Detection of Node Pore Sensing Signals

Maxwell Lin-He

May 14, 2021

Detection of Node Pore Sensing Signals

by

Maxwell Lin-He

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael Lustig, Chair
Professor Laura Waller

Spring 2021

Detection of Node Pore Sensing Signals

by Maxwell Lin-He

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Michael Lustig
Research Advisor

05/14/2021
(Date)

* * * * * * *

Laura Waller
Professor Laura Waller
Second Reader

5/14/21
(Date)

Detection of Node Pore Sensing Signals

Abstract

Detection of Node Pore Sensing Signals

by

Maxwell Lin-He

Master of Science in Electrical Engineering and Computer Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Michael Lustig, Chair

We present methods for detecting cell events in simulated node pore sensing signals. First, we propose an inverse formulation for detection and methods for solving it. Using our complex simulation model, we also show that a data-driven deep learning approach is able to effectively learn to identify cell events and apply them on unseen examples, even with noise and model mismatch.

Contents

Contents i

List of Figures ii

List of Tables iii

1 Introduction 1
1.1 Node Pore Sensing 1
1.2 Algorithms 3

2 Methods 4
2.1 System Model 4
2.2 Data Generation 6
2.3 Inverse Problem Model 7
2.4 Deep Neural Networks 9

3 Results 12
3.1 Signal Analysis 12
3.2 Detection Accuracy 13

4 Discussion 16

5 Conclusion 18

Bibliography 19

List of Figures

1.1 Example ambiguity function for a single cell event unit scale and zero delay, where the largest value is located 2

2.1 Forward model. (a) plot of x with three cell events. Each cell is represented as an impulse located at a certain scale and delay. The height of the impulse is the amplitude of the cell's impedance response, corresponding to the cell's size. (b) dictionary of signals $\{h_i\}_{i=1}^{k}$, the system response over a range of scales. (c) the ideal impedance signal, r, resulting from x and (d) measured signal b with noise. 6

2.2 Output of the alternating direction method of multipliers (ADMM) on a noisy signal with 2 cell events. The input signal is given by (a). The true underlying x is shown in (b). ADMM was run for 500 iterations with regularization penalty $\tau = 0.1, 1, 4$, shown in (c), (d), (e) respectively. As the regularization penalty is increased, the reconstructed solution becomes more sparse, decreasing the effects of noise. However, as the penalty increases, ADMM also becomes more sensitive to dynamic range and fails to detect smaller cells. 10

2.3 Input and output of fully convolutional network on an test example with three cell events. The three peaks for each cell are circled in red. The result of filtering an impedance measurement is shown in (a). The side lobe patterns mask where the true peaks corresponding each cell are located. The ground truth label is shown in (b) and the output of a neural network is shown in (c), where the true peaks are isolated from (a). 11

3.1 Precision-Recall curves of UNet and ADMM. Plot shows precision and recall values on the default dataset with the detection floor ranging from 1 to 1/50 during peak finding. 14

3.2 Analysis of true positives and false negatives coincidence events. The signal-to-interference ratio (SIR) and signal-to-noise ratio (SNR) of each cell event are plotted as a scatter plot. (a) The dataset was simulated with default simulator settings. (b) The dataset contains no velocity variations, modeling cells traveling all at a constant velocity. (c) The dataset models a homogeneous cell size sample of 12 μm. All ideal impedance responses have identical amplitude. 15

List of Tables

3.1 Precision and recall on different datasets. In the peak finding algorithm used to locate cell events, UNet (1) limits dynamic range of peaks to 20. ADMM and UNet (2) limits dynamic range to 5. 13

Acknowledgments

The work is a collaboration with Alan Dong and advised by Professor Michael Lustig. I would like to thank Alan Dong for all guidance and advice on this project. I would also like to thank Professors Michael Lustig and Laura Waller for reading my work and this thesis. Finally, I would like to thank all my professors, graduate student instructors, and fellow students for all I have learned and for intellectual inspiration.

Chapter 1

Introduction

1.1 Node Pore Sensing

Microfluidic cytometry is a technique for measuring characteristics of cells, such as cell size and cell count, by suspending them in fluid and passing them through a device. One method of detecting cells passing through a cytometer is to measure the impedance effects of a cell as it passes through a channel. One of the earliest such devices is the Coulter counter, in which cells suspended in an electrolyte solution travel through a channel with an electric potential applied across it [8]. While the cell is in the channel, the measured electrical impedance is increased, thereby indicating the presence of a cell. The increase in impedance of a cell can be calculated as a function of the cell and channel diameters. When detecting cells of different sizes, channel width becomes a concern, as a channel that is too large leads to poor signal-to-noise ratio (SNR) while a channel that is too small leads to clogging of the device. One solution to this problem is Node Pore Sensing (NPS) [4], in which the channel has narrow sections called pores and wider sections called nodes. The narrower pores are akin to the channel in the Coulter counter, as impedance is increased while the cell passes through. However, when the cell is in the wider nodes, impedance returns to near baseline. A single cell produces a signal determined by the sequence of pores and nodes and their lengths.

NPS alleviates issues in detection of a heterogeneous sample in cell size, but challenges in detecting coincidence events still remain, where multiple cells enter the device channel at the same time. In coincidence events the total impedance measurement is a combination of impedance effects from all cells, making it difficult to isolate the contributions of each individual cell.

Previous work uses a channel geometry that modulates the device's impedance response to produce a Manchester encoded Barker binary-code sequence [15]. Barker sequences are known for their favorable auto-correlation properties, with extremely low side lobes. This makes a Barker-coded NPS signal well-suited for correlation-based cell detection methods. If a cell's velocity through the channel is known, then cross-correlation of the total impedance signal with the same-length Barker sequence will effectively detect the cell's impedance signal,

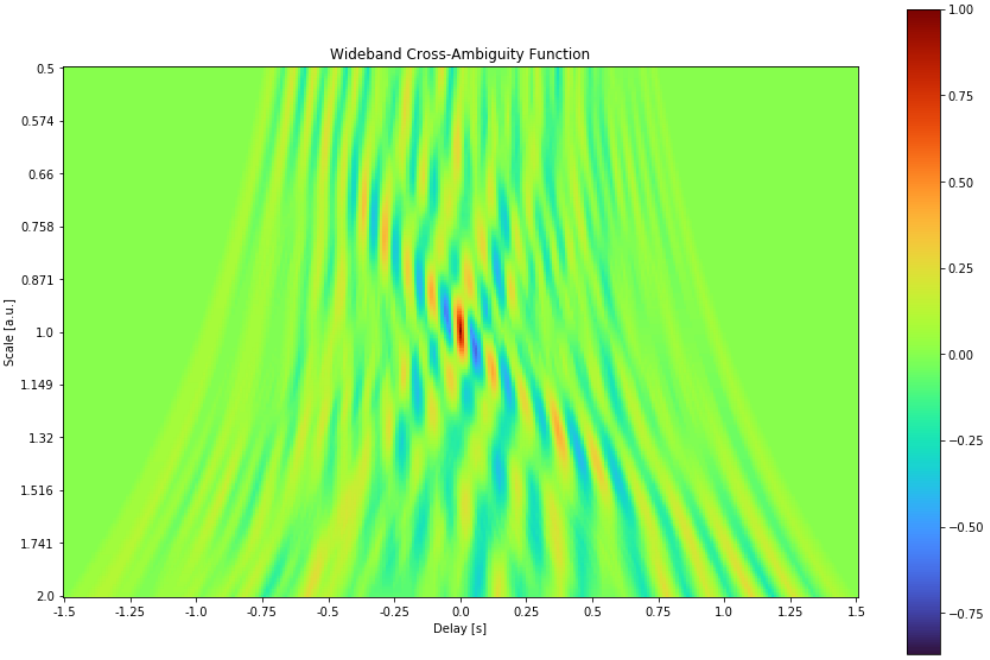


Figure 1.1: Example ambiguity function for a single cell event unit scale and zero delay, where the largest value is located

even in the presence of noise and interfering signals from other coincident cells. However, since cells travel through the channel at different speeds, we must apply a filter bank of the Barker sequence over a range of scales. The output of the filter bank is an ambiguity function shown in Figure 1.1

Although the Barker code is designed to have low side lobes in its autocorrelation, correlations between pairs of them stretched and compressed to different lengths is not insignificant, and coincidence events still remain a challenge, particularly when dynamic range is high. When a small cell and large cell travel through the channel at the same time, the peaks from the contribution of the small cell become obscured by the side lobe patterns from the large cell. Additionally, due to fluid dynamics, cells traveling off center in the channel will deviate to the sides of the nodes, causing time delays that are hard to predict. Because cells do not transit the channel at a constant velocity, the impedance signal becomes distorted in time and leads to model mismatch with the Barker code signal.

Previous work in [10], uses a greedy successive interference cancellation method. This method iteratively uses correlation filtering and selects the largest correlation as a match for a cell. It then subtracts out the model Barker code signal for that cell from the measurement to perform correlation again. When there is model mismatch, the subtracted Barker code does not match perfectly with the true impedance signal, and the result contains large residual spikes. These spikes result in large correlation effects in subsequent iterations. Especially when the signal has high dynamic range, the residual spikes can have correlations larger that the actual cell response, and detection can fail.

1.2 Algorithms

Fully convolutional neural networks, deep neural networks containing only convolutional layers and no fully connected layers, trained end-to-end have shown to be powerful in image segmentation tasks [11]. Fully convolutional networks also offer the advantage of taking in arbitrary size inputs and producing a correspondingly-sized output. A commonly used deep neural network (DNN) architecture for this task is the UNet [16], which was originally developed for biomedical image segmentation. We apply deep learning to the NPS detection problem by first cross-correlating the impedance signal with a filter bank of signal templates and then treating cell detection as an image semantic segmentation task, where we aim to classify if a cell exists at a particular location in the filter bank output. We show that a UNet is be able to detect peaks at true cell events within a noisy ambiguity function containing many side lobes. By extending the normal interpretation of a binary output from a neural network, the UNet is also able to recover cell size information.

Much recent work in biomedical image reconstruction and signal processing has centered around unrolled iterative optimization algorithms with deep networks [13]. Unrolled deep methods use learned networks to denoise or regularize images, while incorporating data consistency steps to conform to an image model [1, 22, 14]. We also propose a signal recovery model with natural connections to correlation filtering. Using a convex formulation, this signal recovery problem can be efficiently solved with the alternating direction method of multipliers (ADMM) [6]. We compare results to deep segmentation methods, and analyze performance on coincidence events in the presence of high dynamic range and model mismatch, as well as the effects of noise and interference in resolving coincidence events.

Chapter 2

Methods

2.1 System Model

As a cell travels through the NPS channel, the impedance (magnitude) across the channel, $r(t)$, changes over time. Under ideal conditions for the physics model of the NPS device, a cell traveling with a constant velocity will produce a signal with shape determined by the physical sequence of nodes and pores. For a single cell passing the midpoint of the channel at time t_0, the ideal impedance signal is

$$r(t) = c \cdot h\left(\frac{t}{\tau_0} - t_0\right) \tag{2.1}$$

where h is a Manchester-Barker length-13 sequence, c is a function of cell size, and τ_0 is a scale parameter equal to the transit time of the cell. Impedance measurement of the NPS device captures the size, timing, and velocity information of cells passing through it. We can model cell transit events as $x(t, \tau)$, where a nonzero value of x represents the impedance response amplitude of a cell passing the midpoint of the channel at time t with a transit time of τ. Zero values represent no cell events.

Let $r_{(m)}$ be the impedance response of the m-th cell, q be baseline drift, and z be measurement noise. We model the discrete measured signal b as

$$b[n] = r(nT) + q(nT) + z(nT) \tag{2.2}$$

$$r(t) = \sum_{m=1}^{N} r_{(m)}(t) \tag{2.3}$$

where n is the time index and T is the sampling period.

From here, we use subscripts as an index into scales. Let $\{h_i\}_{i=1}^{k}$ be a collection of k scales of the unit channel response, such that $h_i[n] = h(nT/s_i)$, where $h(t)$ is the prototypical channel response with some empirically-determined typical transit time (in seconds). The scale parameters are chosen such that their logarithms are evenly spaced, so that the relative scales s_i/s_j are mapped one-to-one to the difference in index $i - j$. Based on the duration of the prototype signal $h(t)$, we define the scale parameter as $s = \frac{\tau}{\text{duration of h}}$.

Let H be a discrete, linear forward model where $r = H(x)$. We define

$$H(x) = \sum_{i=1}^{k} x_i[n] * h_i[-n] \tag{2.4}$$

as a sum of convolutions, where $x_i[n] = x(nT, s_i)$. In this model, x is a function over time and scale, where $x_i[n]$ represents the amplitude of a cell's impedance response for a cell passing the midpoint of the channel at time nT with a transit time of τ_i. A value of zero indicates no cell event. Note that x is always non-negative.

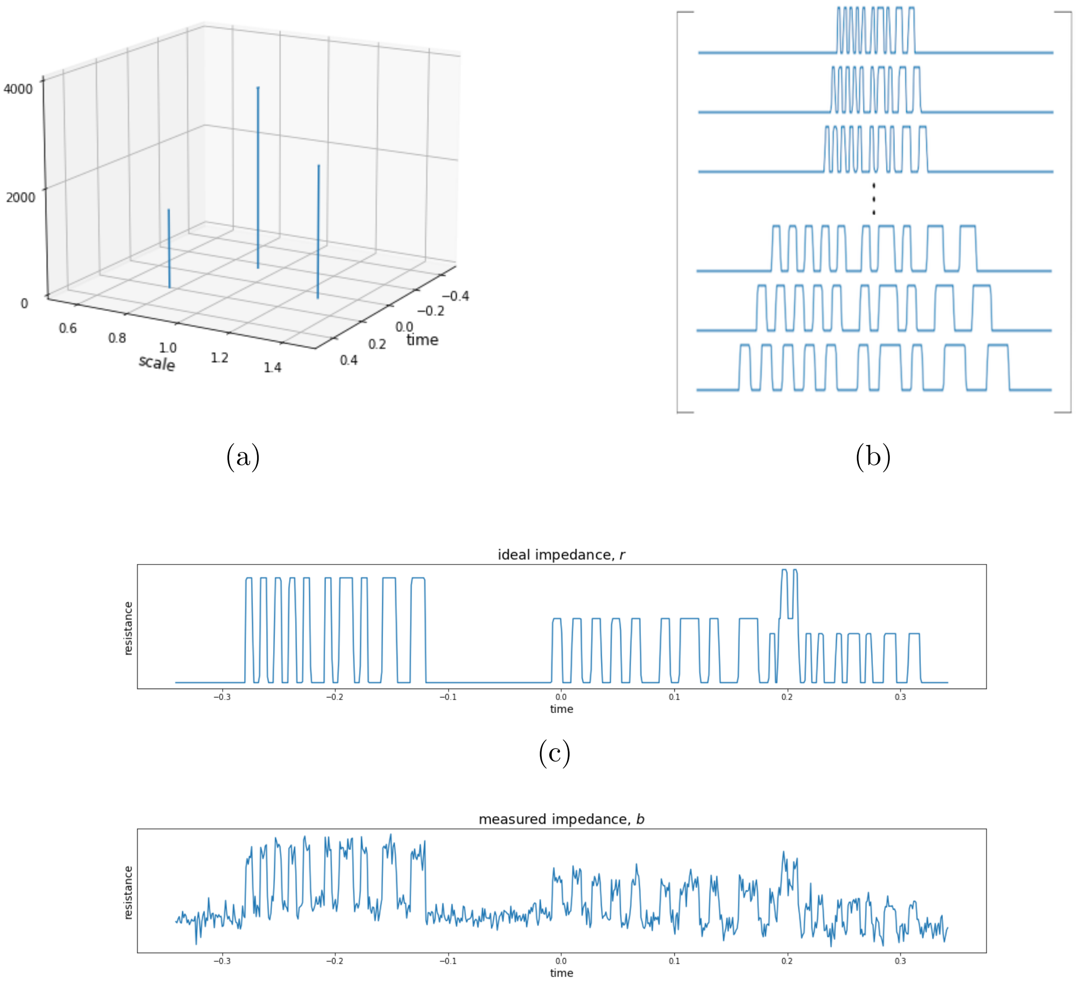
In our implementation, we fix the measured signal to be an odd positive integer d and treat all signals as centered at time 0. Thus, processing signals of greater length requires analyzing segments of length d at a time. Hence $b[n]$ has support on $n = \lfloor \frac{d}{2} \rfloor, \ldots, \lfloor \frac{d}{2} \rfloor$. Similarly, we consider a range of k feasible scales for h_i's and fix the number of delays to length ℓ. Then in in vector form, we have $b \in \mathbb{R}^d$ and $H : \mathbb{R}^{k \times \ell} \to \mathbb{R}^d$. For the filtering operation we introduce next, it is convenient to choose $\ell = \lfloor 2d \rfloor + 1$ and window h_i all to length d.

In analysis of a signal b, we often find it convenient to first apply a filtering transformation $H^\top$. The goal of the filtering operation is to aid in locating the impedance response, r, of a cell passing through the channel from the measurement b. The filtering is defined as $\hat{x} = H^\top(b)$, where

$$\hat{x}_i[n] = b[n] * h_i[-n] \qquad \text{for } i = 1, \ldots, k \tag{2.5}$$

This definition of $H^\top$ also coincides with the adjoint of H.

From $\hat{x}$, we can detect the timing and velocity of a cell traveling through the channel, as well as its size. Suppose a cell travels through the channel with velocity v_i, and and passes the midpoint of the channel at time m. If b only contains signal from this one cell, then $\hat{x}$ has a maximal peak at $\hat{x}_i[m]$. This follows from the design of the channel response and the fact that we center all signals at time 0. Furthermore, the value of $\hat{x}_i[m]$, is proportional to the amplitude of b, and thus the size of the cell. Hence, we can recover the timing and size of the cell simply by locating the peak of $H^\top(b)$. Such a filtering scheme is motivated by wideband cross ambiguity functions in sonar and radar processing [19], and $\hat{x}$ is essentially a discrete sampled ambiguity function. An alternative interpretation is that $\hat{x}$ is a sampling of the continuous wavelet transform of $b(t)$ with mother wavelet $h(t)$. The same collection $\{h_i\}_{i=1}^k$ from the forward model can be used; however, when using $H^\top$ as a matched filter as done in Section 2.4, we instead use, $\bar{h}_i = \sqrt{s_i} h_i$. This matches the formulation for wideband cross ambiguity function and the continuous wavelet transform. Furthermore a bipolar encoding of h can be used in order to reduce effects of DC offset from measurements after correlation.



(a)

(b)

(c)

(d)

Figure 2.1: Forward model. (a) plot of x with three cell events. Each cell is represented as an impulse located at a certain scale and delay. The height of the impulse is the amplitude of the cell's impedance response, corresponding to the cell's size. (b) dictionary of signals $\{h_i\}_{i=1}^{k}$, the system response over a range of scales. (c) the ideal impedance signal, r, resulting from x and (d) measured signal b with noise.

2.2 Data Generation

We aim to analyze different algorithms for determining cell events in a NPS sensing signal. In order to test algorithmic performance on a variety of signals, we require a large dataset of signals, such as examples with different types of coincidence events. Furthermore, algorithms employing deep neural networks will require an even larger amount of training data. In order to train neural networks on NPS data, we first construct a simulator that outputs signals constructed with statistics from historical device data. Our simulator randomly generates a parameters for a cell event for size, speed, and timing, which is reflected in the amplitude, scale, and delay of the measured signal respectively. We model the number of cells with a Poisson process and linearly combine the signals for each cell. Transit times and velocity variations are randomized using the empirical covariance of transit times for each node and pore in the device from previous experimental data. Cell diameters are drawn from a log-normal distribution to estimate the size of epithelial cells. Baseline drift is modeled as additive brown noise, and sensor noise is modeled with additive white Gaussian noise. The impedance values for both cell events and noise are calculated using the physics model given the device geometry.

Experiments done with simulated data allow us to easily and quickly vary parameters and give access to true underlying statistics such as SNR. This allows us to analyze our methods on both ideal signals as well as a multitude of noise models.

2.3 Inverse Problem Model

We can model recovering cell events from a measurement as

$$\arg\min_{x} \ \frac{1}{2}\|Hx-b\|_2^2+\lambda g(x) \tag{2.6}$$

where $x \in \mathbb{R}^{k\times\ell}$ is the reconstructed signal of cell events, $b \in \mathbb{R}^d$ is the received impedance measurement, $H : \mathbb{R}^{k\times\ell} \to \mathbb{R}^d$ is the the forward model, and $g : \mathbb{R}^{k\times\ell} \to \mathbb{R}$ is some regularization function. The number of cells present in x is bounded statistically by the length of the channel as well as the concentration of cells. Modeling cell arrivals as a Poisson process, the number of cell events in a given signal obeys a Poisson distribution, whose tail probability decays exponentially [9, 18]. Hence, we can obtain a high probability bound on the ℓ_0-“norm” on x, $\|x\|_0 \leq N$. The mean number of cells in passing through the channel in a second is much less than the sampling frequency, $1/T$, so $N \ll k\ell$, and we expect x to have extreme native sparsity.

ADMM

For our problem, solving the objective with traditional gradient based methods is problematic due to the unique structure of our signal model. The Hessian of the data consistency term in the objective function is $\frac{\partial^2}{\partial x^2}\frac{1}{2}\|Hx-b\|_2^2 = H^\top H$. This matrix is poorly conditioned and gradient methods fail to converge or even provide a reasonable solution. When explicitly constructing H as matrix of size matching our experiments, the condition number, κ, cannot be computed numerically in a reasonable amount of time. Even a reduced size problem where the number of scales and sampling rate is reduced by 90%, we still have $\kappa > 10^{10}$, with a maximum eigenvalue greater than 10^3. With such a large maximum eigenvalue, improvements in conditioning via regularization is limited unless an absurdly large penalty weight is used. When the Hessian of a quadratic function has a large condition number, a basic gradient descent algorithm with line search for step sizes has an iteration complexity that scales approximately linearly with κ [7]. Other methods such as the conjugate gradient algorithm still have an iteration complexity scaling with the square root of κ [17]. Therefore, we cannot expect a gradient method to converge for our problem in practice. Due to the poor performance of gradient methods, we turn to the alternating direction method of multipliers. ADMM is a primal-dual method for solving convex optimization problems of the form

$$\begin{aligned} \min_{x,z} \; & f(x) + g(z) \\ \text{such that } & Ax + Bz = c \end{aligned} \tag{2.7}$$

For our problem, we let f represent the data consistency function and g be a regularization function. We use the ℓ_1-norm $\|x\|_1 = \sum_{i=1}^{k} \sum_{j=1}^{\ell} |x_{ij}|$ as a convex relaxation of the ℓ_0 constraint to regularize for sparsity. We also add a non-negativity constraint as a cell event cannot cause a negative channel response. Finally, when the forward model is a sum of 1-D convolutions, we can use the splitting $H = DM$ where D is a slice or cropping operator and M is a 2-D convolution [3].

The optimization we wish to solve is

$$\begin{aligned} \arg\min_{x,u,v,w} \; & \|Dv - b\|_2^2 + \lambda\|u\|_1 + \mathbb{1}_+(w) \\ \text{such that } Mx &= v \\ x &= u \\ x &= w \end{aligned} \tag{2.8}$$

where

$$\mathbb{1}_+(x) = \begin{cases} 0 & \text{if all elements of } x \text{ are non-negative} \\ +\infty & \text{otherwise} \end{cases}$$

is a indicator function for non-negativity.

Then the ADMM updates, with μ_1, μ_2, μ_3 as penalty parameters for the augmented Lagrangian, are given by

$$\begin{aligned}
u^{k+1} &= \mathcal{T}_{\lambda/\mu_2}(x^k + \eta^k/\mu_2) \\
v^{k+1} &= \left(D^\top D + \mu_1 I\right)^{-1}(\xi^k + \mu_1 M x^k + D^\top b) \\
w^{k+1} &= \max(\rho^k/\mu^3 + x^k, 0) \\
r^k &= \mu_3 w^{k+1} - \rho^k + \mu_2 u^{k+1} - \eta^k + M^\top(\mu_1 v^{k+1} - \xi^k) \\
x^{k+1} &= (\mu_1 M^\top M + \mu_2 I + \mu_3 I)^{-1} r^k \\
\xi^{k+1} &= \xi^k + \mu_1(M x^{k+1} - v^{k+1}) \\
\eta^{k+1} &= \eta^k + \mu_2(x^{k+1} - u^{k+1}) \\
\rho^{k+1} &= \rho^k + \mu_3(x^{k+1} - w^{k+1})
\end{aligned}$$

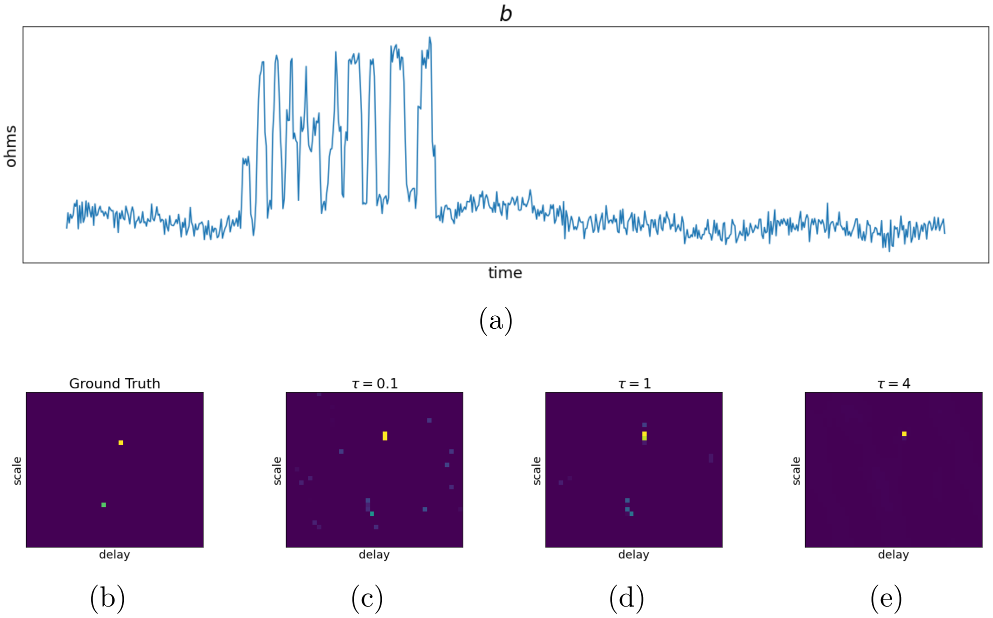
where $\mathcal{T}_\alpha$ is the soft thresholding function with threshold value α, arising from the proximal operator of the ℓ_1-norm [2, 20]. In the ADMM updates, u, v, w are auxiliary primal variables and ξ, η, ρ are their corresponding dual variables.

We employ the penalty parameter tuning strategy in [6], where in each iteration, μ_i's are updated to balance the primal and dual residuals. For our problem, every update can be computed efficiently. The operation M is a convolution, which can implemented using Fast Fourier Transforms (FFTs). All other operations are have linear time complexity in the size of the inputs.

2.4 Deep Neural Networks

The simulator allows us to generate an arbitrary number of example signals, along with ground truth labels, to train deep neural networks. We use a standard UNet, with a fully convolutional architecture. Instead of supplying the network directly with the measurement b, it is useful to first perform the filtering operation $H^\top$, which imparts additional structure of the known system system response on the data. The 2D "image" $H^\top(b)$ contains peaks at nonzero points in x, but also contains many non-negligible side-lobes resulting from high correlations of h at different scales. As a result, side lobe patterns from multiple cells create ambiguity for the location of the true peaks. While the side lobe patterns from $H^\top$ cannot be modeled as a 2D convolution with x as our signal model is not shift invariant in the scale axis, the pattern does maintain the same shape for cells of different velocities, so we expect a CNN to be able to detect them. Instead of training the model with impulse labels, we instead apply a Gaussian blur to the impulse. The network is able learn to output peaks more quickly under this scheme.

Figure 2.3 shows an example of a UNet finding the true peaks from the correlation filtering of an impedance signal. The UNet is clearly able to detect the two most salient peaks on the left in the ground truth label, even when the side lobe of the tallest peak covers the smaller peak. We hypothesize that a convolutional neural network with a sufficiently large receptive field [12], will be able learn the sidelobe patterns and suppress them, leaving only

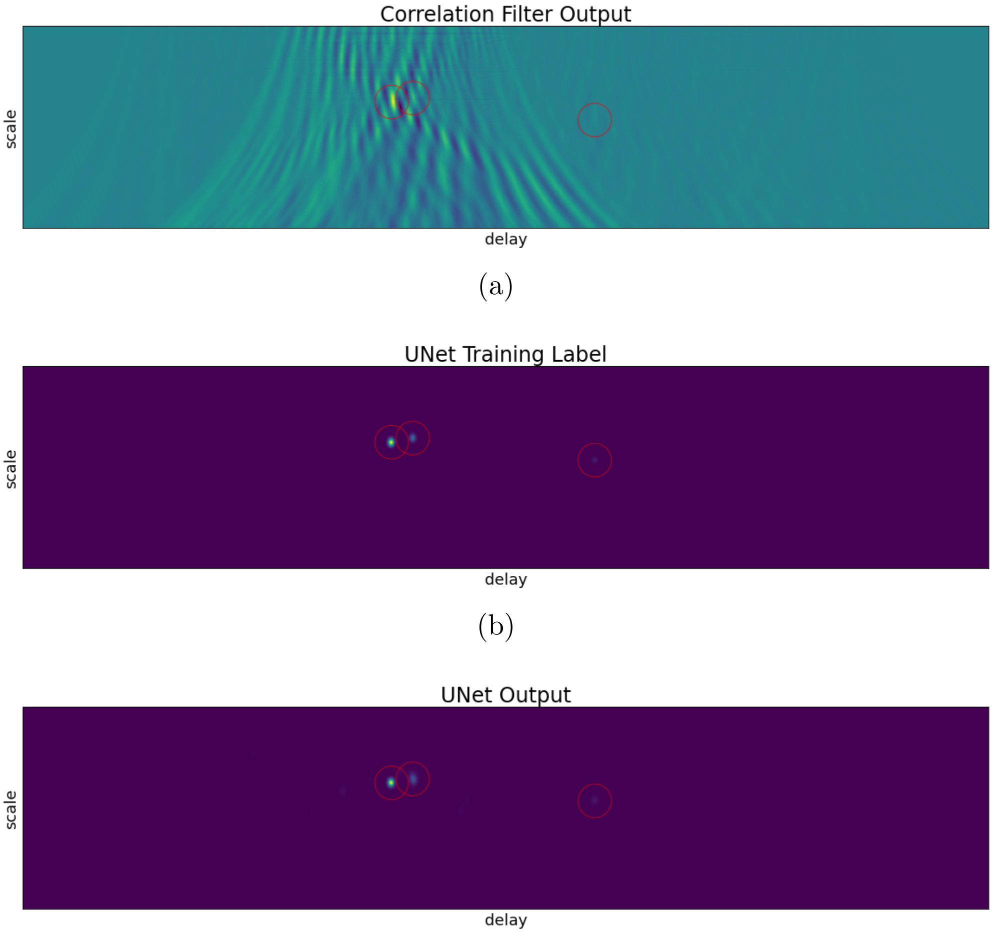


(a)

(b) (c) (d) (e)

Figure 2.2: Output of the alternating direction method of multipliers (ADMM) on a noisy signal with 2 cell events. The input signal is given by (a). The true underlying x is shown in (b). ADMM was run for 500 iterations with regularization penalty $\tau = 0.1, 1, 4$, shown in (c), (d), (e) respectively. As the regularization penalty is increased, the reconstructed solution becomes more sparse, decreasing the effects of noise. However, as the penalty increases, ADMM also becomes more sensitive to dynamic range and fails to detect smaller cells.

the true peaks. This problem can be framed as an image segmentation task on $\hat{x}$, where we would like to identify samples in $\hat{x}$ that belong to the true peak. However, instead of using the UNet as a binary classifier by applying a softmax operation to the last layer, we use the UNet as a regression function. A rectified linear unit (ReLU) is applied to the output to enforce non-negativity of x, and the network is trained using mean squared error loss to retain amplitude information. Figure 2.3 shows an example of a UNet finding the true peaks from the correlation filtering of an impedance signal.



(a)

(b)

(c)

Figure 2.3: Input and output of fully convolutional network on an test example with three cell events. The three peaks for each cell are circled in red. The result of filtering an impedance measurement is shown in (a). The side lobe patterns mask where the true peaks corresponding each cell are located. The ground truth label is shown in (b) and the output of a neural network is shown in (c), where the true peaks are isolated from (a).

Chapter 3

Results

3.1 Signal Analysis

For each method presented, we analyze the performance on cell detection in different regimes. We compare performance of each model processing signals with and without velocity variations and noise. We study the robustness of our algorithms to coincidence events by analyzing performance in cases of high dynamic range (large cell size variation) and high overlap (many cells close to each other). Additionally, as our trained deep neural networks are designed to locate true peaks in the presence of side lobes, we consider the signal-to-interference ratio (SIR) and signal-to-noise ratio of the filtered signal $H^\top(b)$.

From equation (2.2), we define the dynamic range of a signal as the ratio of the amplitude of the largest and smallest $r_{(m)}$'s. For SIR, we work in the correlation domain and compute a ratio for each $r_{(m)}$ as the ratio between $H^\top(r_{(m)})$ and $\sum_{m' \neq m} H^\top(r_{(m')})$ at the true peak position of cell i. Similarly, we use the noise value in the correlation domain in calculating SIR and SNR. This measure of SIR attempts to quantify the amount that a cell's correlation pattern is hidden by the side lobe patterns of all other cells.

Both the ADMM and UNet detection algorithms output a 2D array, x_{out}, indicating cell events at at specified scales and delays. In our simulations, we take the ground truth x, and for each cell event in x, we map the closest peak from x_{out}. Peaks are found by iteratively finding the largest value in the array above a specified detection floor and removing a small ball centered at that point. We declare successful detection if the coordinates for the peak in x_{out} are within a Manhattan distance of 10 from the true peak in x. We allow this small margin of error, because noise and velocity variations in the measured signal can cause it to have higher correlations with filters at nearby scales and delays. Successes are counted as true positives. If a peak from x does not have a matching peak from x_{out}, then we consider the cell event as a false negative. Finally any remaining peak from x_{out} is considered a false positive.

For this problem there is no natural concept of a true negative (unless we consider an infinite number of true negatives at every possible scale and delay). Therefore, in analyzing

		TP	FN	FP	precision	recall
ADMM	normal dataset	1031	1101	510	66.9%	43.4%
	no velocity variation	1101	870	298	78.7%	55.9%
	no cell size variation	1507	532	331	82.0%	73.9%
UNet (1)	normal dataset	1900	232	319	85.6%	89.1%
	no velocity variation	1803	168	256	87.6%	91.5%
	no cell size variation	2011	28	371	84.4%	98.6%
UNet (2)	normal dataset	1662	470	42	97.5%	78.0%
	no velocity variation	1576	395	26	98.4%	80.0%
	no cell size variation	2005	34	33	98.4%	98.3%

Table 3.1: Precision and recall on different datasets. In the peak finding algorithm used to locate cell events, UNet (1) limits dynamic range of peaks to 20. ADMM and UNet (2) limits dynamic range to 5.

the overall accuracy of our algorithms on a dataset, we consider the precision (positive predictive value) and recall (true positive rate), which are defined as

$$\text{precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}}$$
$$\text{recall} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

In our experiments, we simulate signals of length 683 ms, with a sampling rate of 1000 Hz. The collection of channel response templates $\{h_i\}_{i=1}^{201}$ is composed of 201 scales with transit times ranging from 83.3 ms to 333.3 ms. Cell sizes are simulated from a log-normal distribution with a mean of 12 μm and a standard deviation of 3 μm. The number of cells in each signal is drawn from a Poisson distribution with mean 2.

3.2 Detection Accuracy

We test our methods on datasets containing 1000 signals each. We create three different datasets: The first dataset contains cell events with all the parameters described in Section 2.2. The second dataset removes velocity variation in the signals, modeling all cells as traveling with a constant velocity. The third dataset models a homogeneous sample of cells, with all cells simulated with a diameter of 12 μm. The second dataset gives us a baseline for how the methods perform when there is no model mismatch and allows us to compare how much performance degrades when compared to the first dataset when the measured signals do not perfectly match our channel response templates h_i. The third dataset instead gives a baseline for model performance when dynamic range is constant, and how it affects detection.

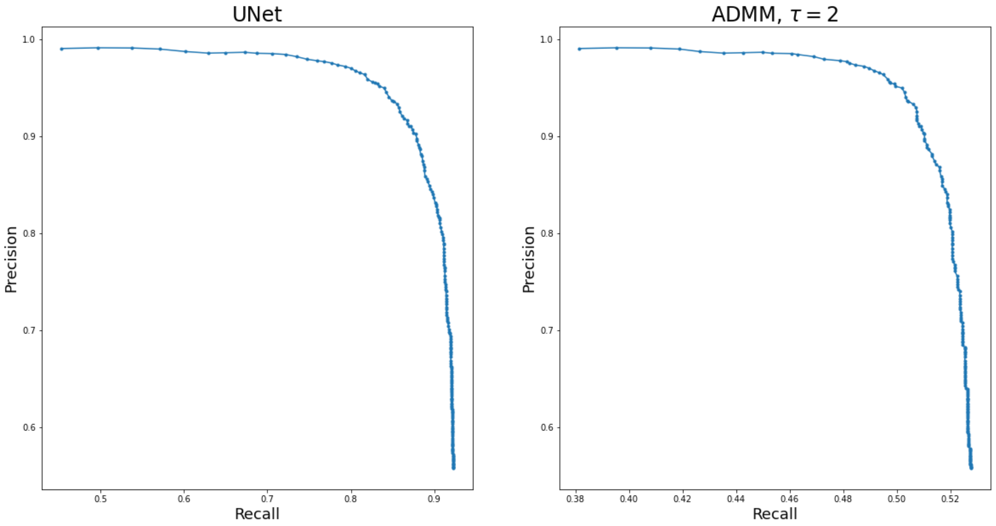


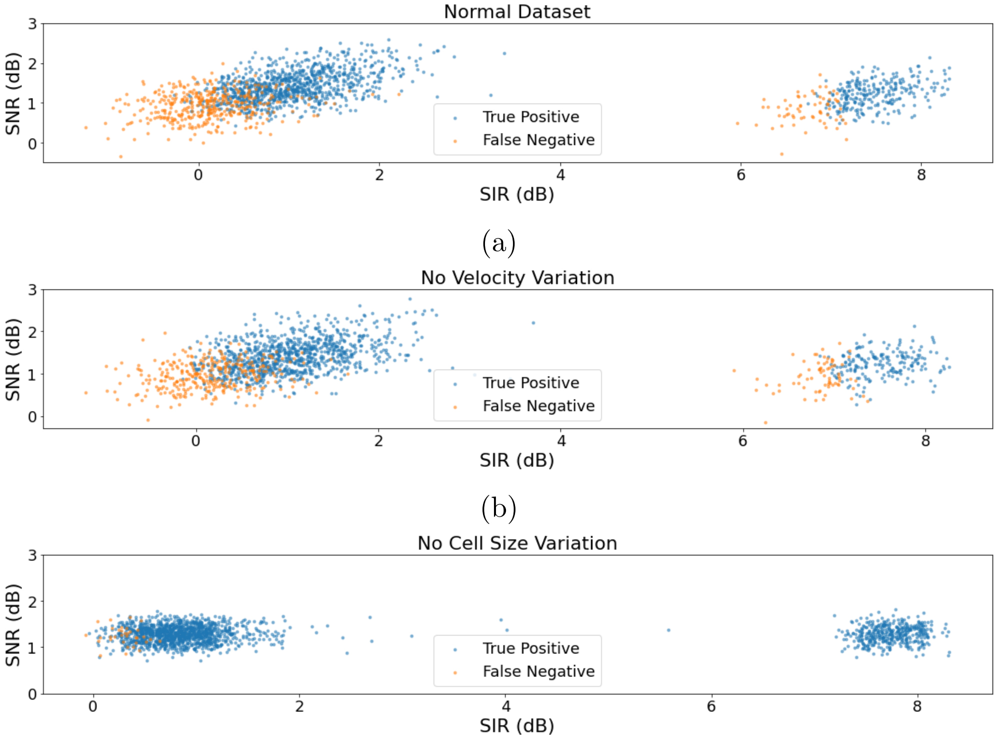
Figure 3.1: Precision-Recall curves of UNet and ADMM. Plot shows precision and recall values on the default dataset with the detection floor ranging from 1 to 1/50 during peak finding.

ADMM is run on inputs for 500 iterations and the regularization parameter set to $\tau = 2$. We run the peak finding algorithm with a floor of 1/5 of the maximum peak, which empirically gives a reasonable balance between precision and recall.

We run a UNet on unseen examples from our test datasets on the same model. The model used is trained on a simulated dataset containing 50000 examples. We run on all the datasets twice with a different floor chosen for the peak finding algorithm. Since the UNet has better dynamic range performance, we try two different peak detection floors. In run 1, we set the floor to 1/20 of the maximum peak, since a dynamic range of 20 falls in the 95-percentile of all signals with cell sizes chosen from the log-normal distribution. In run 2, we increase the floor to 1/5, increasing recall at the cost of precision. The results are included in Table 3.1. Furthermore, we show the trade off in precision and recall caused by the detection floor in Figure 3.1.

Figure 3.2 shows noise statistics from run 2 of each cell in our simulations with finite SIR (i.e. part of a coincidence event). We see that most of the false negatives occur on cells with poor signal to interference ratio as well as low signal to noise ratio. We see that cells with low SIR can still be predicted, given that SNR is not too low. Especially when there are coincidences, our UNet struggles to detect cases where SNR is very low.

Examining the precision and recall for the three datasets, when there is no velocity variation and model mismatch, both precision and recall rise by a couple percentage points. In run 2, when we fix cell size, precision rises dramatically. However, recall does not increase, suggesting that false positives are less dependent on this input data. The UNet performance drops off much less than ADMM when velocity variations are enabled in simulation. This



(a)

(b)

(c)

Figure 3.2: Analysis of true positives and false negatives coincidence events. The signal-to-interference ratio (SIR) and signal-to-noise ratio (SNR) of each cell event are plotted as a scatter plot. (a) The dataset was simulated with default simulator settings. (b) The dataset contains no velocity variations, modeling cells traveling all at a constant velocity. (c) The dataset models a homogeneous cell size sample of 12 μm. All ideal impedance responses have identical amplitude.

suggests that a learned approach from data can more effectively deal with model mismatch.

Chapter 4

Discussion

Using traditional optimization algorithms to solve an inverse problem suffers greatly from noise and model mismatch. In our simulations based on our device model, the baseline drift and Gaussian noise are significant, measuring in the mega-ohms. To contrast, the impedance contributions of individual cells are measured in kilo-ohms. As such the input signals to ADMM have heavy distortion from noise, which is not explicitly modeled in our inverse problem. Combined with the fact the measured signals do not perfectly conform our ideal system response templates, even with sparsity regularization, the outputs have high sparsity compared to the ground truth.

ADMM shows promise as it performs well when processing homogeneous cell size samples. As seen in Figure 2.2, ADMM trades off the ability to identify signals with high dynamic range when increasing the regularization parameter. Increasing the sparsity in order to reduce false positives also causes smaller cells to disappear in the output. Even in the simulations with full simulation model, it is still able to produce clusters that are easily identified visually. However, without extra re-processing or more sophisticated clustering algorithms and tuning of the regularization weight, it can be more difficult to localize the peaks in the output algorithmically. These parameters must be tuned according to the cell sizes and concentration of a sample in order to be effective.

Much recent work in biomedical and computational imaging has explored unrolled deep networks, combining traditional optimization algorithms with deep networks. In MRI, ADMM has been successfully combined with convolutional neural networks in [21]. Such an approach could work for microfluidic cytometry as well.

Results from a single pass UNet on correlation filtered signals suggests that a convolutional neural network is able to learn the point spread function for cells traveling at different velocities. Our UNet shows good performance in resolving coincidence events. It mainly struggles where SNR is very poor. But this is expected, because any algorithm will fail when SNR is too low. Similar to ADMM, performance can be increased with more advanced peak finding and clustering algorithms such that extremely small peaks and noise patterns are not picked out. Consequently, we have an extra variable in post processing that trades off precision and recall. We see the UNet struggles more with false positives, especially in the heterogeneous cell size case. One possible solution to reduce false positives is to train the network with a masked loss, where false positive predictions, or pixels outside of ground truth peaks are penalized with a larger weight. A point of failure for both methods is the rare case that two cells travel at the same speed at close proximity, where it is exceedingly difficult to separate their impedance effects.

One disadvantage of a UNet is that training on large arrays requires large amounts of GPU memory. During training, we can only use a batch size of 5 on an Nvidia GeForce GTX Titan XP with 12 GB of memory. If the UNet is actually able to learn the correlation patterns, we could directly output coordinates instead of up sampling to the image domain such as in [5].

Chapter 5

Conclusion

We have presented two methods of finding cell events in node pore sensing signals. Formulating an inverse problem, ADMM is able to efficiently solve the optimization, even with an ill-conditioned forward model. ADMM does have several pitfalls, including being sensitive to hyperparameters. We show that a convolutional neural network is effective in identifying peaks from signals after matched filtering, and the UNet fails to detect cells only when SNR is extremely low or in pathological coincidence events.

Bibliography

[1] Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. "MoDL: Model-Based Deep Learning Architecture for Inverse Problems". In: *IEEE Transactions on Medical Imaging* 38.2 (Feb. 2019), pp. 394–405. ISSN: 1558-254X. DOI: 10.1109/tmi.2018.2865356. URL: http://dx.doi.org/10.1109/TMI.2018.2865356.

[2] Nick Antipa et al. *3159989.pdf*. Dec. 2017. DOI: 10.6084/m9.figshare.5640856.v1. URL: https://osapublishing.figshare.com/articles/journal_contribution/3159989_pdf/5640856/1.

[3] Nick Antipa et al. "DiffuserCam: lensless single-exposure 3D imaging". In: *Optica* 5.1 (Jan. 2018), pp. 1–9. DOI: 10.1364/OPTICA.5.000001. URL: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-5-1-1.

[4] Karthik Balakrishnan et al. "Node-Pore Sensing Enables Label-Free Surface-Marker Profiling of Single Cells". In: *Analytical chemistry* 87 (Jan. 2015). DOI: 10.1021/ac504613b.

[5] Nicholas Boyd et al. "DeepLoco: Fast 3D Localization Microscopy Using Neural Networks". In: *bioRxiv* (2018). DOI: 10.1101/267096. eprint: https://www.biorxiv.org/content/early/2018/02/16/267096.full.pdf. URL: https://www.biorxiv.org/content/early/2018/02/16/267096.

[6] Stephen Boyd et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends in Machine Learning* 3 (Jan. 2011), pp. 1–122. DOI: 10.1561/2200000016.

[7] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge Univ. Pr., 2011.

[8] Marshall Don. "The Coulter Principle: Foundation of an Industry". In: *JALA: Journal of the Association for Laboratory Automation* 8.6 (2003), pp. 72–81. DOI: 10.1016/s1535-5535(03)00023-6. eprint: https://doi.org/10.1016/s1535-5535(03)00023-6. URL: https://doi.org/10.1016/s1535-5535(03)00023-6.

[9] Peter W. Glynn. "Upper bounds on Poisson tail probabilities". In: *Operations Research Letters* 6.1 (1987), pp. 9–14. ISSN: 0167-6377. DOI: https://doi.org/10.1016/0167-6377(87)90003-4. URL: https://www.sciencedirect.com/science/article/pii/0167637787900034.

[10] M. Kellman et al. "BARKER-CODED NODE-PORE RESISTIVE PULSE SENSING WITH BUILT-IN COINCIDENCE CORRECTION". In: *Proc IEEE Int Conf Acoust Speech Signal Process* 2017 (Mar. 2017), pp. 1053–1057.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: `1411.4038`. URL: `http://arxiv.org/abs/1411.4038`.

[12] Wenjie Luo et al. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: `https://proceedings.neurips.cc/paper/2016/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf`.

[13] Vishal Monga, Yuelong Li, and Yonina C. Eldar. "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing". In: *IEEE Signal Processing Magazine* 38.2 (2021), pp. 18–44. DOI: `10.1109/MSP.2020.3016905`.

[14] Patrick Putzky and Max Welling. *Recurrent Inference Machines for Solving Inverse Problems*. 2017. arXiv: `1706.04008 [cs.NE]`.

[15] Francois Rivest et al. "Toward Real-Time Cell Detection and Characterization using Barker-Coded Node-Pore Sensing". In: Oct. 2015.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: `1505.04597 [cs.CV]`.

[17] Jonathan R Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep. USA, 1994.

[18] Michael Short. "Improved Inequalities for the Poisson and Binomial Distribution and Upper Tail Quantile Functions". In: *ISRN Probability and Statistics* 2013 (2013), pp. 1–6. DOI: `10.1155/2013/412958`.

[19] David A. Swick. "A Review Of Wideband Ambiguity Functions". In: (1969). DOI: `10.21236/ad0700657`.

[20] STEPHEN J.. RECHT BENJAMIN. WRIGHT. *OPTIMIZATION FOR DATA ANALYSIS*. CAMBRIDGE UNIV PRESS, 2021.

[21] Yan Yang et al. "ADMM-Net: A Deep Learning Approach for Compressive Sensing MRI". In: *CoRR* abs/1705.06869 (2017). arXiv: `1705.06869`. URL: `http://arxiv.org/abs/1705.06869`.

[22] Jian Zhang and Bernard Ghanem. *ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing*. 2018. arXiv: `1706.07929 [cs.CV]`.