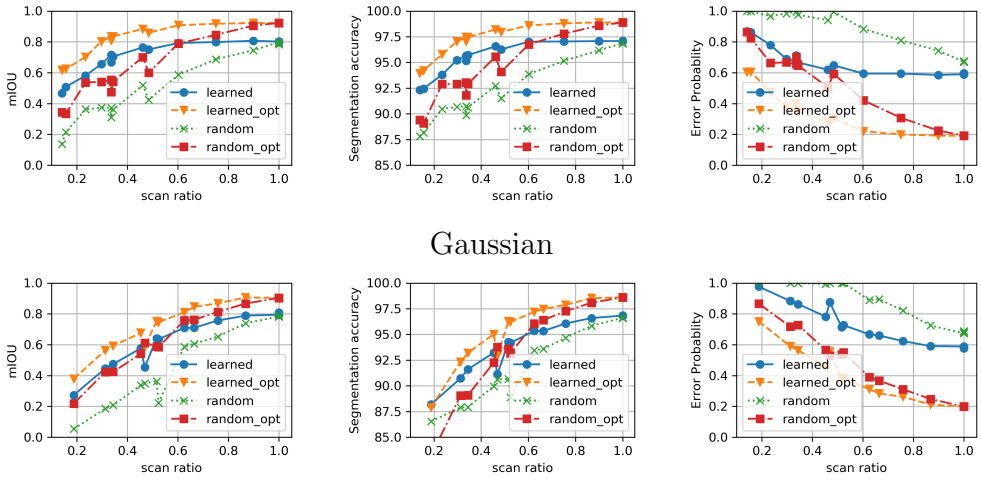


Figure 6.17: mIOU, accuracy and error probability versus scan ratios using the line selection scheme when $\sigma = 1.8$. Top row: mIOU, accuracy and error probability under Gaussian spatial prior. Bottom row: the same except under uniform spatial prior.

Figure 6.17 shows the mIOU, accuracy and error probability using the line selection scheme, when the scan ratio varies from 0.1 to 1. The legends in each subfigure are the same as in Figure 6.4. For both object location priors, the DNN classifies better using the learned line selection than a fixed random line selection at all scan ratios. But when the scan ratio reaches 1, the error rates of the DNN is much larger than that of the optimal decision rule while the mIOU and accuracy of the DNN and the optimal decision rule are close because these metrics do not have one-to-one correspondence. This observation indicates that this U-Net structure may not be the most suitable segmentation network for such a simple dataset.

Figure 6.18 shows the accuracy and mIOU of the band selection scheme. Here too we observe a slight improvement of the learned band selection scheme over a random one. Notice that the accuracy and mIOU in the band selection scheme are slightly lower than those of line selection due to fewer DOF in the former CS scheme.

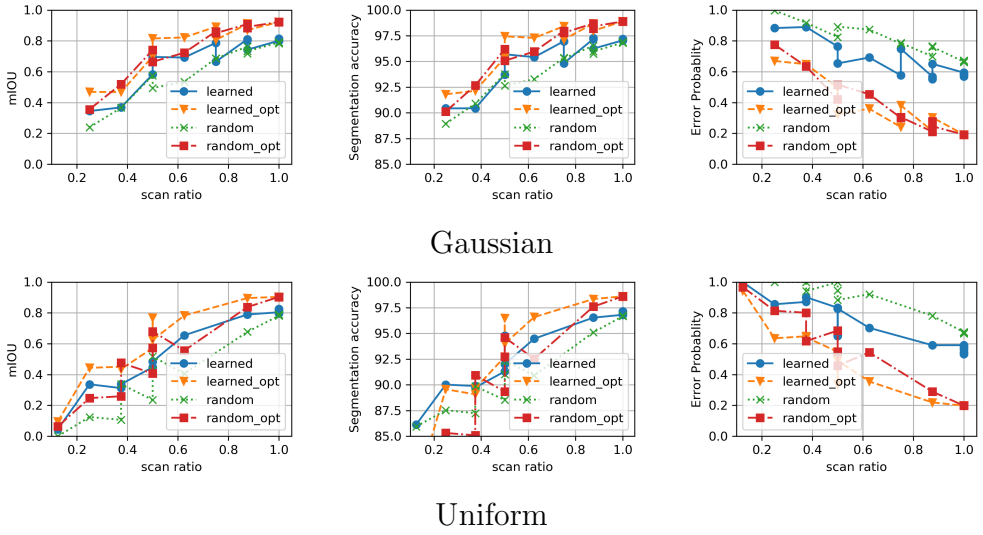


Figure 6.18: mIOU, accuracy and error probability versus scan ratios using the band selection scheme when $\sigma = 1.8$. Top row: mIOU, accuracy and error probability under Gaussian spatial prior. Bottom row: the same except under uniform spatial prior.

6.2.3 Summary

In this chapter, we optimize a physically constrained CS scheme together with the DNN for classification and semantic segmentation tasks. Using several synthetic datasets, we demonstrate that this deep neural network architecture is able to learn a CS acquisition mask that is not only better than a random mask that is often adopted in the compressed sensing literature, but is also interpretable. Furthermore, the DNN's prediction can be close to that of a known optimal decision rule when performing inference in the compressed domain.

CHAPTER 7

NUMERICAL EXPERIMENTS ON REAL MULTISPECTRAL DATASET

7.1 Tumor Biopsy Multispectral Dataset

In this chapter we report on the results of using the multispectral image segmentation scheme of the previous chapters to segment a clinical multispectral image data set. This tumor biopsy multispectral dataset is obtained using Fourier-transform infrared (FTIR) spectroscopic imaging. This dataset contains 99 19-spectral-band images with wavenumbers ranging from 984 cm^{-1} to 1765 cm^{-1}. Each image's size varies (approximately 700×700 pixels) and contains one disk-shaped tissue biopsy. After the IR images of biopsy are acquired, one oncologist expert labels a small fraction of the images with different cell types such as epithelium, stroma, and their subtypes. A single-pixel classifier is trained based on these small-amount labels, and this classifier is then applied to all images to create full labels for segmentation. There are in total seven semantic classes: benign epithelium, malignant epithelium, loose stroma, dense stroma, desmoplastic stroma, background and the other types of cells.

7.2 Benchmark Segmentation Network in the Data Domain

The goal of the segmentation task is to map each pixel of the biopsy image to its cell type label. We use a U-Net to train a segmentation network on fully scanned images to obtain a benchmark, and denote this trained U-Net as f_0. This fully scanned data is preprocessed by lower-clipping the voxel values to 0 and subsequently z-scored. The voxel values below 0 are due to measuring noise and the z-scoring is a common preprocessing step for

biometric data [68]. In the z-scoring step, the empirical mean and empirical standard deviation are calculated for each band separately because the tumor biopsy image has varying intensities at different wavelengths.

The U-Net structure is shown in Figure 4.3 and the detailed description can be found in Section 4.5.4.

7.3 Compressed Segmentation Networks

The line and band selection compressed sensing schemes, introduced in Section 3.2, are used to compressively sense the images. The compressed segmentation network is shown in Figure 4.1, where the segmentation network f_θ is a U-Net and the loss function of the line and band selection schemes is (4.31). We empirically choose the hyperparameter λ_1 in (4.31) to range from 0.01 to 100.

We adopt a two-stage training strategy for tunable sensing matrix $A(w)$: initialize the binary parameter w^b randomly such that its sparsity (number of 1s / total elements) is 0.5, train the segmentation network f_θ jointly with the parameter w using the Adam optimizer with learning rate 0.001, for 150 epochs, then fix w and train the U-Net f_θ for another 150 epochs. In order to compare the effects of a learned sensing scheme versus a random one, we train the same network but with a random fixed w. For a fair comparison, the U-Net is trained with the same setting for 150 epochs for a random sensing scheme.

To compare with the traditional reconstruction + inference pipeline (called "composite network"), we used the composite network described in Section 4.6.1, and its block diagram is shown in Figure 4.4. Similar to the compressed segmentation network, we train g_θ of a tunable and a fixed sensing matrix $A(w)$ with the Adam optimizer and learning rate 0.001 for 300 epochs and 150 epochs, respectively.

We also use MTL network, introduced in Section 4.6.2, to predict the semantic labels. Figure 4.5 shows the block diagram of the MTL network. We train the h_ζ, f_θ of tunable and fixed $A(w)$ with the same optimizer and learning rate for 300 epochs and 150 epochs, respectively.

7.3.1 Preliminary results

We conduct experiments on the CS segmentation network on 16×16 small patches. We divide 99 tumor cell images to a 80-image training set and a 19-image test set. Then we extract 10,000 $16 \times 16 \times 19$ patches from the training set and 1,000 patches of the same size from the test set. In the following, we use pixel-wise segmentation accuracy and mean IoU to evaluate the performance of each network architecture.

Figure 7.1 shows the preliminary results of line and band selection schemes of the compressed segmentation network. When the scan ratio reaches 0.3, the learned band selection is able to perform as well as fully scanned data. However, the accuracy or mIOU is not ideal for practical purposes.

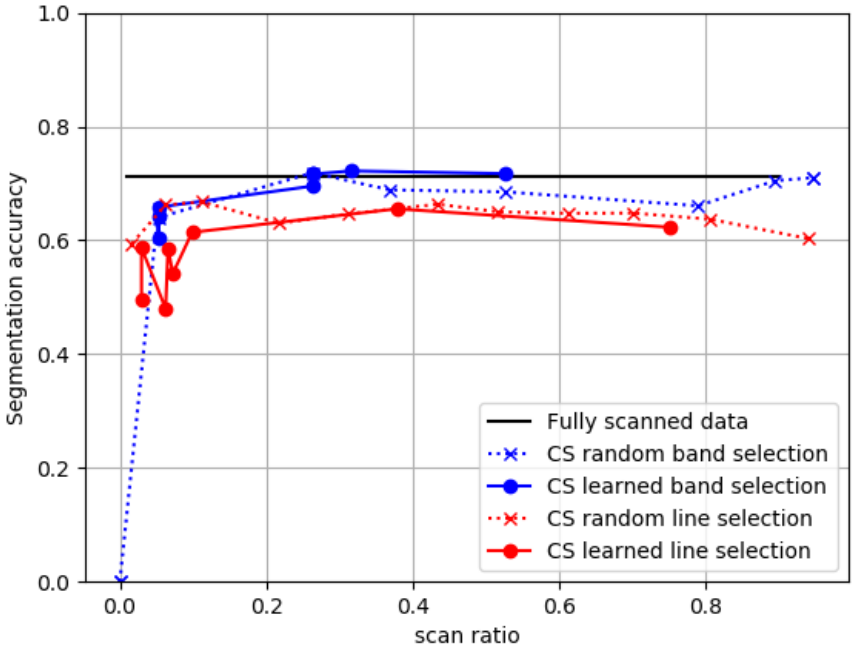
Figure 7.2 and 7.3 show two segmentation examples of the test images using compressed segmentation network with the learned band selection. The test image is divided into small non-overlapping patches and the network provides the predicted semantics for each patch. The final segmentation of the image simply stitches all the patches together. We can see there are some artifacts around the edge of patches, yielding a grid-like artifact. This motivates us to use larger patches to reduce the discontinuity around the edge.

7.4 Results and Evaluations

To overcome the artifacts due to small patch size, We divide these 99 images to an 80-image training set and a 19-image test set, and extract 10,000 $64 \times 64 \times 19$ patches from the training set and 1,000 patches of the same size from the test set. The 64×64 patch captures more complex image structures. In this section, we present results on the compressed segmentation network, the composite network and the MTL network.

Figures 7.4 and 7.5 show two examples of segmentation output using compressed segmentation network with the line selection scheme. The grid-like artifact observed in Figure 7.2 and 7.3 is nearly invisible and the trained compressed segmentation network for larger patch is able to produce smoother boundaries between cell types.

Figure 7.6 shows the accuracy and mIOU of each network versus different



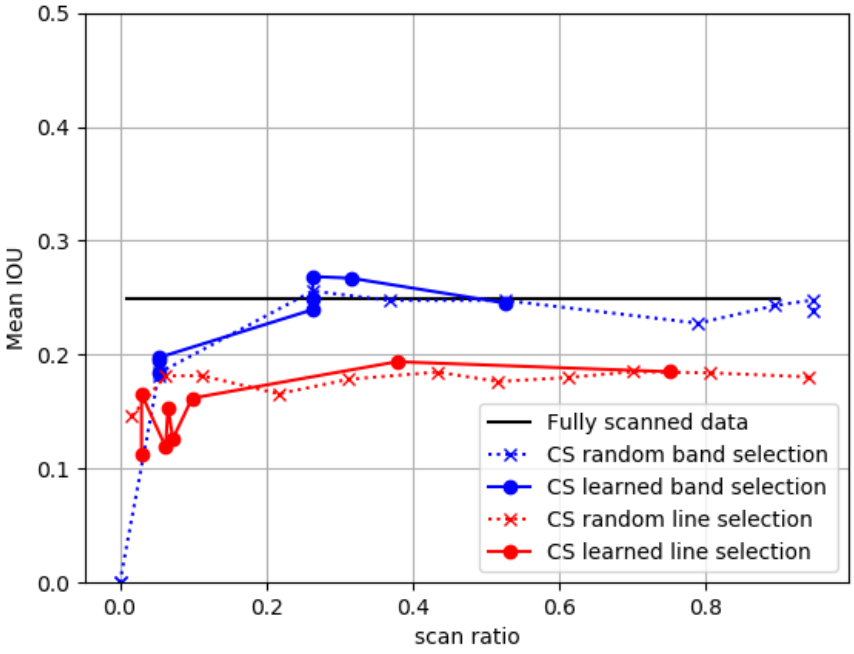


Figure 7.1: Segmentation accuracy (Top) and mean IoU (Bottom) of the trained network using the learned or a random band selection and the line selection CS scheme.

scan ratios for the band selection CS scheme: The black line is the benchmark accuracy of a segmentation U-Net trained on fully scanned data. The blue dotted line and blue solid line represent a random and the learned band selection scheme for MTL network, respectively. The green dotted line and green solid line represent a random and the learned band selection compressed segmentation network, respectively. The red dotted line and red solid line are for the composite network. As the scan ratio increases, it is natural that the accuracy of all three kinds of networks approaches that of the fully scanned data. From this figure, MTL network performs the best because it eavesdrops the ground truth image. The compressed segmentation network performs nearly the same as the MTL network (accuracy difference $< 5\%$ for the learned band selection and $< 1\%$ for random band selection). Composite segmentation network performs significantly worse as the reconstruction network's goal is not aligned with the ultimate goal of segmentation. The accuracy of the composite network drops around 5% for both the learned and a random band selection for scan ratio ranging from 0.1 to 0.8. Finally, we can see that the learned band selection, at most times, is slightly better than a random band selection.

Figure 7.7 shows the accuracy and mIOU of three networks for line selection compressed scheme. The color codes are the same and we also observe the ranking of accuracy of the networks is the same. On average, MTL and CS network accuracies exceed that of the composite network by around 3%. Also, the learned line selection scheme is always better than a random line scheme when the scan ratio exceeds 0.2.

7.5 Summary

In this chapter, we optimize a physically constrained CS scheme together with the DNN for semantic segmentation tasks on a real tumor biopsy dataset. With three different network architectures, we demonstrate that the physically constrained CS acquisition scheme, jointly with network training, can be improved more than a random acquisition scheme. We also test the effect of the patch size of the image in segmentation and verify that a larger patch size helps to capture more complex image structures.

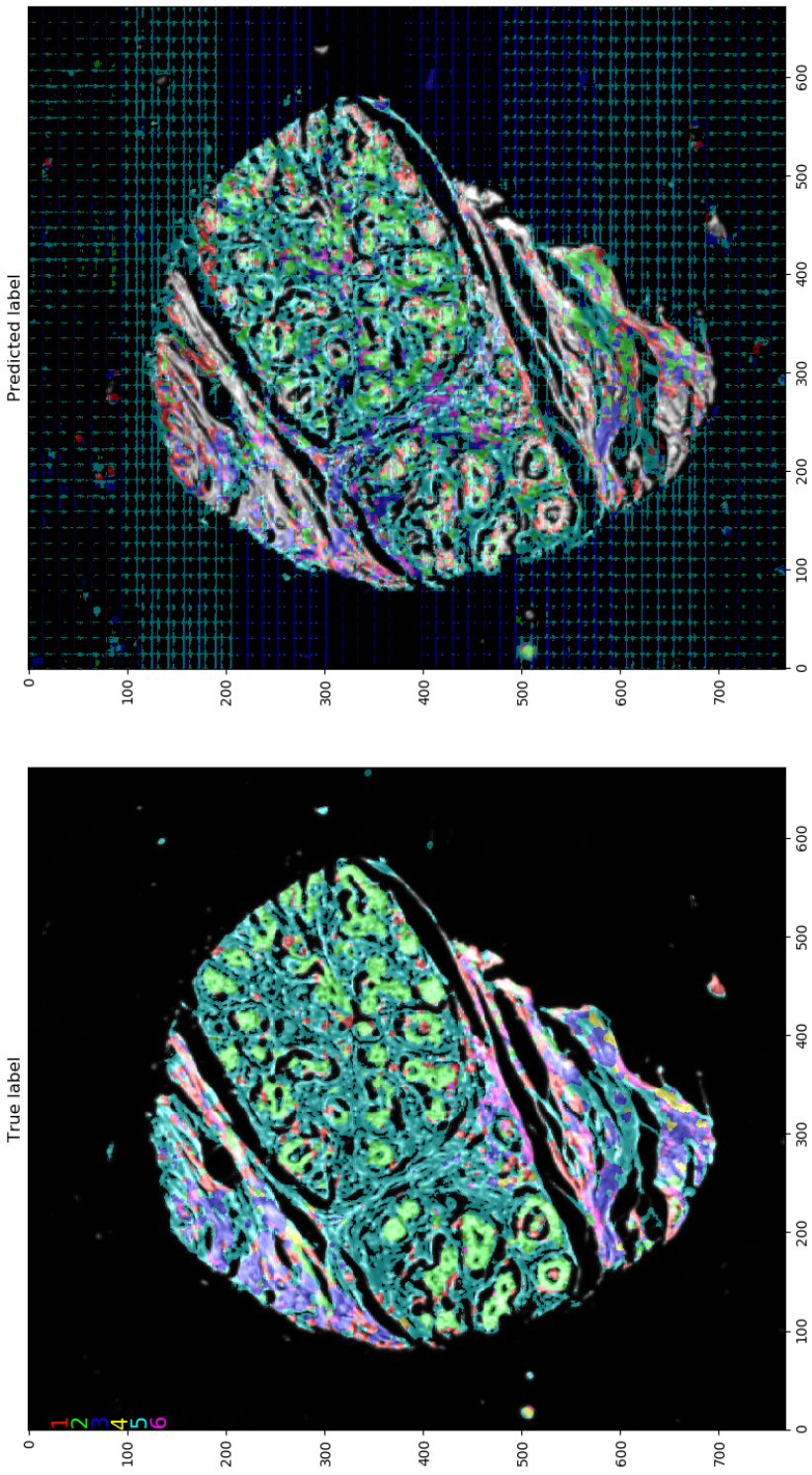


Figure 7.2: One example of tumor cell image segmentation. Left: ground truth. Right: predicted labels.

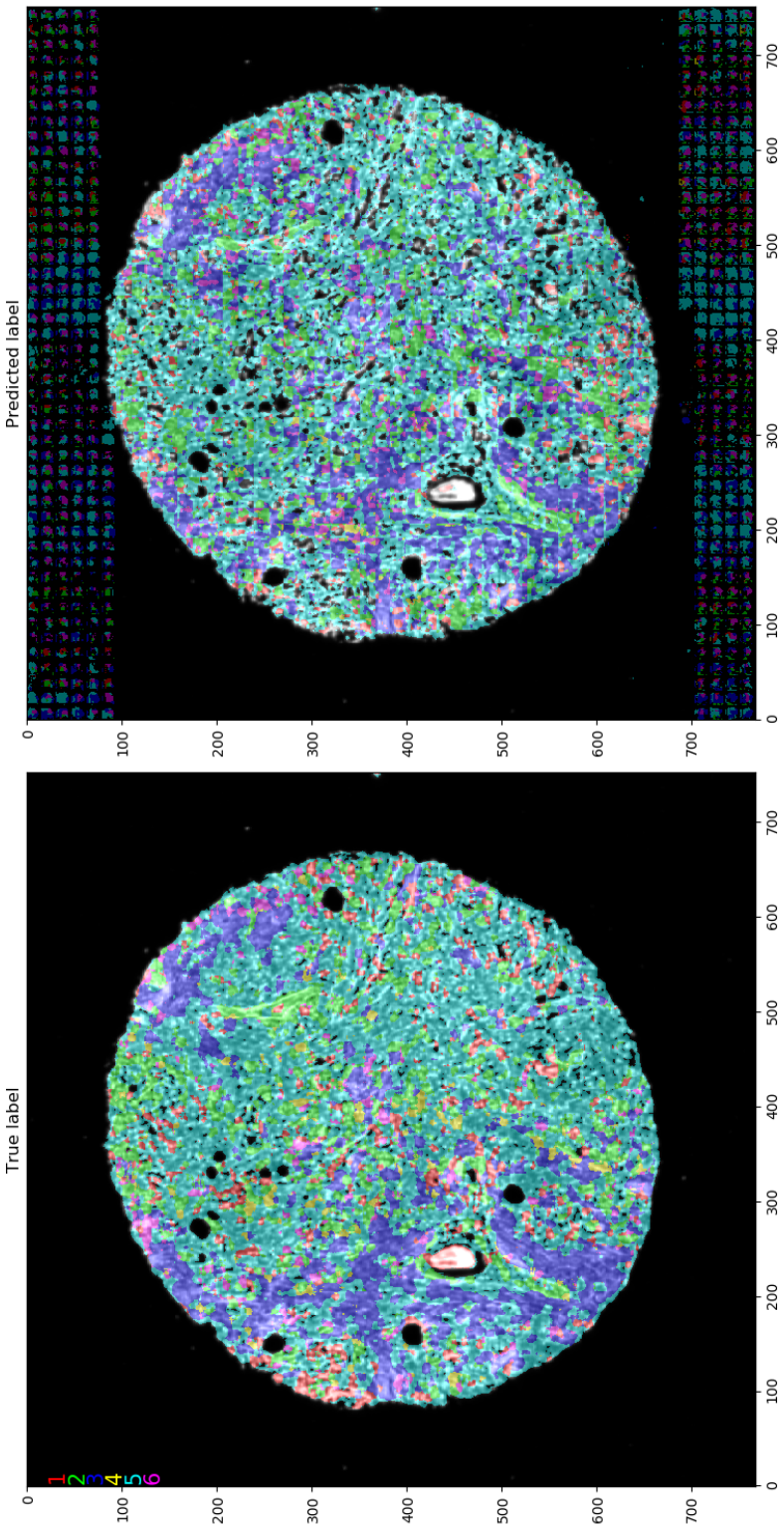


Figure 7.3: Another example of tumor cell image segmentation. Left: ground truth. Right: predicted labels.

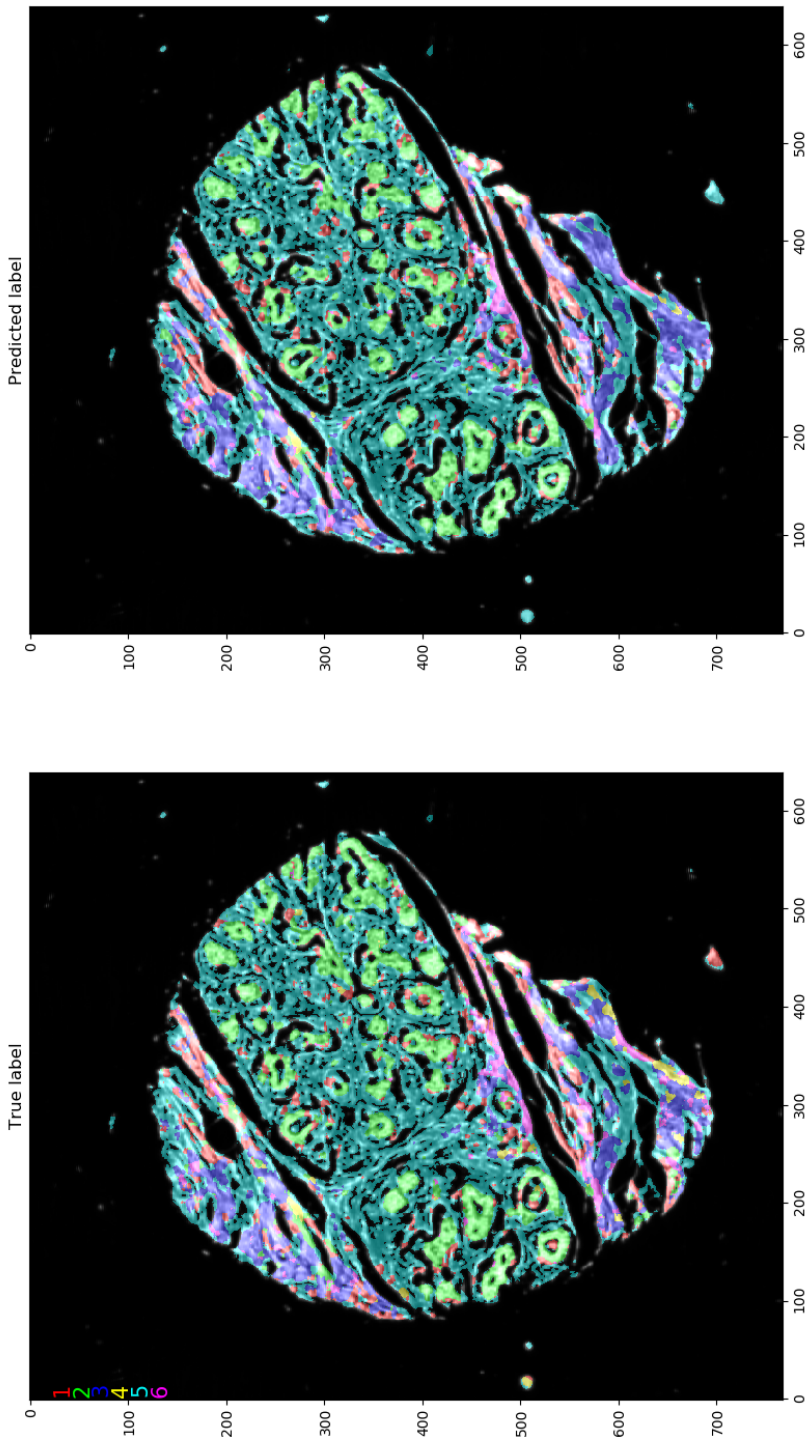


Figure 7.4: One example of tumor cell image segmentation. Left: ground truth. Right: predicted labels.

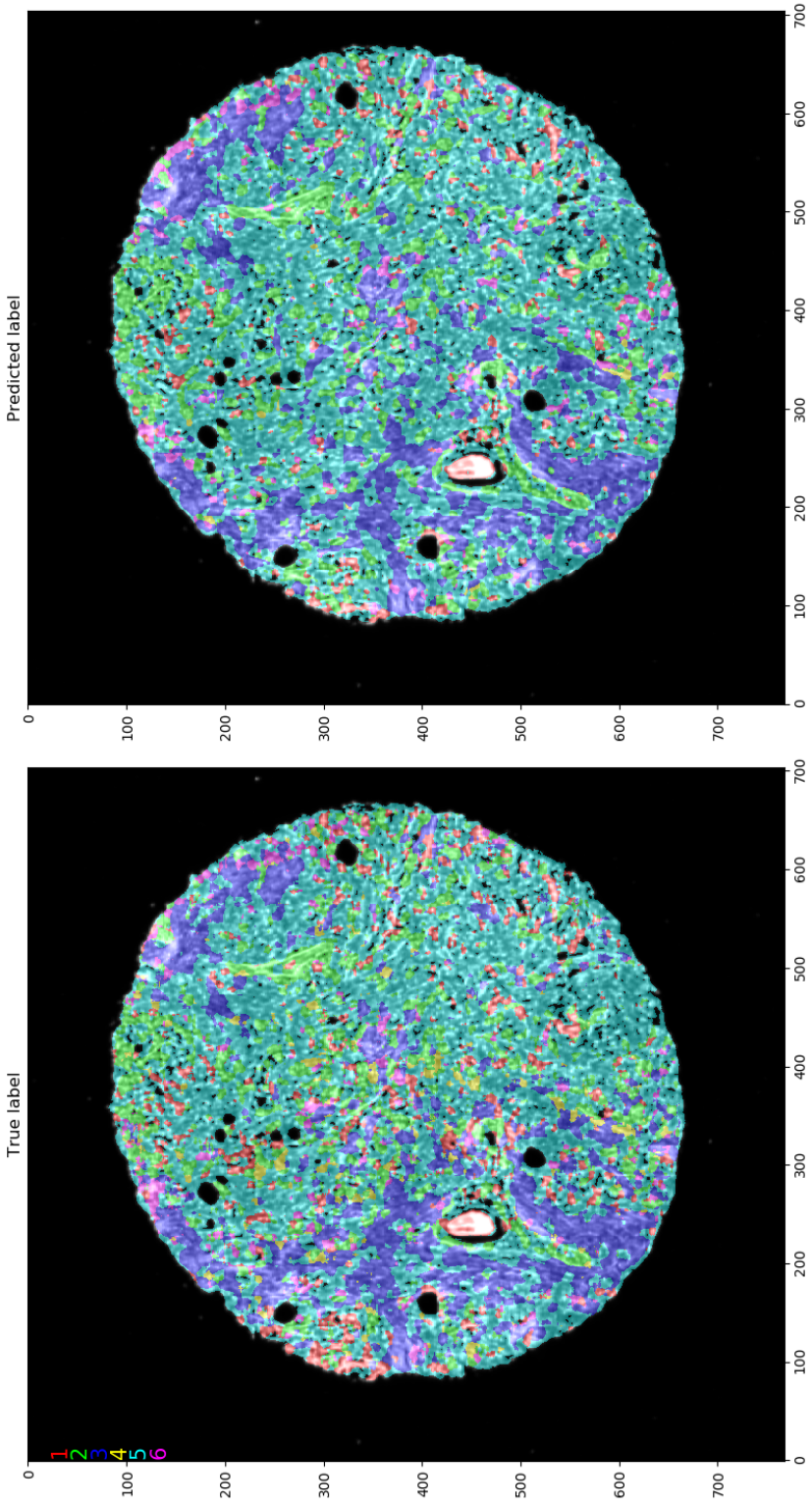


Figure 7.5: Another example of tumor cell image segmentation. Left: ground truth. Right: predicted labels.

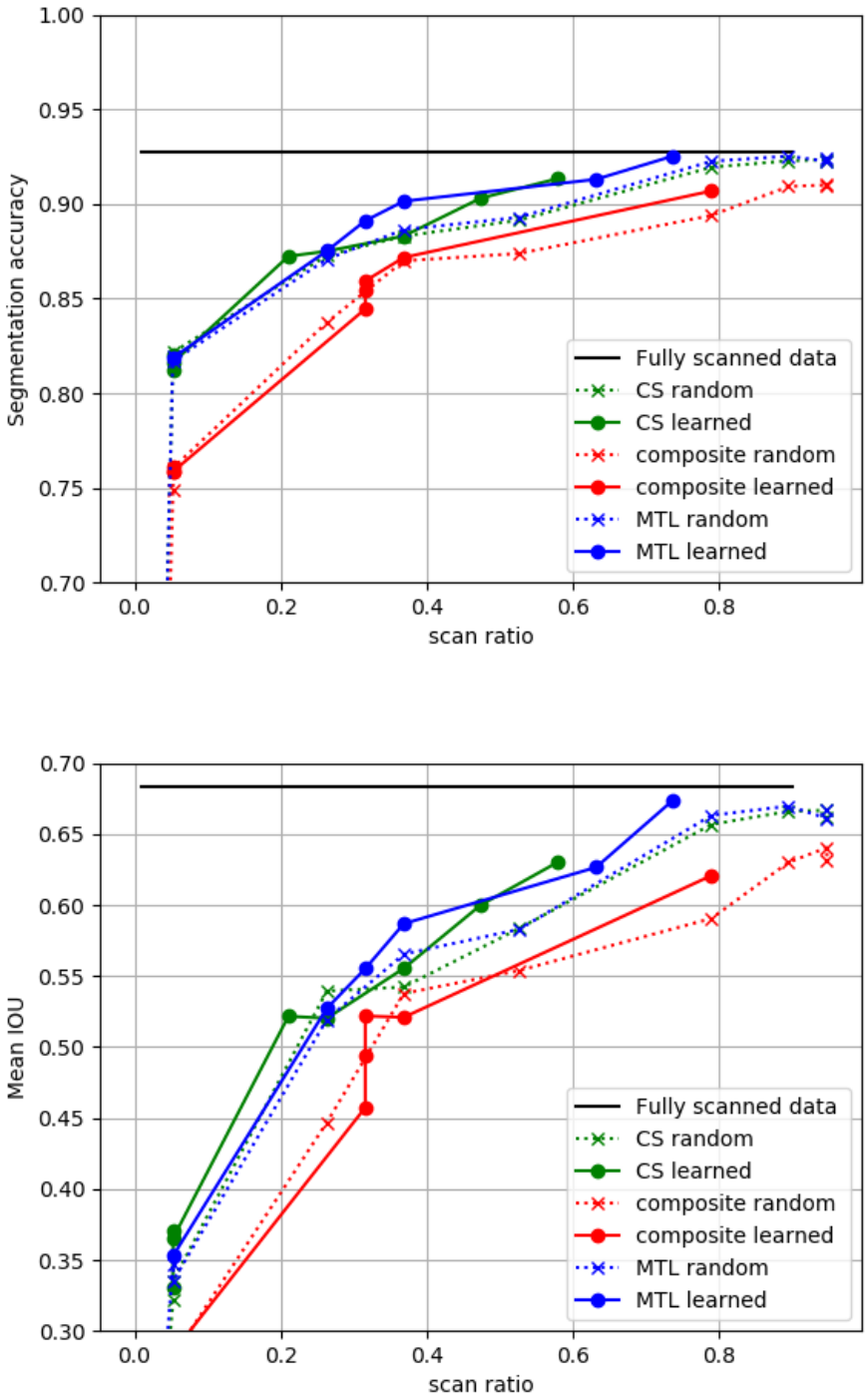


Figure 7.6: Segmentation accuracy (Top) and mean IoU (Bottom) of different networks with learned or random masks in band selection compressed sensing scheme.

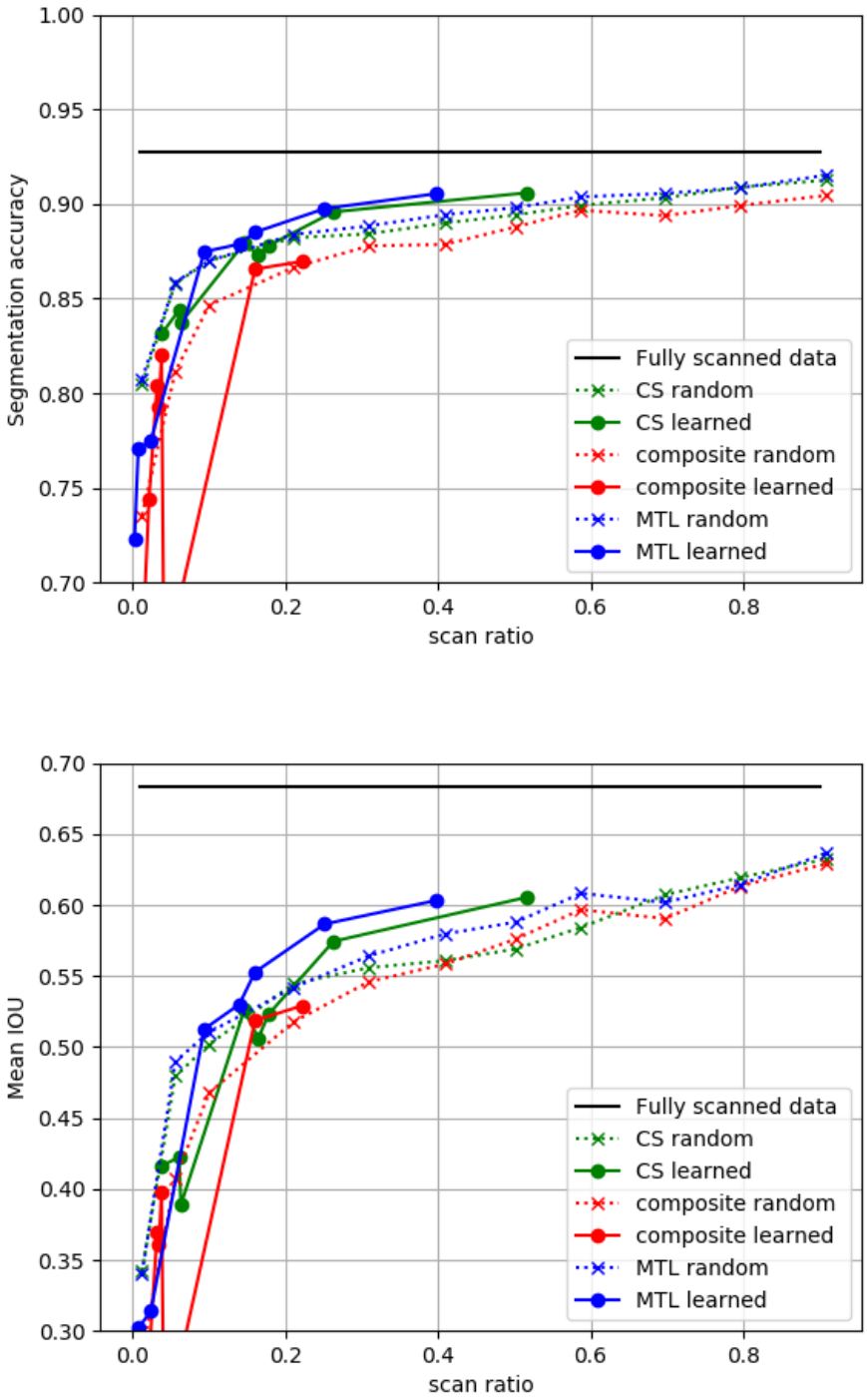


Figure 7.7: Segmentation accuracy (Top) and mean IoU (Bottom) of different networks with learned or random masks in line selection compressed sensing scheme.

CHAPTER 8

CONCLUSION

In this thesis, we study performing inference in compressed domain for multispectral images and optimizing optical acquisition schemes jointly with the inference network. We first introduce the compressed sensing (CS) theory which is the foundation of CS imaging systems for multispectral images. Since using the traditional CS imaging system, the reconstruction and prediction pipeline is wasteful in terms of computation power and the prediction accuracy strongly depends on the quality of the reconstruction, we explored the possibility of making inference directly on the compressed domain for realizable optical CS imaging systems for multispectral images.

Next, we present the optimal decision rule to perform compressed detection, classification and segmentation in a simple data setting. To compare a learned neural network performance with the optimal decision rule, we propose three neural network architectures that are guaranteed to achieve the optimal decision rule. We compare the performance of a neural network with the optimal decision rule in the data domain with a synthetic dataset. We experimentally show the neural networks' capability of learning the optimal weights and show the lower bound of the sample complexity of each network architecture empirically.

Finally, we connect the existing realizable optical compressed sensing imaging systems designed for multispectral images and their forward process of data acquisition to a deep neural network. We optimize the physically constrained CS scheme together with the DNN for classification and semantic segmentation tasks and compare the performance of DNN to that of the optimal decision rule. We show the DNN is able to learn an interpretable CS acquisition mask and the performance of DNN is close to that of the optimal decision rule with moderate compression ratio on multiple synthetic datasets. We verify the improvement of the acquisition scheme over a random one on a tumor biopsy dataset.

Several interesting questions are still left open for future work. One challenging problem lies in the fundamental information limits of the CSI system in inference tasks. It would be interesting to study the fundamental bounds of the inference performance versus the number of measurements and the complexity of the inference task. In addition, we are also interested in the difference between the learned acquisition scheme from DNN and the underlying truly optimal acquisition scheme.

CHAPTER 9

REFERENCES

[1] T. Mewes, J. Franke, and G. Menz, "Spectral requirements on airborne hyperspectral remote sensing data for wheat disease detection," *Precision Agriculture*, vol. 12, no. 6, p. 795, 2011.

[2] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose et al., "Agriculture-vision: A large aerial image database for agricultural pattern analysis," *arXiv preprint arXiv:2001.01306*, 2020.

[3] G. Lu and B. Fei, "Medical hyperspectral imaging: A review," *Journal of Biomedical Optics*, vol. 19, no. 1, p. 010901, 2014.

[4] Y. Knyazikhin, M. A. Schull, P. Stenberg, M. Mõttus, M. Rautiainen, Y. Yang, A. Marshak, P. L. Carmona, R. K. Kaufmann, P. Lewis et al., "Hyperspectral remote sensing of foliar nitrogen content," *Proceedings of the National Academy of Sciences*, vol. 110, no. 3, pp. E185–E192, 2013.

[5] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Springer Science & Business Media, 2003.

[6] R. M. Willett, R. F. Marcia, and J. M. Nichols, "Compressed sensing for practical optical imaging systems: a tutorial," *Optical Engineering*, vol. 50, no. 7, p. 072601, 2011.

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

[8] Y. August, C. Vachman, Y. Rivenson, and A. Stern, "Compressive hyperspectral imaging by random separable projections in both the spatial and the spectral domains," *Applied Optics*, vol. 52, no. 10, pp. D46–D54, 2013.

[9] 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*, vol. 25, no. 2, pp. 83–91, 2008.

[10] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling [a sensing/sampling paradigm that goes against the common knowledge in data acquisition]," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[11] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[12] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[13] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[17] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2007, pp. 801–808.

[18] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[20] H. Arguello and G. R. Arce, "Colored coded aperture design by concentration of measure in compressive spectral imaging," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1896–1908, 2014.

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

[22] S. Oymak and J. A. Tropp, "Universality laws for randomized dimension reduction, with applications," *Information and Inference: A Journal of the IMA*, vol. 7, no. 3, pp. 337–446, 2018.

[23] C. Hegde, A. C. Sankaranarayanan, W. Yin, and R. G. Baraniuk, "Numax: A convex approach for learning near-isometric linear embeddings," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6109–6121, 2015.

[24] Y. Li, C. Hegde, A. C. Sankaranarayanan, R. Baraniuk, and K. F. Kelly, "Compressive image acquisition and classification via secant projections," *Journal of Optics*, vol. 17, no. 6, p. 065701, 2015.

[25] X. Ma, H. Zhang, X. Ma, G. R. Arce, T. Xu, and T. Mao, "Snapshot compressive spectral imaging based on adaptive coded apertures," in *Compressive Sensing VII: From Diverse Modalities to Big Data Analytics*, vol. 10658. International Society for Optics and Photonics, 2018, p. 1065803.

[26] L. Wang, T. Zhang, Y. Fu, and H. Huang, "HyperReconNet: Joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2257–2270, 2018.

[27] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction—Does AI come at a cost?" *arXiv preprint arXiv:1902.05300*, 2019.

[28] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," *preprint*, 2009.

[29] M. A. Davenport, P. Boufounos, M. B. Wakin, R. G. Baraniuk et al., "Signal processing with compressive measurements." *J. Sel. Topics Signal Processing*, vol. 4, no. 2, pp. 445–460, 2010.

[30] R. J. Durrant and A. Kabán, "A tight bound on the performance of Fisher's linear discriminant in randomly projected data spaces," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 911–919, 2012.

[31] R. Durrant and A. Kabán, "Sharp generalization error bounds for randomly-projected classifiers," in *International Conference on Machine Learning*, 2013, pp. 693–701.

[32] T. Wimalajeewa, H. Chen, and P. K. Varshney, "Performance limits of compressive sensing-based signal classification," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2758–2770, 2012.

[33] S. Lohit, K. Kulkarni, P. Turaga, J. Wang, and A. C. Sankaranarayanan, "Reconstruction-free inference on compressive measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 16–24.

[34] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[35] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[36] S. Elmalem, R. Giryes, and E. Marom, "Learned phase coded aperture for the benefit of depth of field extension," *Optics Express*, vol. 26, no. 12, pp. 15 316–15 331, 2018.

[37] Y. F. Cheng, M. Strachan, Z. Weiss, M. Deb, D. Carone, and V. Ganapati, "Illumination pattern design with deep learning for single-shot Fourier ptychographic microscopy," *Optics Express*, vol. 27, no. 2, pp. 644–656, 2019.

[38] B. Wen, S. Ravishankar, L. Pfister, and Y. Bresler, "Transform learning for magnetic resonance image reconstruction: From model-based learning to building neural networks," *IEEE Signal processing Magazine*, vol. 37, no. 1, pp. 41–53, 2020.

[39] S. Ravishankar and Y. Bresler, "Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015.

[40] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X.-P. Zhang, "Bayesian nonparametric dictionary learning for compressed sensing MRI," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5007–5019, 2014.

[41] W. Xu and B. Hassibi, "Compressed sensing over the Grassmann manifold: A unified analytical framework," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 562–567.

[42] M. A. Iwen, F. Krahmer, S. Krause-Solberg, and J. Maly, "On recovery guarantees for one-bit compressed sensing on manifolds," *arXiv preprint arXiv:1807.06490*, 2018.

[43] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall PTR, 1993.

[44] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015.

[45] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, 2018, pp. 8778–8788.

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

[47] R. F. Marcia and R. M. Willett, "Compressive coded aperture superresolution image reconstruction," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 833–836.

[48] X. Lin, Y. Liu, J. Wu, and Q. Dai, "Spatial-spectral encoded compressive hyperspectral imaging," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–11, 2014.

[49] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz, "Single-shot compressive spectral imaging with a dual-disperser architecture," *Optics Express*, vol. 15, no. 21, pp. 14 013–14 027, 2007.

[50] B. Guo, S. R. Gunn, R. I. Damper, and J. D. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 522–526, 2006.

[51] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158–4171, 2007.

[52] R. Mankar, M. J. Walsh, R. Bhargava, S. Prasad, and D. Mayerich, "Selecting optimal features from Fourier transform infrared spectroscopy for discrete-frequency imaging," *Analyst*, vol. 143, no. 5, pp. 1147–1156, 2018.

[53] L. Pfister and K. Falahkheirkhah, personal communication, 2019/11.

[54] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[55] M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, "Detection and estimation with compressive measurements," Dept. of ECE, Rice University, Tech. Rep., 2006.

[56] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[57] N. Ailon and B. Chazelle, "The fast Johnson–Lindenstrauss transform and approximate nearest neighbors," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 302–322, 2009.

[58] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," International Computer Science Institute, Tech. Rep., 1999.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[60] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes," *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, pp. 2663–2674, 2018.

[61] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems*, 2015, pp. 3123–3131.

[62] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[63] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[64] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[65] Y. Takai, A. Sannai, and M. Cordonnier, "On the number of linear functions composing deep neural network: Towards a refined definition of neural networks complexity," *arXiv preprint arXiv:2010.12125*, 2020.

[66] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[67] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[68] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[69] S.-H. Chang, P. C. Cosman, and L. B. Milstein, "Chernoff-type bounds for the Gaussian error function," *IEEE Transactions on Communications*, vol. 59, no. 11, pp. 2939–2944, 2011.

APPENDIX A

APPENDIX

A.1 Fabry-Perot Etalon (FPE)

FPE is typically made of a transparent plate with two reflecting surfaces with reflection coefficients r_1, r_2, and its transmission spectrum as a function of wavelength exhibits peaks of large transmission corresponding to resonances of the etalon. Suppose the separation of two surfaces is l (which can be tuned), then the round-trip travel time is $\tau_{RT} = 2l/c$, where c is the speed of light. The transmission of FPE is a function of frequency ν:

$$T(\nu) = \frac{\left(1 - |r_1|^2\right)\left(1 - |r_2|^2\right)}{\left(1 - |r_1 r_2|\right)^2 + 4\,|r_1 r_2| \sin^2\left(\pi\nu\tau_{RT} + \phi\right)} \tag{A.1}$$

The relative transmission is periodic with period $= 1/\tau_{RT}$:

$$\frac{T(\nu)}{\max_\nu T(\nu)} = \frac{\left(1 - |r_1 r_2|\right)^2}{\left(1 - |r_1 r_2|\right)^2 + 4\,|r_1 r_2| \sin^2\left(\pi\nu\tau_{RT} + \phi\right)} \tag{A.2}$$

With respect to wavelength λ, the relative transmission is:

$$\frac{T(\lambda)}{\max_\lambda T(\lambda)} = \frac{\left(1 - |r_1 r_2|\right)^2}{\left(1 - |r_1 r_2|\right)^2 + 4\,|r_1 r_2| \sin^2\left(\pi c\tau_{RT}/\lambda + \phi\right)} \tag{A.3}$$

A.2 DD-CASSI Optimal Decision Rule

The forward model of DD-CASSI described in Section 3.3 is

$$\begin{aligned} \mathbf{y} &= A\mathbf{x}_{vec} \in \mathbb{R}^{K\times HW} \\ \text{where } A &: \mathbb{R}^{B\times HW} \to \mathbb{R}^{K\times HW} \\ A &= \begin{bmatrix} (\mathbf{1}_B^T \otimes \mathbf{I})\text{diag}([(\mathbf{c}_1^{(1)})^T, \ldots, (\mathbf{c}_B^{(1)})^T]^T) \\ \cdots \\ (\mathbf{1}_B^T \otimes \mathbf{I})\text{diag}([(\mathbf{c}_1^{(K)})^T, \ldots, (\mathbf{c}_B^{(K)})^T]^T) \end{bmatrix} \end{aligned} \tag{A.4}$$

where $\mathbf{I}$ denotes an identity operator of size $HW \times HW$, $\mathbf{c}^{(k)} \in \{0,1\}^{(H+B-1)W}$ denotes the DMD pattern for k-th snapshot, $\mathbf{x}_{vec} = [\vec{\mathbf{x}_1}^T, \ldots, \vec{\mathbf{x}_B}^T]^T \in \mathbb{R}^{B\times HW}$ denotes the flattened multispectral image cube, and vector $\vec{\mathbf{x}_b} \in \mathbb{R}^{HW}$ represents the vectorized b-th spectral band image slice.

Note that the action of A on the image tensor $x \in \mathbb{R}^{B\times H\times W}$ can be separated to independent actions on image matrices $x(\cdot,\cdot,w)$ for all $w \in [W]$, as shown in Figure 3.2. For simplicity, we consider one such image matrix $x(\cdot,\cdot,w)$ for $w = 1$ in the following calculation. We use $A_k : \mathbb{R}^{B\times H} \to \mathbb{R}^H$, which corresponds to the k-th block row in A, to denote the linear DD-CASSI operator for kth snapshot, and use $c = \mathbf{c}^{(k)} \in \{0,1\}^{H+B-1}$ to denote the corresponding binary DMD pattern in kth snapshot. Recalling (3.3), $c_i \triangleq c[i : i+H-1] \in \{0,1\}^H$ is a sliding window of size H from the DMD pattern, which encodes the image's ith spectral band. The forward operator A_k maps a matrix $x \in \mathbb{R}^{B\times H}$ to $\mathbb{R}^H$:

$$(A_k x)_i = \langle x_{:,i}, c_i \rangle = \sum_{b=0}^{B} x_{b,i} c_{i+b}, \text{ for } i = 1, \ldots, H \tag{A.5}$$

To find the adjoint operator A_k^T of A_k, for $x \in \mathbb{R}^{B\times H}, y \in \mathbb{R}^H$ we have

$$\begin{aligned} \langle A_k x, y\rangle &= \langle x, A_k^T y\rangle \\ \sum_{i=0}^{H-1} y_i (A_k x)_i = \sum_{i=0}^{H-1} y_i \sum_{b=0}^{B-1} x_{b,i} c_{i+b} &= \sum_{i=0}^{H-1}\sum_{b=0}^{B-1} x_{b,i} y_i c_{i+b} \end{aligned} \tag{A.6}$$

So $(A_k^T y)_{b,i} = y_i c_{i+b}$. Now we have

$$(A_k A_k^T y)_i = \sum_{b=0}^{B-1} c_{i+b} (A_k^T y)_{b,i} = \sum_{b=0}^{B-1} c_{i+b}^2 y_i = y_i \sum_{b=0}^{B-1} c_{i+b} \tag{A.7}$$

The last step is because the elements of c are binary. It follows that $A_k A_k^T$ is an $H \times H$ diagonal matrix,

$$A_k A_k^T = \text{diag}(\rho), \tag{A.8}$$

where $\rho_i = \sum_{b=0}^{B-1} c_{i+b}$

Now combining the all K snapshots, we can write

$$(A_k A_j^T y)_i = \sum_{b=0}^{B} \mathbf{c}_{i+b}^{(k)} (A_j^T y)_{b,i} = y_i \sum_{b=0}^{B} \mathbf{c}_{i+b}^{(k)} \mathbf{c}_{i+b}^{(j)} \tag{A.9}$$

It follows that $A_k A_j^T$ is also an $H \times H$ diagonal matrix

$$A_k A_j^T = \text{diag}(\rho^{k,j}), \rho_i^{k,j} = \sum_{b=0}^{B-1} \mathbf{c}_{i+b}^{(k)} \mathbf{c}_{i+b}^{(j)} \tag{A.10}$$

Now we have

$$AA^T = \begin{bmatrix} A_1 A_1^T & \dots & A_1 A_K^T \\ \vdots & \ddots & \vdots \\ A_K A_1^T & \dots & A_K A_K^T \end{bmatrix} = \begin{bmatrix} \text{diag}(\rho^{1,1}) & \dots & \text{diag}(\rho^{1,K}) \\ \vdots & \ddots & \vdots \\ \text{diag}(\rho^{K,1}) & \dots & \text{diag}(\rho^{K,K}) \end{bmatrix} \tag{A.11}$$

The optimal decision rule (4.19) for the DD-CASSI system involves finding the inverse of the matrix AA^T, whose size is $KHW \times KHW$. We use instead a suboptimal decision rule for DD-CASSI scheme, which is easier to implement:

$$\begin{aligned} i^* &= \arg\min_i \sum_{h=0}^{H-1} \sum_{j=0}^{W} \sum_{k=0}^{K} (y - Ax_i)_{h,j,k}^2 - 2\sigma^2 \log(\pi_i) \\ &= \arg\min_i \|y - Ax\|_2^2 - 2\sigma^2 \log(\pi_i) \end{aligned} \tag{A.12}$$

A.3 Detection in the Compressed Domain: Error Probability Upper and Lower Bounds

From (4.2), we have

$$\begin{aligned}\frac{p\left(y|H_1\right)}{p\left(y|H_0\right)} &= \exp\left(\frac{1}{2\sigma^2}\left(y^T(\sigma^2AA^T)^{-1}y-(y-Ax)^T(\sigma^2AA^T)^{-1}(y-Ax)\right)\right)\\ &= \exp\left(\frac{1}{\sigma^2}\left(y^T(\sigma^2AA^T)^{-1}Ax-\frac{1}{2}x^TA^T(\sigma^2AA^T)^{-1}Ax\right)\right)\end{aligned} \tag{A.13}$$

It follows that the likelihood ratio test (4.12) is equivalent to

$$x^TA^Ry \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\|P_{A^T}x\|^2}{2}+\sigma^2\log(\frac{\pi_0}{\pi_1}) \tag{A.14}$$

where $A^R \triangleq A^T(AA^T)^{-1}$ is the right inverse of A, and $P_{A^T} \triangleq A^RA$ is the orthonormal projection matrix onto the row space of A. Since $y|H_0 \sim \mathcal{N}(0,\sigma^2AA^T)$ and $y|H_1 \sim \mathcal{N}(Ax,\sigma^2AA^T)$, we can compute the exact error probability for a given sensing matrix A:

$$\begin{aligned}P_e &= \pi_0P(H_1 \text{ accepted} \mid H_0)+\pi_1P(H_0 \text{ accepted} \mid H_1)\\ &= \pi_0Q\left(\frac{\|P_{A^T}x\|_2}{2\sigma}+\frac{\sigma\log(\pi_0/\pi_1)}{\|P_{A^T}x\|_2}\right)\\ &+ \pi_1Q\left(\frac{\|P_{A^T}x\|_2}{2\sigma}-\frac{\sigma\log(\pi_0/\pi_1)}{\|P_{A^T}x\|_2}\right)\end{aligned} \tag{A.15}$$

where Q is the Q-function for normal distribution. When $\pi_0=\pi_1=1/2$, this expression reduces to (4.24) for $C=2$.

To analyze the error probability in the case of a random sensing matrix, we follow the approach of Davenport et al. [55] and apply Johnson-Lindenstrauss (JL) lemma (Theorem 1) to concentrate the probability.

If matrix A has a uniformly random row space of dimension m satisfying the condition in (4.7), then we have

$$\begin{aligned}\frac{\|P_{A^T}x\|_2}{2\sigma}+\frac{\sigma\log(\pi_0/\pi_1)}{\|P_{A^T}x\|_2} &\geq \frac{(1-\epsilon)\sqrt{\text{SNR}_m}}{2}+\frac{\log(\pi_0/\pi_1)}{(1+\epsilon)\sqrt{\text{SNR}_m}}\\ \frac{\|P_{A^T}x\|_2}{2\sigma}-\frac{\sigma\log(\pi_0/\pi_1)}{\|P_{A^T}x\|_2} &\geq \frac{(1-\epsilon)\sqrt{\text{SNR}_m}}{2}-\frac{\log(\pi_0/\pi_1)}{(1-\epsilon)\sqrt{\text{SNR}_m}}\end{aligned} \tag{A.16}$$

where $\mathrm{SNR}_m = \frac{m}{n}\mathrm{SNR}, \mathrm{SNR} = \frac{\|x\|_2^2}{\sigma^2}$. Using the Chernoff upper bound on Q-function,

$$Q(x) \leq \exp(-x^2/2) \tag{A.17}$$

the error probability has the upper bound stated below. Without loss of generality, let $\pi_0 \geq \pi_1$.

$$\begin{aligned} P_e \leq \pi_0 \exp\left(-\left(\frac{(1-\epsilon)^2\mathrm{SNR}_m}{8} + \frac{\log^2(\pi_0/\pi_1)}{2(1+\epsilon)^2\mathrm{SNR}_m} + \frac{(1-\epsilon)\log(\pi_0/\pi_1)}{2(1+\epsilon)}\right)\right) \\ + \pi_1 \exp\left(-\left(\frac{(1-\epsilon)^2\mathrm{SNR}_m}{8} + \frac{\log^2(\pi_0/\pi_1)}{2(1-\epsilon)^2\mathrm{SNR}_m} - \frac{\log(\pi_0/\pi_1)}{2}\right)\right) \end{aligned} \tag{A.18}$$

To derive a corresponding lower bound, we use the following result by Chang et al. [69], which provides a Chernoff-type lower bound for erfc function.

Theorem 3. *The function* $f(x) = \alpha\exp(\beta x)$ *is a lower bound of* $\mathrm{erfc}(x)$ *if*

$$\beta > 1 \quad \text{and} \quad 0 < \alpha \leq \sqrt{\frac{2e}{\pi}}\frac{\sqrt{\beta-1}}{\beta} \tag{A.19}$$

Recalling that the Gaussian Q-function is related to the erfc function by

$$Q(x) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right), \quad x \geq 0 \tag{A.20}$$

we choose $\alpha = 0.5, \beta = 1$ and obtain the Chernoff-type lower bound for the Q-function:

$$Q(x) \geq \exp(-x^2) \tag{A.21}$$

The error probability hence has the lower bound:

$$\begin{aligned} P_e \geq \pi_0 \exp\left(-\left(\frac{(1+\epsilon)^2\mathrm{SNR}_m}{4} + \frac{\log^2(\pi_0/\pi_1)}{(1-\epsilon)^2\mathrm{SNR}_m} + \frac{(1+\epsilon)\log(\pi_0/\pi_1)}{(1-\epsilon)}\right)\right) \\ + \pi_1 \exp\left(-\left(\frac{(1+\epsilon)^2\mathrm{SNR}_m}{4} + \frac{\log^2(\pi_0/\pi_1)}{(1+\epsilon)^2\mathrm{SNR}_m} - \log(\pi_0/\pi_1)\right)\right) \end{aligned} \tag{A.22}$$

For $\pi_0 = \pi_1 = 0.5$, the upper and lower bounds simplify to

$$\exp\left(-\frac{(1+\epsilon)^2\mathrm{SNR}_m}{4}\right) \leq P_e \leq \exp\left(-\frac{(1-\epsilon)^2\mathrm{SNR}_m}{8}\right) \tag{A.23}$$

A.4 Classification in the Compressed Domain: Error Probability Upper Bound

We derive an upper bound of error probability. Denoting the true underlying hypothesis by H_t, we have $y \sim \mathcal{N}(Ax_t, \sigma^2 AA^T)$ and

$$\begin{aligned}
&P(\text{Error occurs } \mid H_t) \\
&= P(t \neq \arg\min_m (y - Ax_m)^T (AA^T)^{-1} (y - Ax_m) - 2\sigma^2 \log(\pi_m) \mid H_t) \\
&= P\Big(\exists \ell \neq t, (y - Ax_\ell)^T (AA^T)^{-1} (y - Ax_\ell) - 2\sigma^2 \log(\pi_\ell) \\
&\qquad \leq (y - Ax_t)^T (AA^T)^{-1} (y - Ax_t) - 2\sigma^2 \log(\pi_t) \mid H_t\Big) \\
&\leq \sum_{\ell \neq t} P\Big(2y^T (AA^T)^{-1} A(x_t - x_\ell) \\
&\qquad \leq x_t P_{A^T} x_t - x_\ell P_{A^T} x_\ell + 2\sigma^2 \log\left(\frac{\pi_\ell}{\pi_t}\right) \mid H_t\Big) \\
&= \sum_{\ell \neq t} Q\left(\frac{-\|P_{A^T}(x_t - x_\ell)\|_2^2 + 2\sigma^2 \log\left(\frac{\pi_t}{\pi_\ell}\right)}{2\sigma \|P_{A^T}(x_t - x_\ell)\|_2}\right) \\
&= \sum_{\ell \neq t} Q\left(-\frac{\|P_{A^T}(x_t - x_\ell)\|_2}{2\sigma} + \frac{\sigma \log\left(\frac{\pi_t}{\pi_\ell}\right)}{\|P_{A^T}(x_t - x_\ell)\|_2}\right)
\end{aligned} \tag{A.24}$$

because $y^T (AA^T)^{-1} A(x_t - x_\ell) \sim \mathcal{N}(x_t^T P_{A^T}(x_t - x_\ell), \sigma^2 \|P_{A^T}(x_t - x_\ell)\|_2^2)$ under hypothesis H_t. The total error probability can then be upper-bounded:

$$P_e \leq \sum_{t=1}^{C} \pi_t \sum_{\ell \neq t} Q\left(-\frac{\|P_{A^T}(x_t - x_\ell)\|_2}{2\sigma} + \frac{\sigma \log\left(\frac{\pi_t}{\pi_\ell}\right)}{\|P_{A^T}(x_t - x_\ell)\|_2}\right) \tag{A.25}$$

Under uniform priors, $\pi_i = \frac{1}{C}, \forall i$, the error probability upper bound reduces to

$$P_e \leq \frac{1}{C} \sum_{t=1}^{C} \sum_{\ell \neq t} Q\left(-\frac{\|P_{A^T}(x_t - x_\ell)\|_2}{2\sigma}\right) \tag{A.26}$$