

DECODING OF CODE-MULTIPLEXED COULTER SENSOR SIGNALS VIA DEEP LEARNING

Ningquan Wang, Ruxiu Liu, Norh Asmare, Dakshitha B. Anandakumar, and A. Fatih Sarioglu
Georgia Institute of Technology, Atlanta, USA

ABSTRACT

Code-multiplexed Coulter sensors can easily be integrated into microfluidic devices and provide information on spatiotemporal manipulations of suspended particles for quantitative sample assessment. In this paper, we introduced a deep learning-based decoding algorithm to process the output waveform from a network of code-multiplexed Coulter sensors on a microfluidic device. Our deep learning-based algorithm both simplifies the design of coded Coulter sensors and increases the signal processing speed. As a proof of principle, we designed and fabricated a microfluidic platform with 10 code-multiplexed Coulter sensors, and used a suspension of human ovarian cancer cells as a test sample to characterize the system. Our deep learning-based algorithm resulted in an 87% decoding accuracy at a sample processing speed of 800 particles/s.

KEYWORDS

Lab-on-a-chip, bioanalysis, machine learning, deep learning, convolutional neural network, Coulter sensing

INTRODUCTION

Coulter counters are routinely employed for rapid enumeration and characterization of electrolyte-suspended particles. The impedance-based detection mechanism of Coulter counters, also known as the resistive pulse sensing, provides a robust and high throughput sensing scheme that can readily be combined with the Lab-on-a-Chip (LoC) technology. LoC platforms with integrated Coulter counters, in the form of microelectrodes within microfluidic channels, were used for biomedical [1] and basic research applications [2].

To integrate multiple Coulter sensors into the same microfluidic platform, we have introduced a code-multiplexed Coulter sensor technology, the Microfluidic CODES [3]. In terms of the physical interface, the Microfluidic CODES operates similarly to a conventional Coulter sensor as it only requires a single excitation input, and yields a single output electrical waveform. However, unlike a conventional Coulter counter, the output waveform of the Microfluidic CODES contains the information from all of the Coulter sensors integrated on the microfluidic device. To achieve multiplexing, the Microfluidic CODES uses code division multiple access (CDMA), which is a technique that is commonly used in telecommunications. Specifically, each Coulter sensor in the Microfluidic CODES is micromachined to form a distinct electrode pattern that generates a signature output waveform when activated.

Following the conventional CDMA networks, the Microfluidic CODES can be designed based on an orthogonal code-set such as Gold sequences, which are mutually-orthogonal bi-polar sequences commonly used in a CDMA uplink [4]. Orthogonal code-set allows individual

sensor signals to be recovered via correlation with a set of code templates with minimum crosstalk. However, reliance solely on orthogonal codes constraints the design of the individual sensors [5], and the template-based decoding of sensor network signal limits the signal processing speed. Therefore, a straightforward sensor design based on a flexible coding scheme coupled with a more efficient decoding algorithm can help enhance the utility of the Microfluidic CODES in creating LoC platforms with integrated Coulter sensor networks.

Machine learning-based algorithms are often used in solving complex pattern recognition problems. Recently, deep learning has emerged as a key learning model, especially in the processing of time-series [6]. More specifically, deep learning is a representation learning method, which allows a machine to be fed with raw data, and to automatically learn and discover the representations needed for performing further pattern recognition on the input data. A deep neural network is a deep learning structure. Like the vast network of neurons in a brain, a deep neural network is based on a collection of multilayered, interconnected nodes called artificial neurons. These artificial neurons makeup multiple computational layers, and each layer performs a non-linear transformation on the output of the previous layer. In this way, the input signal is approximated by a hierarchy of features from lower levels to higher levels layer by layer. Because of this non-linear multi-layer structure, a deep neural network offers great potential in representing complex functions and solving highly non-linear problems. Specifically, deep learning has been successfully implemented in speech recognition [7], human activity recognition [8], and patient-specific electrocardiogram (ECG) classification [9].

In this paper, we couple the Microfluidic CODES with a deep learning-based signal processing algorithm. We design the algorithm based on a convolutional neural network (ConvNet), which is a specific type of deep neural network. Then we acquire training signals by processing human cancer cells using a Microfluidic CODES device. We train the ConvNet to compute the probability with which a given sensor signal belongs to each and every coded Coulter sensor in the network. Finally, we query the ConvNet to characterize its accuracy and throughput.

METHODS

Design of Coded Sensors

We designed a network of 10 code-multiplexed Coulter sensors. The assigned code for each Coulter sensor was a randomly generated 15-bit bipolar digital code (i.e., each bit was treated as a Bernoulli random variable with $p = 0.5$) (Figure 1). The code-set then dictated the physical design of the Coulter sensor network. Specifically, the sensor network consisted of three coplanar electrodes,

namely a common electrode to excite the sensor network, and two sensing electrodes, one positive and one negative, to acquire sensor signals. Individual coded sensors were created by placing positive and negative 5 μm -wide electrode fingers in a sequence that follows the assigned code and routing the common electrode meandering between these coding electrode fingers. (Figure 1).

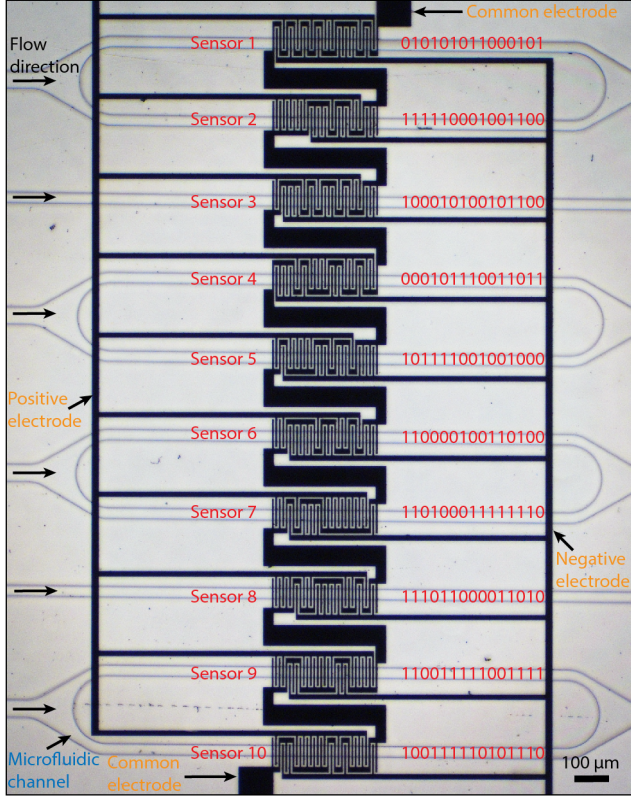


Figure 1: A microscope image of the fabricated microfluidic device with a network of 10 code-multiplexed Coulter sensors.

Fabrication

As a test platform, we fabricated a microfluidic device with 10 parallel channels, each of which included a coded Coulter sensor (Figure 1). The device consisted of a glass substrate with micromachined electrodes and a polydimethylsiloxane (PDMS) microfluidic layer. Briefly, the electrodes on the substrate were fabricated using a lift-off process. A 20/480 nm Cr/Au stack was micropatterned on the substrate to form three interdigitated coplanar electrodes described earlier. The microfluidic layer was fabricated using a soft lithography process. A 15- μm -thick SU-8 mold was fabricated via photolithography, and then used to cast the PDMS layer with 10 parallel 30- μm -wide microfluidic channels. The glass substrate and PDMS layer were activated under O_2 plasma, aligned with a microscope, and bonded together to form the final device. The detailed fabrication process can be found in [10].

Sample Preparation

To test the device, we used human ovarian cancer cells (HeyA8) suspended in phosphate buffered saline (PBS) as a biological sample. The cells were cultured in a cell culture flask containing RPMI 1640 cell culture medium supplemented with fetal bovine serum (FBS) at a ratio of 9:1 (v/v). The cell culture was kept in an incubator that was

maintained at 37 $^{\circ}\text{C}$ in 5% CO_2 atmosphere. Once reached >80% confluence, the cells were treated with trypsin, centrifuged, and resuspended in PBS with gentle pipetting. The concentration of the cells in the suspension was set to 10^5 cells/mL using a Nageotte chamber.

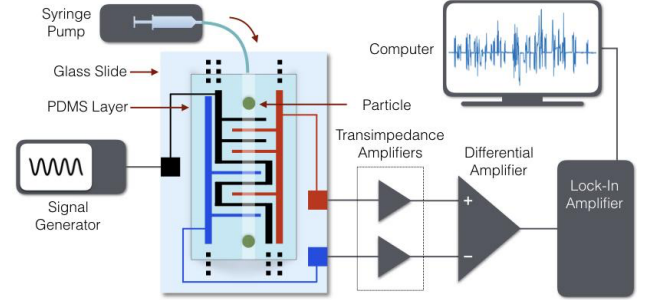


Figure 2: A schematic showing the experimental setup used to test the decoding of sensor network signals.

Experiment Setup

The experimental setup included a syringe pump for sample delivery, electronic hardware for signal acquisition and processing, and an inverted optical microscope for data validation. The cell suspension was driven into the microfluidic device at a constant flow rate (500 $\mu\text{L}/\text{h}$). The sensor network was excited by applying a 460 kHz, 1 Vpk sine wave to the common electrode. The sensor network output signal was obtained by first converting the current signals into voltages using transimpedance amplifiers and then subtracting one from the other using a differential amplifier. The differential signal was demodulated using a lock-in amplifier and subsequently low-pass filtered in the digital domain. During electrical measurements, the cell flow over the code-multiplexed sensor network was optically recorded using an inverted optical microscope equipped with a high-speed camera. As all of the sensors were within the microscope field of view, the recorded footage was later used for data validation.

Data Preparation

To build the training data set, we first implemented a signal-identification program to extract all the sensor signals from raw sensor output, and classify each identified signal as non-interfering or interfering (Figure 3a). Then a data augmentation process was implemented to enlarge the training data size. To perform the data augmentation, we first randomly selected a non-interfering signal collected in the previous step, and applied variations in three signal parameters: power, duration, and time delay (Figure 3b). For the variation in power, we randomly chose a scaling factor within a set range, determined by the expected variation of cancer cell size, and varied the power of the selected signal accordingly. A similar procedure was also performed on the variations in duration and time delay. Then an additive white Gaussian noise (SNR = 30 dB) was added to the signal. This pick-and-vary process was repeated to construct a “signal database” containing 1,000,000 variations of the original non-interfering sensor signals. 150,000 signals were randomly selected from the constructed signal database and were used as the training data set for non-interfering events. The remaining 850,000 signals were then randomly interfered in pairs to construct the training data set for resolving coincidence events. This

augmented training data set was then used to train the ConvNet (Figure 4a).

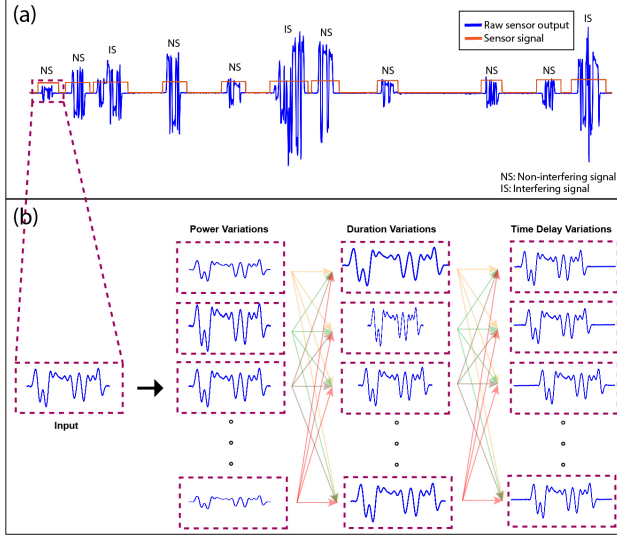


Figure 3: Data augmentation. (a) Sensor signals were extracted from a raw sensor output signal. (b) Each non-interfering sensor signal was varied in the power, duration, and time delay to construct a signal database to form a large training dataset.

Implementation of the ConvNet

We built a ConvNet that contained four convolutional layers (Figure 4b). The first convolutional layer (Conv-1) had 32 convolutional kernels of size 5, resulting in 192 trainable parameters. The output feature map was then processed by Conv-2, which also had 32 convolutional kernels of size 5, and a total of 5152 trainable parameters. A max-pooling layer was placed right after Conv-2 to down-sample the convolved signal, and the output was fed into Conv-3, which contained 64 kernels and 10304 trainable parameters, and then Conv-4, which contained 64 kernels and 20544 trainable parameters. Another max-pooling layer was placed after Conv-4. Following the second max-pooling layer were a flatten layer and two dense (fully connected) layers, in which the first dense layer had 180224 trainable parameters and the second had 640 trainable parameters. The model had a total of 217056 trainable parameters. The second dense layer fed the final output layer, which had 10 nodes, representing 10 Coulter sensors (10 classes). Given an input signal, the ConvNet predicted the probability with which the input signal belonged to each and every Coulter sensor in the microfluidic device. A predetermined probability threshold was used to predict the identity of the activated sensor (Figure 4c).

RESULTS

Training of the ConvNet

The ConvNet was trained with a batch size of 500 (batch size: the number of training signals processed before the model is updated), and an epoch number of 30 (epoch number: the number of times the learning algorithm works through the entire training dataset). The binary cross entropy (BCE) was used as the loss function to calculate the classification error. The Adam optimizer was used to minimize the error in each iteration. The learning rate was

set to 0.001 for the first 10 epochs, 0.0001 for epochs 11 through 20, and 0.00001 for epochs 21 through 30.

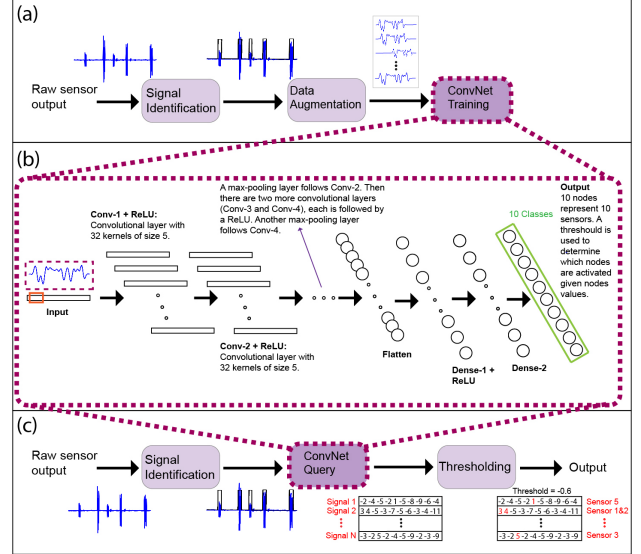


Figure 4: ConvNet implementation. (a) The training process of the ConvNet. (b) The structure of the ConvNet. (c) The query process of the ConvNet.

As the number of training epoch increased, the ConvNet improved itself to better represent the input data. After 25 epochs, the training and testing losses remained below 0.11 and 0.13, respectively, and the training and testing accuracies remained above 95% and 86%, respectively (Figure 5).

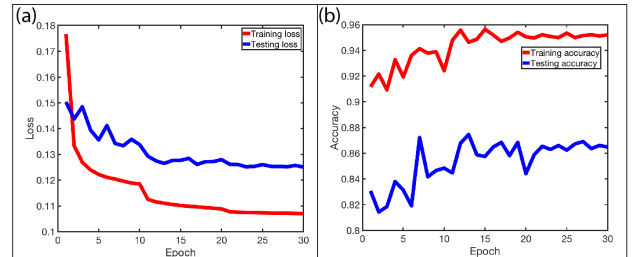


Figure 5: Training and testing performances of the ConvNet. (a) Training and testing losses. (b) Training and testing accuracies.

Query of the ConvNet

For non-interfering sensor signals (Figure 6, the first and second rows), the ConvNet output for the corresponding sensor was close to 100%, while outputs for other sensors were nearly 0%. In this case, we could easily identify the activated sensor. For interfering sensor signals due to coincident particles (Figure 6, the third row), the corresponding output probabilities were reduced but the activated sensor identity could be correctly determined by using a predetermined threshold value (33% for this work).

The query speed was over 800 particles/s, which represented an $\sim 100\times$ increase over the template-based algorithm demonstrated in our earlier work [11]. Query speeds of this order can potentially enable real-time particle analysis in LoC devices with code-multiplexed sensor networks.

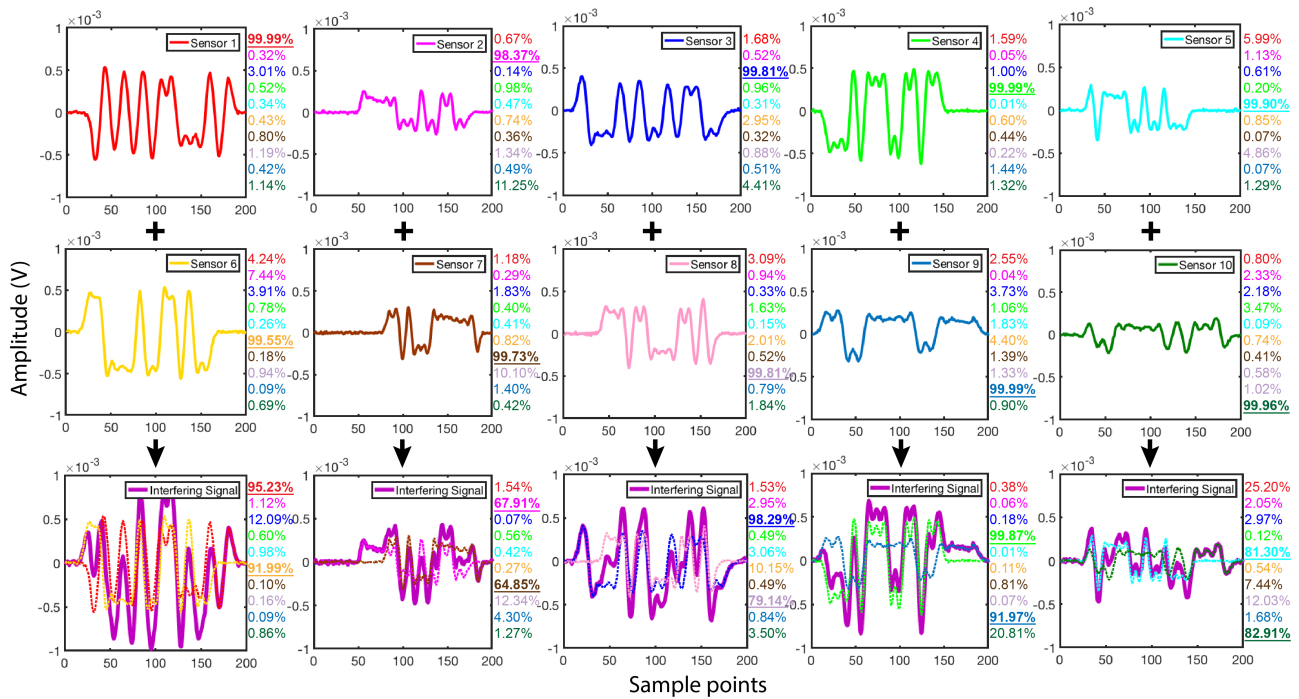


Figure 6: The query process of the ConvNet on both non-interfering and interfering signals. The first two rows demonstrate the results from representative non-interfering signals that correspond to all 10 code-multiplexed Coulter sensors in the network. The third row demonstrates the results for interfering signals. Each signal in the third row is a combination of two non-interfering signals from the first two rows in the same column.

CONCLUSION

We introduced a deep learning-based pattern recognition algorithm to decode code-multiplexed Coulter sensor signals in microfluidic devices. Besides simplifying the physical design of code-multiplexed Coulter sensor networks, our work significantly improved the processing speed of sensor signals to a level that can potentially enable real-time applications.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation, USA (Award no. ECCS 1610995, no. ECCS 1752170), and the Arnold and Mabel Beckman Foundation, USA (Beckman Young Investigator Award to A.F.S.).

REFERENCES

- [1] Z. Mei, *et al.*, “Counting Leukocytes from Whole Blood Using a Lab-on-a-chip Coulter Counter,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Diego, CA, 2012, pp. 6277-6280.
- [2] J. Rickel, *et al.*, “A Flow Focusing Microfluidic Device with an Integrated Coulter Particle Counter for Production, Counting and Size Characterization of Monodisperse Microbubbles,” *Lab Chip*, vol. 18, pp. 2653-2664, 2018.
- [3] R. Liu, *et al.*, “Microfluidic CODES: A Scalable Multiplexed Electronic Sensor for Orthogonal Detection of Particles in Microfluidic Channels,” *Lab Chip*, vol. 16, pp. 1350-1357, 2016.
- [4] R. Gold, “Optimal Binary Sequences for Spread Spectrum Multiplexing,” *IEEE Trans. Inf. Theory*, vol. 13, pp. 619-621, 1976.

- [5] R. Liu, *et al.*, “Scaling Code-multiplexed Electrode Networks for Distributed Coulter Detection in Microfluidics,” *Biosens. Bioelectron.*, vol. 120, pp. 30-39, 2018.
- [6] M. Langkvist, L. Karlsson, A. Loutfi, “A Review of Unsupervised Feature Learning and Deep Learning for Time-series Modeling,” *Pattern Recognit. Lett.*, vol. 42, pp. 11-24, 2014.
- [7] G. Hinton, *et al.*, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Process. Mag.*, vol. 29, pp. 82-97, 2012.
- [8] A. Ignatov, “Real-Time Human Activity Recognition from Accelerometer Data Using Convolutional Neural Networks,” *Appl. Soft Comput.*, vol. 62, pp. 915-922, 2018.
- [9] S. Kiranyaz, T. Ince, M. Gabbouj, “Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks,” *IEEE Trans. Biomed. Eng.*, vol. 63, pp. 664-675, 2016.
- [10] N. Wang, R. Liu, A. F. Sarioglu, “Microfluidic Platform with Multiplexed Electronic Detection for Spatial Tracking of Particles,” *J. Vis. Exp.*, vol. 121, e55311, 2017.
- [11] R. Liu, *et al.*, “Design and Modeling of Electrode Networks for Code-division Multiplexed Resistive Pulse Sensing in Microfluidic Devices,” *Lab Chip*, vol. 17, pp. 2650-2666, 2017.

CONTACT

*A. F. Sarioglu, tel: +1-404-8945032;
sarioglu@gatech.edu