University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

5-2022

Machine learning analysis of acoustic attenuation measurements for cellular characterization.

John T. Moore
University of Louisville

MACHINE LEARNING ANALYSIS OF ACOUSTIC ATTENUATION

MEASUREMENTS FOR CELLULAR CHARACTERIZATION

By

John T. Moore
B. S. Bioengineering, University of Louisville, May 2021

A Thesis
Submitted to the Faculty of the
University of Louisville
J.B. Speed School of Engineering
As Partial Fulfillment of the Requirements
For the Professional Degree

MASTER OF ENGINEERING

Department of Bioengineering

May 2022

ACKNOWLEDGEMENTS

I would first like to thank Dr. Jonathan Kopechek for being a mentor to me throughout my college career. You have been an invaluable part of my success thus far. I am extremely grateful for your guidance and support throughout my undergraduate and Masters years and I am excited to continue working with you into my Ph.D.

Next, I would like to thank the following University of Louisville faculty members for agreeing to be part of my defense committee and for the time they have spent reviewing this thesis to ensure it will add to the body of scientific knowledge:

Dr. Hermann Frieboes   Dr. Kavitha Yaddanapudi

I would also like to thank the members of Dr. Kopechek's Theranostic Ultrasound Laboratory for their help on this thesis, specifically Ekaterina "Katya" Kovatsenko and Zachary Fowler, as well as Mariah Priddy for her assistance back to when I was first starting at the lab. I also want to thank Connor Centner for being a role model of mine and another mentor to me throughout my time working at the lab. I wish you the best of luck in all your future endeavors.

Finally, I must thank my parents and my friends for all their unwavering support and motivation throughout my academic career and life. My achievements so far have stemmed from you all, and I cannot thank you enough for all you have done for me.

ABSTRACT

T-cell therapies have been gaining popularity in recent years due to their cancer fighting potential. With remission rates improving in this field of immunotherapy, the demand for T-cell therapies has also increased; however, the cell processing techniques for these therapeutic products have yet to rise to the level of demand. The manufacturing process takes too long and a significant amount of processed cell batches can fail to meet safety requirements. These limitations of cell processing can be detrimental to patients seeking T-cell therapies. While current products have improved the time it takes to manufacture these therapeutic products, there is still a lack of an in-line non-destructive sample quality control monitoring system to reduce the risk of batch failures and delays. In this thesis, theoretical and experimental testing was conducted to serve as a proof of concept that machine learning analysis of acoustic attenuation signals could be utilized for cellular characterization. A machine learning analysis method was able to determine sizes of theoretical microparticles and concentrations of different cell lines from acoustic attenuation signals collected in a static ultrasound chamber, as well as a continuous flow ultrasound chamber. It was found that the machine learning technique called scratch learning generally served as a better model for these cellular characterization trials, rather than transfer learning. With further refinement of the scratch learning architecture, as well as the further development of the attenuation signal acquisition system, optimization of classification accuracy of the machine learning analysis method could be further improved. This optimization could enable an in-line ultrasound-based quality control module for classification of multiple cell characteristics to be implemented into cell processing procedures for T-cell therapies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I.   INTRODUCTION

A. Cell Processing

The idea for this research stems from an invention disclosure that was submitted by the University of Louisville Theranostic Ultrasound Laboratory in 2020. The main goal of the invention disclosure was to address the unmet needs of an improved cell manufacturing process to enable more widespread adoption of T-cell therapies. With these life-saving therapies achieving higher remission rates compared to conventional treatments, some with efficacy exceeding 80%, there is currently a higher demand for these T-cell therapies [9].

The interest in using a patient's own immune cells to treat their diseases has skyrocketed in the past years. Billions of dollars are being invested into new therapies that utilize this concept. For example, Bristol Myers Squibb acquired Celgene for $74 billion in 2019, and sales for these types of companies are forecasted to be in the billion-dollar range as well due to their cell therapies [17]. Despite this significant growth in cell therapies, the process of manufacturing the therapeutic products is still challenging [8]. The current processing time takes four or more weeks before processed cells can be re-infused into a patient, and there is a significant number (10% of runs) of processed cell batches that routinely fail to meet quality standard requirements [8]. Quality failures of T-cell therapies can result in significant side effects that can cause permanent multiorgan damage, such as cytokine release syndrome, in addition to neurologic toxicity and an increased risk of pathogenesis [19].

There have been several automated cell processing systems that have come to the market to address these limitations. These systems, while improving cell processing times, are lacking in-line sample quality control monitoring system. Current cell processing

products still perform product testing and characterization off-line with samples that were taken from the final cell therapy product that cannot be used in patients afterwards. This current procedure can be considered a waste of time and material, and needs to be addressed, especially since CAR T-cell therapy could cost an average individual patient between $373,000 and $475,000 [3]. Redoing or delaying a patient's therapy treatment could be detrimental. One study showed that when increasing the wait time of receiving CAR T-cell therapy from 1 to 9 months, the 1-year mortality rate would increase from 36.1% to 76.3% [20]. Needless to say, it is paramount to the patient's health that cell processing runs efficiently.

To address this, researchers at the University of Louisville submitted an invention disclosure to develop a new "acoustofluidic" cell processing module that would consist of single-use sterile flow chamber cartridges and two ultrasound transducers for acoustic detection and machine learning analysis. The module would utilize low-energy ultrasound waves for non-destructive, in-line characterization and cell separation/sorting. Having a module like this could improve the manufacturing process of the cell therapy T-cells as well as improve batch-to-batch consistency.

In this thesis, acoustic attenuation measurements were investigated. Acoustic attenuation is the decrease in ultrasound pressure amplitude due to scattering or absorption of energy by cells or particles within the ultrasound field, which occurs due to changes in acoustic impedance caused by differences in density and speed of sound of materials. The measurements of the acoustic attenuation can give an insight on the acoustic properties of any particles or cells in solution because ultrasound waves will reflect or scatter off interfaces that have a different density or speed of sound compared to the surrounding

media [12]. Acoustic attenuation would be influenced by particle size, the concentration of those particles, and potentially other properties, such as shape, that cause differences in the particle's density or speed of sound. A larger cell or particle would most likely have a higher attenuation when compared to a smaller cell or particle of the same material. Even when cells or particles are at the same concentration, the set of larger cells would have a higher acoustic attenuation than a set of smaller cells. Cell aggregation could also impact acoustic attenuation measurements. Acoustic attenuation can indicate differences in cell properties similar to methods such as flow cytometry, which can sense cell properties using optical detection. The main difference is that flow cytometry utilizes lasers to detect fluorescence and light scattering, while acoustic attenuation uses ultrasound waves.

There are multiple published instances of prior research utilizing machine learning analysis on flow cytometry measurements. Different machine learning algorithms, such as support vector machines with kernel methods, convolutional neural networks, and deep neural networks, have been used for such analysis [1, 18, 15]. These publications have shown the possibility of use machine learning algorithms for cellular characterization from light scattering imagery and other mediums. In contrast, this thesis applied the idea of characterizing cells with machine learning using acoustic attenuation measurements instead.

This thesis is a proof of concept for the cell characterization and machine learning portions of that invention disclosure. The main focus was to determine if a machine learning algorithm could distinguish theoretical signals of varying micro-particle sizes as well as experimentally acquired signals of varying cell concentrations. Other properties, including activation status, viability, and possibly identification of sample contamination,

a major limitation of current quality control plans in cell therapies, could be investigated in later research projects [10]. The machine learning methods tested in this thesis utilized spectrograms of advanced ultrasound frequency-sweep attenuation waveforms to train their convolutional neural network (CNN) models.

B. Machine Learning

Over the past years, machine learning has seen a rise in popularity. Machine learning has revolutionized data analysis in so many fields, from the commercial world to bioinformatics [14]. There are many advantages to machine learning. Some examples would be that machine learning can review large volumes of data and easily identify patterns, it can be continuously improved, it has a wide range of applications, and it is completely automated [2, 5]. Since the research conducted for this thesis was to be a proof of concept for utilizing acoustic attenuation measurements for cell characterization, using machine learning could be beneficial for an in-line quality control module for cell processing. The machine learning models can be trained and then used in a product on the market. The commercial module would take advantage of the fact that machine learning can work efficiently all the time without human intervention needed, and can up upgraded at any time in order to improve the accuracy of its analysis. Utilizing machine learning for the future in-line fully-automated acoustofluidic quality control module is the most logical path.

The two machine learning methods investigated were transfer learning and scratch learning. Transfer Learning is a machine learning method of taking a prebuilt and pretrained CNN, modifying it, and retraining it for another task. This method is used to improve a model for one domain by transferring information from another domain. An

example of transfer learning would be two people learning how to play the piano. One person already has musical knowledge from playing another instrument, and the other has no musical knowledge. Transfer learning is like the person who already knows how to play an instrument, and applying that knowledge to learning a new instrument [21]. Scratch learning is a machine learning method, where instead of taking an already pretrained CNN, modifying it, and retraining it like in transfer learning, the programmer builds all layers of the CNN themselves.

The deep learning CNN algorithm was chosen for this research over other machine learning algorithms due to its ability to analyze large data sets efficiently and automatically detect important features without human supervision [4, 7]. In addition, CNN models can be run on any device, making it an ideal choice for a future in-line, non-destructive, fully-automated acoustofluidic quality control module for cell processing.

<u>C. Objective</u>

The objective of this thesis was to compare two different machine learning CNN methods for detection of two properties, size and concentration, of theoretical microparticles and two cell lines, respectively, from ultrasound attenuation measurements. This thesis serves as a proof of concept for utilizing machine learning for analysis of acoustic attenuation measurements for cell characterization.

II. METHODS

A. Static Cuvette Acoustic Attenuation Setup

A device system was designed at the University of Louisville Theranostic Ultrasound Laboratory that consisted of a 3-D printed container for two ultrasound transducers and a cuvette. This design was developed in SolidWorks. The SolidWorks part model and the completed device can be seen in Figures 1 and 2, respectfully.
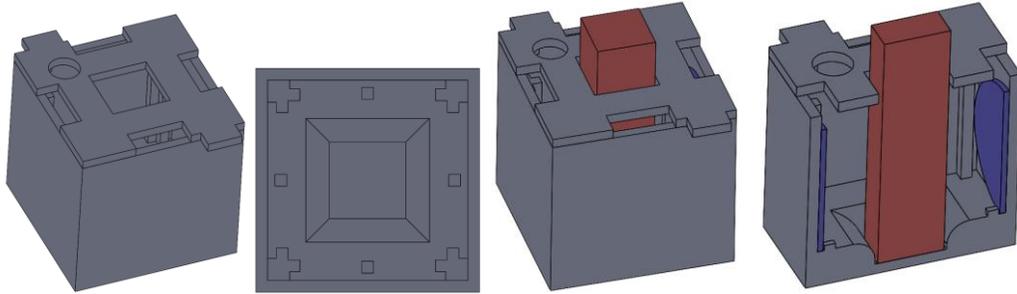
Figure 1: Renderings of the static cuvette acoustic attenuation device as a SolidWorks assembly. The gray parts constitute the device, while the red and blue parts indicate where the cuvette and ultrasound transducers are placed, respectfully.
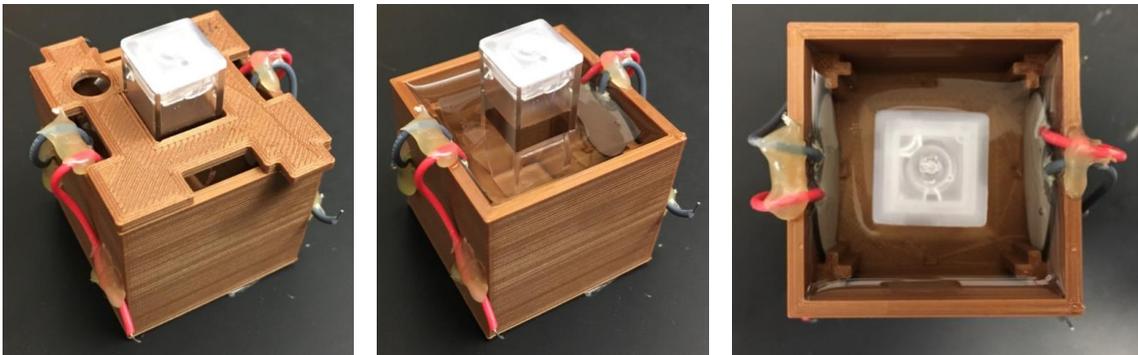
Figure 2: Images of the printed and completed static cuvette acoustic attenuation device.

Within this device, one ultrasound transducer transmits an ultrasound frequency-sweep ("chirp") of 3.0 MHz to 3.6 MHz through the cell sample within the cuvette, as well as the water surrounding the cuvette, and would be received by the second transducer. The

frequency of the chirp was based on the fundamental frequency of the transducer used, which was determined to be 3.3 MHz. The frequency chirp was created as an arbitrary waveform in MATLAB and a Rigol DG800 Function Generator was used power the transmitting ultrasound transducer. As the generated ultrasound waveform passes through the sample, the strength of the waveform is reduced, or attenuated. The attenuated signal was then acquired by the receiving ultrasound transducer and displayed on a Rigol DS1000Z-E Oscilloscope. This acoustic attenuation signal data, shown on the oscilloscope, was collected and processed in MATLAB. All attenuation waveform signals were converted into spectrograms to be used in the machine learning training. A stir bar was added into the sample cuvette to ensure the sample was well mixed and the cells did not all fall to the bottom of the cuvette during measurement acquisition. The entire static cuvette acoustic attenuation setup is shown in Figure 3.
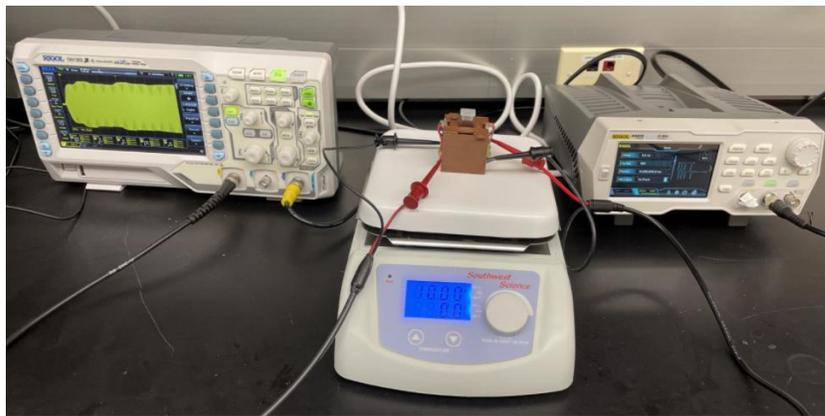
Figure 3: Image of entire static cuvette acoustic attenuation setup. Function generator (on right) is connected to the transmitting ultrasound transducer of the device. Device is placed on hot plate (in middle). No heat was used. The hot plate was only set to stir. Oscilloscope (on left) is connected to receiving ultrasound transducer of the device, and is displaying attenuation waveform.

Initial testing of the static cuvette acoustic attenuation setup was conducted with solutions of varying cationic microbubble concentrations, shown in Table 1 [6]. This was done to determine if the static setup could successfully acquire attenuation waveforms. These waveforms were used in the calculation of the attenuation coefficient for each concentration based on the following equation from Kopechek et. al:

$$\alpha_{dB}(f) = \frac{10 \log_{10}\left(\frac{\langle S_r(f)\rangle}{\langle S_s(f)\rangle}\right)}{x},$$

(1)

where $\langle S_s(f)\rangle$ and $\langle S_r(f)\rangle$ are the average power spectrum with and without scatters, respectfully, and x is the acoustic path length within the sample [13].

Table 1: Table of all concentrations of Cationic Microbubbles in solution used.

	Cationic Microbubble
Concentration (v/v)	0.5%
	1.0%
	2.5%
	5.0%
	10.0%

Once the initial testing of the static cuvette acoustic attenuation setup was complete, samples of varying red blood cell concentrations and Jurkat T-cell concentrations, shown in Table 2, were then tested individually in the setup. The starting concentrations were determined using a hematology analyzer for the red blood cells and a hemocytometer for the Jurkat T-cells. A trypan blue assay was used to count the number of viable Jurkat T-cells. Subsequent concentrations were created with serial dilutions. The red blood cells came from porcine whole blood collected from an abattoir. They were isolated via serial centrifugation and were stored in CPD/AS-3 solution for up to 4 weeks. The Jurkat T-cells were cultured with complete RPMI-1640 (10% fetal bovine serum, 1%

penicillin/streptomycin) at 37°C and 5% CO_2 in a flat-bottom tissue culture flask. They

were harvested when 70-90% confluent and were resuspended in complete RPMI-1640 at

a concentration of 100,000 cells per mL after centrifugation at 1500 g for 5 min at 4°C.

Examples of experimentally acquired attenuation waveforms, along with the Fast Fourier

Transform of each waveform, can be seen in Figures 4 through 7.

Table 2: Table of all concentrations of Red Blood Cells and Jurkat T-Cells used.

	Cell Types	
	Red Blood Cells	Jurkat T-Cells
	187.5	0.3125
	325	0.625
Concentrations (million cells per mL)	750	1.25
	1500	2.5
	3000	5
	6000	10
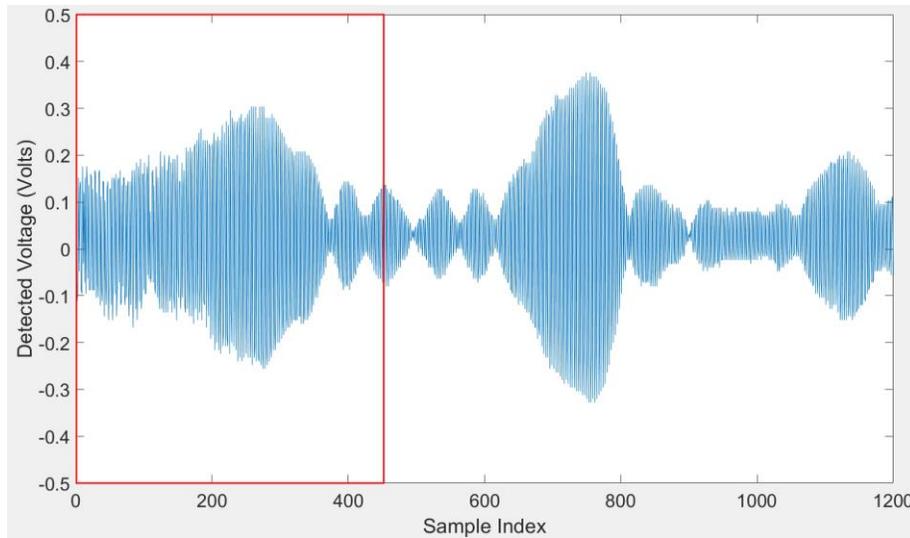
Figure 4: Acoustic attenuation waveform from static cuvette acoustic attenuation setup
for 6 billion red blood cells per mL. The red rectangle represents the Hanning window
applied to the waveform for spectrogram creation.
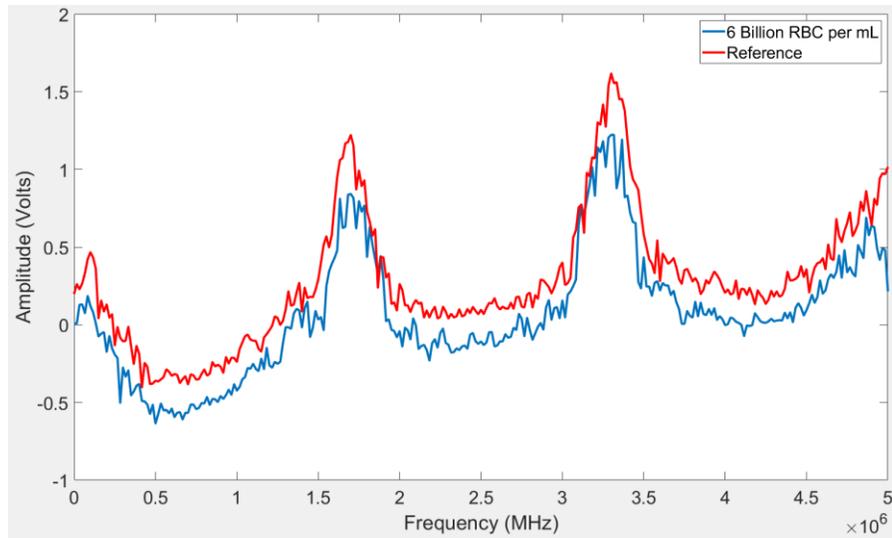
Figure 5: Fast Fourier Transform (FFT) spectra of acoustic attenuation waveform from static cuvette acoustic attenuation setup for 6 billion red blood cells per mL vs reference FFT for only deionized water showing how waveform become attenuated with addition of cells. Harmonics produced by the transducer can be observed around the fundamental frequency of approximately 3.3 MHz.
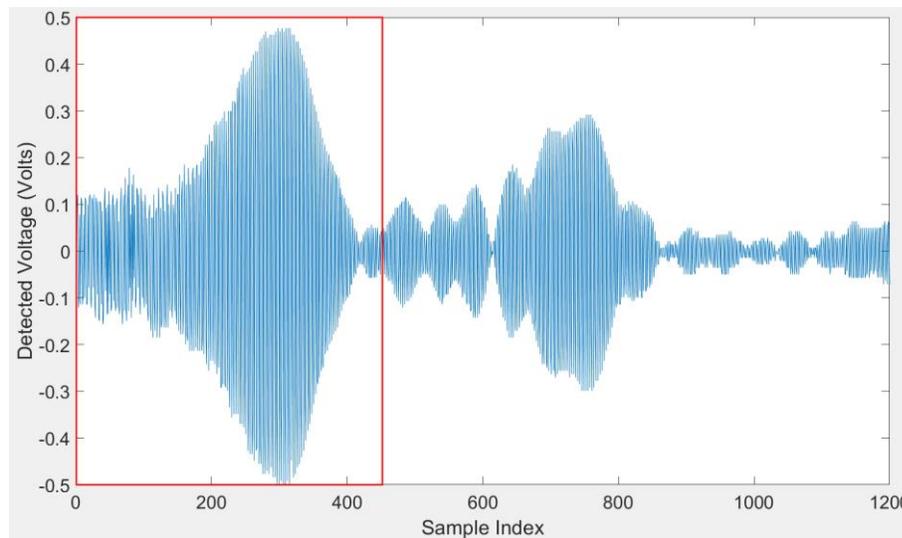
Figure 6: Acoustic attenuation waveform from static cuvette acoustic attenuation setup for 10 million Jurkat T-cells per mL. The red rectangle represents the Hanning window applied to the waveform for spectrogram creation.
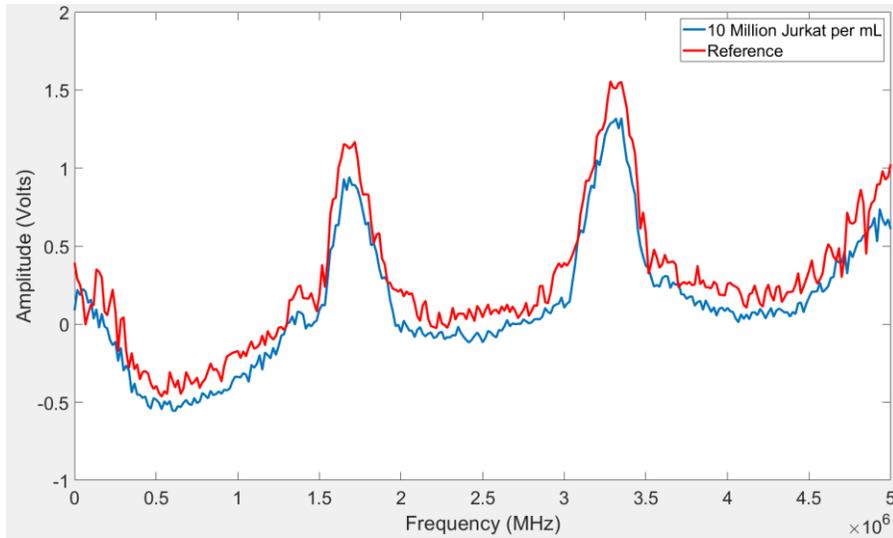
Figure 7: Fast Fourier Transform (FFT) spectra of acoustic attenuation waveform from static cuvette acoustic attenuation setup for 10 million Jurkat T-cells per mL vs reference FFT for only deionized water showing how waveform become attenuated with addition of cells. Harmonics produced by the transducer can be observed around the fundamental frequency of approximately 3.3 MHz

The acoustic attenuation waveforms acquired from the static cuvette acoustic attenuation setup for all concentrations of red blood cells and Jurkat T-cells were used to calculate the attenuation coefficients for each concentration with Equation 1. Statistical analysis was performed on all experimentally measured and calculated attenuation coefficients for the concentrations of cationic microbubbles, red blood cells, and Jurkat T-cells using ANOVA for comparison of multiple experimental groups. Tukey tests were also performed for post-hoc analysis. These waveforms were preprocessed with a Hanning window which was applied to the first 450 samples in each signal, as denoted with a red rectangle in Figures 4 and 6, and were converted into spectrograms to be used in the training, validation, and testing of the machine learning models. A total of 18,000 spectrograms were created (1,500 of each concentration) for the red blood cells and Jurkat T-cells. Example spectrograms are shown in Figures 8 and 9.
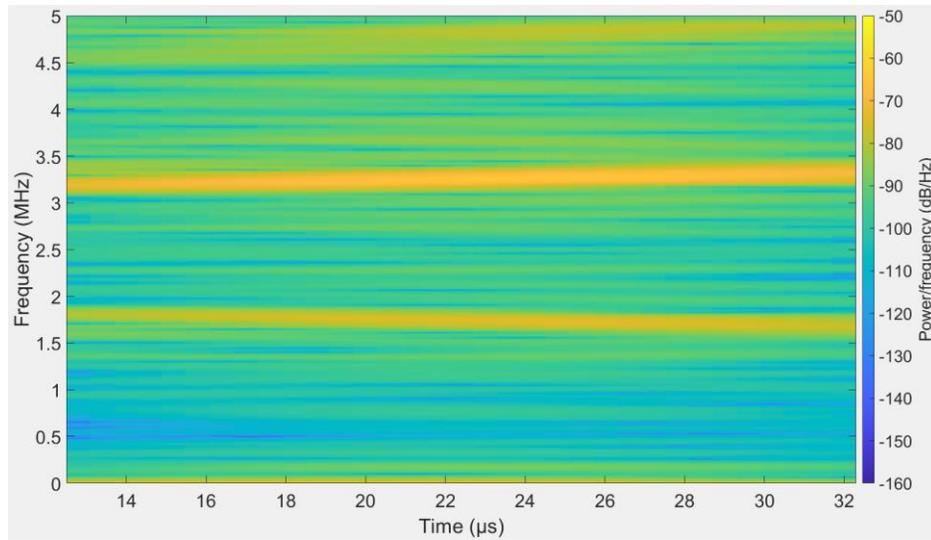
Figure 8: Spectrogram of signal from static cuvette acoustic attenuation setup for 6 billion (6000 million) concentration red blood cells per mL.
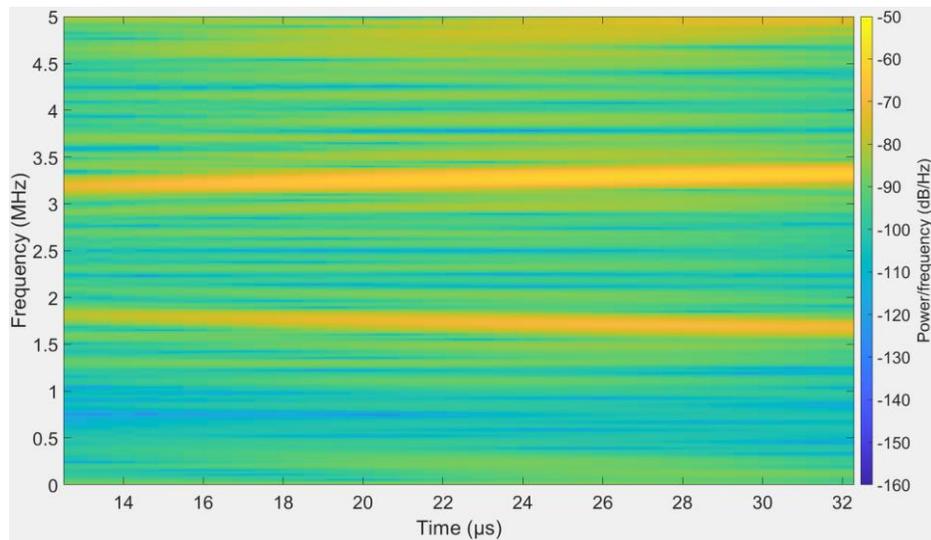
Figure 9: Spectrogram of signal from static cuvette acoustic attenuation setup for 10 million Jurkat T-cells per mL.

B. Theoretical Microparticle Sizes

A MATLAB program, based on the work done by Kopechek, et al, was created by

researchers at the University of Louisville Theranostic Ultrasound Laboratory to simulate

a frequency sweep, like what was used in the static cuvette acoustic attenuation setup, passing through a sample of polystyrene microparticles of with varying radius sizes of 40 µm, 50 µm, and 60 µm, with a density of 1055 g/cm^3 and a speed of sound of 2380 m/s [13]. The MATLAB program produced the theoretical attenuation waveforms of a solution containing those varying sized microparticles, using the following equation:

$$\alpha_{dB}(f) = \sum_m \left(n(a_m) \frac{2\pi}{k^2} \sum_{q=0}^{q_{max}} (2q+1)|A_{q,m}|^2 \right), \tag{2}$$

where m indexes bins corresponding to microspheres of radius a_m, $n(a_m)$ is the number density of microspheres of radius a_m, and k is the wavenumber outside the scatter corresponding to frequency, f. $A_{q,m}$ is the amplitude of scattered partial waves, q indexes over the spherical Bessel function order used in computing $A_{q,m}$, and q_{max} is the highest order spherical Bessel function used [13]. An example is shown in Figure 10. Gaussian white noise was added to the waveforms with a mean of 0 and a variance of 0.001. Then the waveforms were converted into spectrograms for the machine learning model training and validation. An example is shown in Figure 11. A total of 3,000 spectrograms were created (1,000 of each size) for the theoretical microparticles.
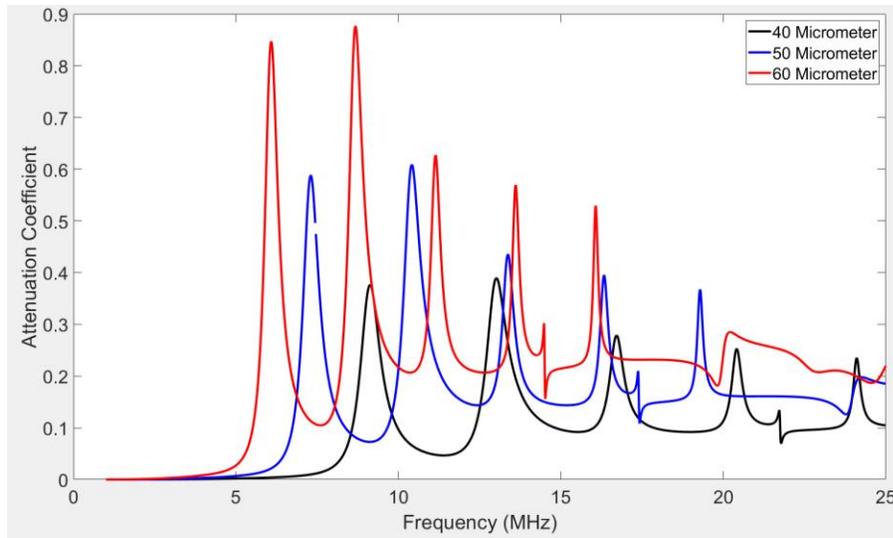
Figure 10: Theoretical data of acoustic attenuation coefficient over frequency sweep through solution with 40, 50, and 60 µm microspheres (200 per mL).
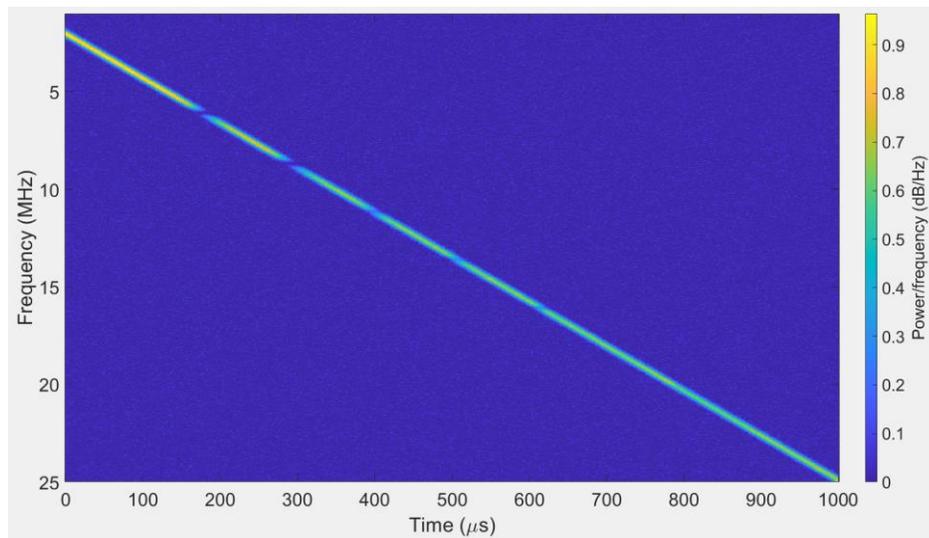
Figure 11: Theoretical data of spectrogram for frequency sweep through solution with 60 µm microspheres (200 per mL).

C. Machine Learning Methods

Once all spectrograms were created, two different machine learning methods for CNNs were investigated: Transfer Learning and Scratch Learning. In this research, the

14

GoogLeNet CNN was used for transfer learning. This architecture consists of twenty-two layers and was pretrained on the ImageNet data set. Utilizing a CNN that was pretrained on the ImageNet data set was useful for the type of classification in this research since spectrograms are visual representations of a signal waveform through a spectrum of frequencies as it changes over time.

The training parameters of the transfer learning model consisted of the "adam" optimizer, a piecewise learning rate schedule, a learning rate drop period of 20, and a max epochs of 30. All other training parameters were set to default. These parameters were chosen from a tutorial course for using deep learning techniques in MATLAB. They were not changed for this use of this research, but optimization of these parameters could lead to shorter training times and better accuracy of models in the future [16].

The scratch learning model developed for this thesis consisted of thirty layers. All training parameters for the model were set to default other than the "adam" optimizer, a piecewise learning rate schedule, a learning rate drop period of 15. These parameters and the architecture of the scratch learning model were based on those from a tutorial course for using deep learning techniques in MATLAB [16].

The experiments above were conducted to explore the analysis of acoustic attenuation signals by machine learning methods in order to determine which method, transfer or scratch learning, would be better suited for a future in-line acoustofluidic attenuation module for cell characterization. This thesis represents a preliminary proof of concept and suggests areas for further optimization of the machine learning techniques tested. This thesis provides a starting point for further research concerning the use of machine learning analysis for characterizing cell properties.

III.   RESULTS/DISCUSSION

A. Acoustic Attenuation Coefficient Measurements

Experiments conducted with varying cationic microbubble concentrations were initially done to determine if the attenuation signals collected from this setup were accurate. Instead of converting the attenuation waveforms into spectrograms, the attenuation coefficient was calculated for these samples using Equation 1 [13]. The attenuation coefficient measurements, shown in Figure 12, follow what would be expected: the higher the concentration of microbubbles in the sample, the higher the attenuation coefficient. There is a significant difference between the two higher concentrations and the three lower concentrations.
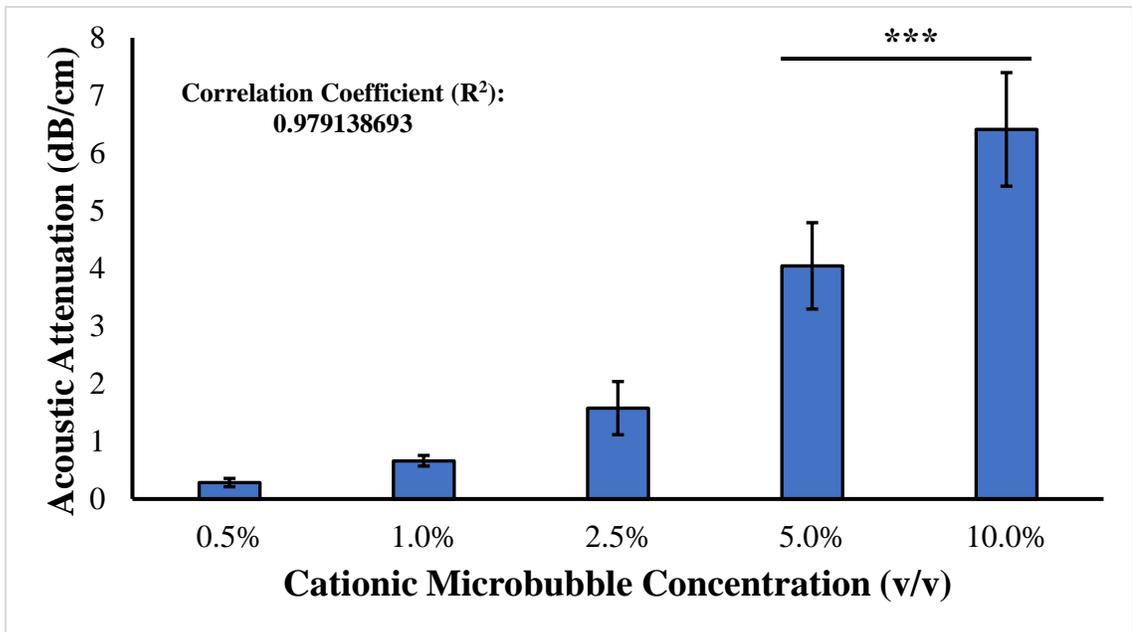
Figure 12: Acoustic attenuation coefficient measurements of cationic microbubble concentrations. Concentration groups of 5.0% and 10.0% were significantly different from the other groups at a p-value < 0.001.

The acoustic attenuation coefficients were calculated, using Equation 1, for all the red blood cells and Jurkat T-cell concentrations, shown in Figures 13 and 14 [13]. The difference in acoustic attenuation is easily distinguishable for the red blood cell concentrations, however that is not the case for the Jurkat T-cell concentrations.
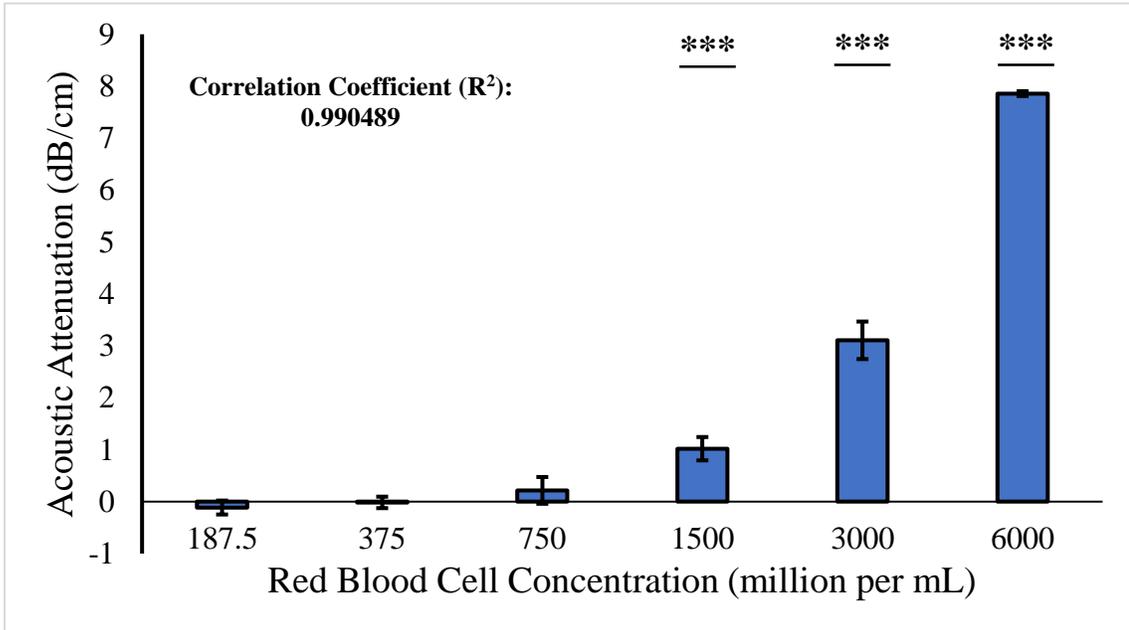
Figure 13: Acoustic attenuation coefficient measurements of red blood cell concentrations. Concentration groups of 6000, 3000, and 1500 million (6 billion, 3 billion, and 1.5 billion) cells per mL were all individually significantly different at a p-value < 0.001.
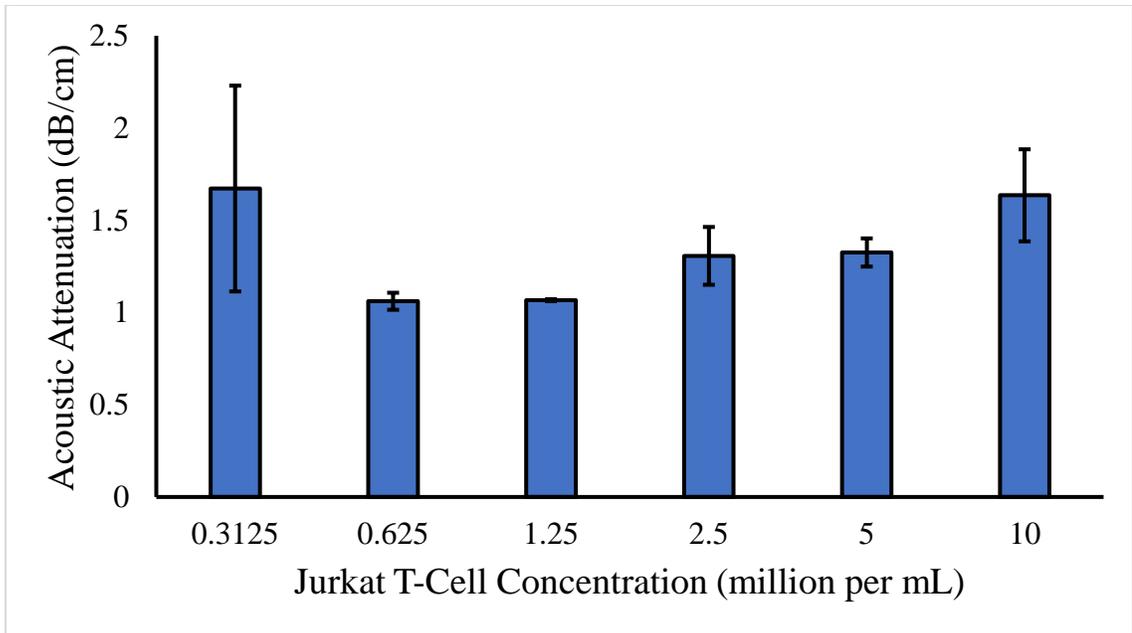
Figure 14: Acoustic attenuation coefficient measurements of Jurkat T-cell concentrations. No statistically significant differences were observed when comparing all six concentrations between 0.3125 million (312.5 thousand) and 10 million cells per mL.

The three highest red blood cell concentrations are significantly different from the lower concentrations, and the differences between these concentrations can be visibly seen when looking at their attenuation coefficients. When looking at all six concentrations of the Jurkat T-cells, no groups are significant, and the correlation coefficient between the concentrations is 0.22. The lowest Jurkat T-cell concentration gives an abnormally large attenuation coefficient value, so it was determined that analysis with the five other concentrations, excluding the 0.3125 million (312.5 thousand) cells per mL, should be investigated. With this exclusion, the 10 million cells per mL concentration becomes significantly different, at a p-value of 0.040, from the 0.625 million (625 thousand) cells per mL, and the data set has a correlation coefficient of 0.92.

Both transfer and scratch learning models were programmed to split the data set being analyzed into training, external validation, and testing sets at an 80-to-10-to-10 split. They were also programmed to display a Training Progress window to show the training and validation accuracy, as well as loss while the models were trained. In addition to these plots, confusion charts were generated, and their metrics calculated, for all datasets in order to determine which predictions from the models were false. These data sets were analyzed using simulated acoustic attenuation data with polystyrene microspheres, as well as experimental acoustic attenuation data acquired with red blood cells and Jurkat T-cells.

1. Theoretical Microparticle Sizes

The outputs of the transfer learning and scratch learning models that were trained on the theoretical microparticle sizes are shown in Figures 15 and 17, respectfully. Confusion charts and their metrics for each model type are shown in Figures 16 and 18, as well as Tables 3 and 4.

Figure 15: Transfer learning model training progress for theoretical data spectrograms of varying microparticle sizes. The validation accuracy was 33.33%.
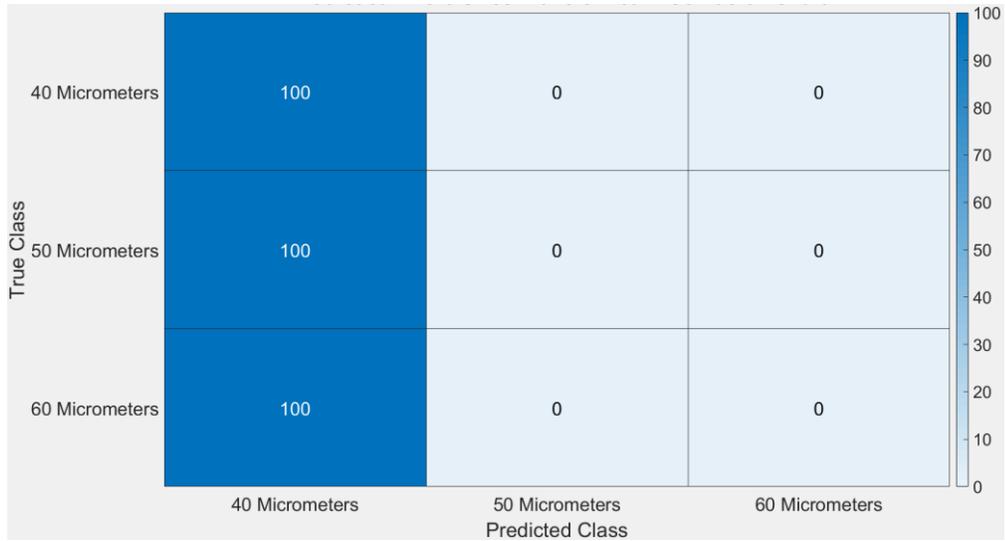
Figure 16: Confusion chart for transfer learning model training progress on theoretical data spectrograms of varying microparticle sizes.

Table 3: Confusion chart metrics for transfer learning model on theoretical data spectrograms of varying microparticle sizes.

	Sensitivity	Specificity	Precision
40 Micrometers	1	0	0.33333
50 Micrometers	0	1	NaN
60 Micrometers	0	1	NaN
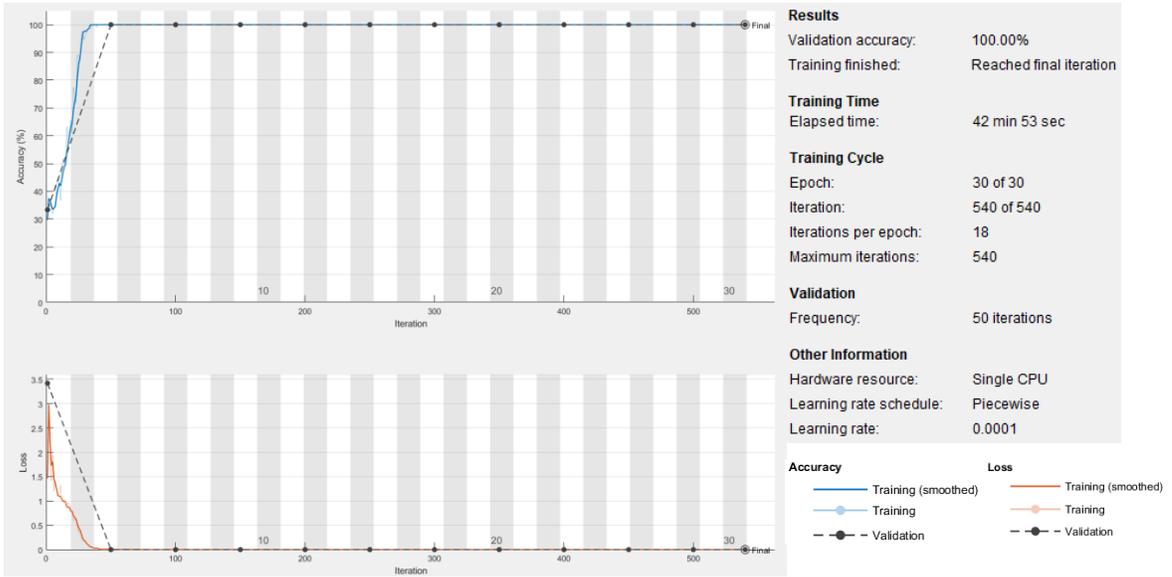
Figure 17: Scratch learning model training progress for theoretical data spectrograms of varying microparticle sizes. The validation accuracy was 100%.
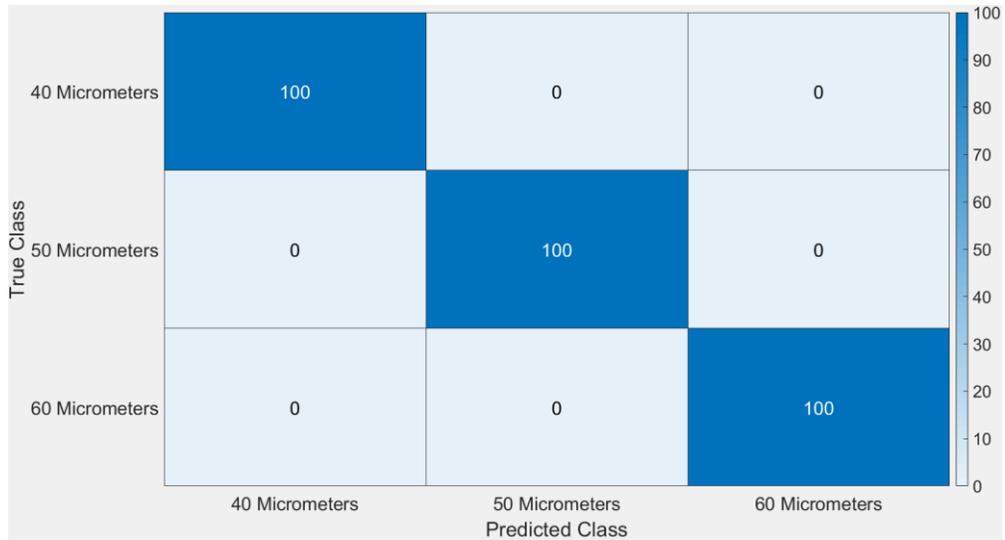
Figure 18: Confusion chart for scratch learning model training progress on theoretical data spectrograms of varying microparticle sizes.

Table 4: Confusion chart metrics for scratch learning model on theoretical data spectrograms of varying microparticle sizes.

	Sensitivity	Specificity	Precision
40 Micrometers	1	1	1
50 Micrometers	1	1	1
60 Micrometers	1	1	1

For the models pertaining to the theoretical microparticle sizes, the transfer learning model had a validation accuracy of 33.33%, while the scratch learning model achieved a validation accuracy of 100%.

2. Experimentally Measured in Static Cuvette Setup

There are multiple differences between the characteristics of red blood cells and Jurkat T-cells. There is a major size difference between these two cell types. A typical red blood cell has a biconcave-discoid shape with a diameter of about 7 to 8 micrometers, while Jurkat T-cells have a spherical shape with a diameter ranging from 10 to 16 micrometers

[11, 22]. These differences in size and shape will cause differences in the acoustic attenuation measurements of these cell types based on their acoustic impedance. If the acoustic attenuation of both red blood cells and Jurkat T-cells were measured at equal cell concentrations, the Jurkat T-cells would be expected to show a higher attenuation coefficient due to its larger size compared to the red blood cells. Other parameters, such as any dynamic changes or inaccuracies of the cellular concentrations, could potentially affect the acoustic attenuation measurements, as well as the machine learning analysis accuracy. This would occur due to the change in the number of cells, or the summated density of the cells, within the ultrasound field.

a. Red Blood Cell Concentrations

The outputs of the transfer learning and scratch learning models that were trained on the six red blood cell concentrations are shown in Figures 19 and 21, respectfully. Confusion charts and their metrics for each model type are shown in Figures 20 and 22, as well as Tables 5 and 6.

Figure 19: Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup. The validation accuracy was 93.11%.

Figure 20: Confusion Chart for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup.

Table 5: Confusion chart metrics for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
187.5Mil_per_mL	0.9	0.96394	0.83851
375Mil_per_mL	0.78205	0.95034	0.7673
750Mil_per_mL	0.82667	0.99157	0.95385
1.5Bil_per_mL	0.99333	0.98983	0.95513
3Bil_per_mL	1	1	1
6Bil_per_mL	1	1	1

Figure 21: Scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup. The validation accuracy was 89.00%.

Figure 22: Confusion chart for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup.

Table 6: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for red blood cell concentrations acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
187.5Mil_per_mL	0.86667	0.9721	0.87248
375Mil_per_mL	0.62	0.98174	0.87736
750Mil_per_mL	0.81333	0.94233	0.74847
1.5Bil_per_mL	0.98	0.95273	0.82123
3Bil_per_mL	1	0.99535	0.98039
6Bil_per_mL	1	1	1

For the models pertaining to the red blood cell concentrations, the transfer learning model achieved a validation accuracy of 93.11%, while the scratch learning model had a validation accuracy of 89.00%.

The outputs of the transfer learning and scratch learning models that were trained on the six Jurkat T-cell concentrations are shown in Figures 23 and 25, respectfully. Confusion charts and their metrics for each model type are shown in Figures 24 and 26, as well as Tables 7 and 8.
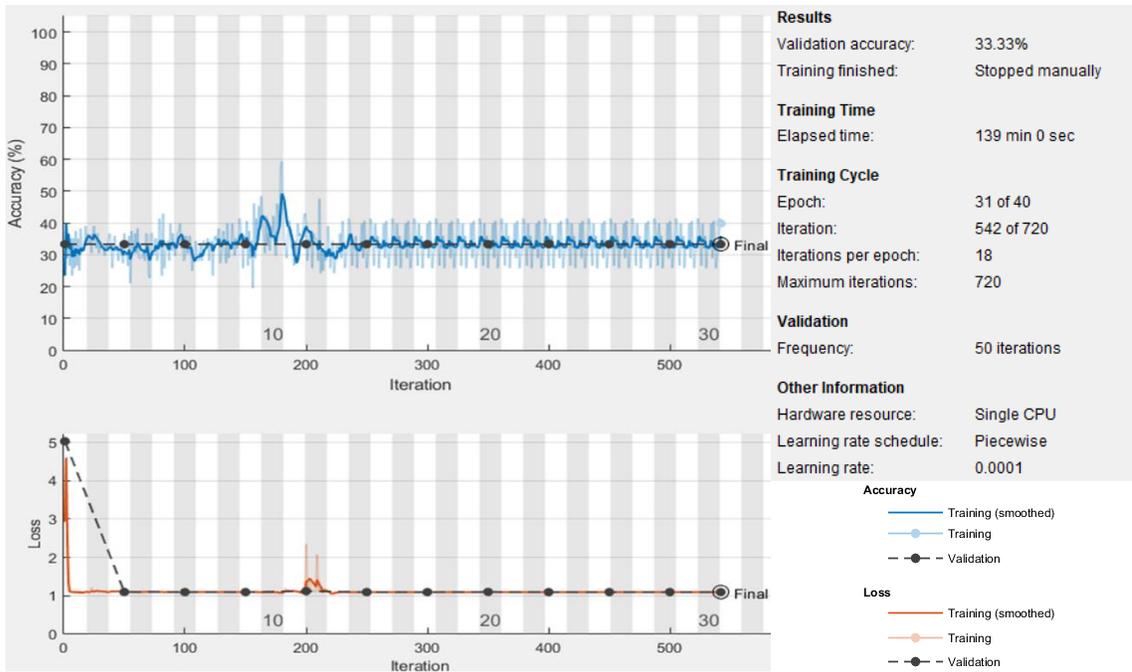

Figure 23: Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-Cell concentrations acquired in static cuvette acoustic attenuation setup. The validation accuracy was 16.67%.

Figure 24: Confusion chart for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations acquired in static cuvette acoustic attenuation setup.

Table 7: Confusion chart metrics for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
312.5k_per_mL	0	1	NaN
625k_per_mL	0	1	NaN
1.25Mil_per_mL	1	0	0.16667
2.5Mil_per_mL	0	1	NaN
5Mil_per_mL	0	1	NaN
10Mil_per_mL	0	1	NaN
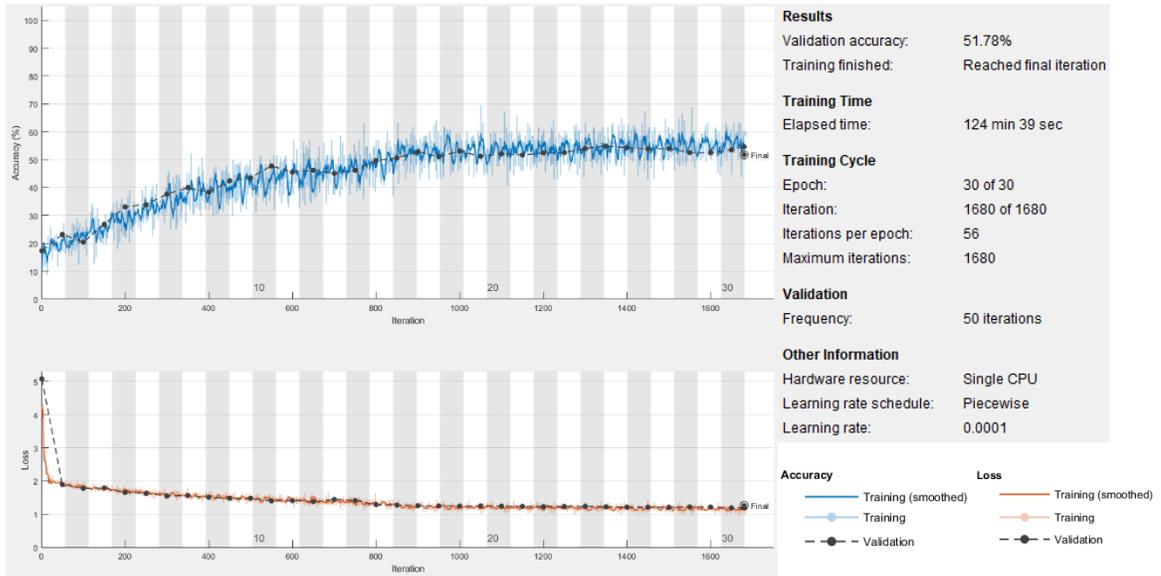
Figure 25: Scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations acquired in static cuvette acoustic attenuation setup. The validation accuracy was 51.78%.
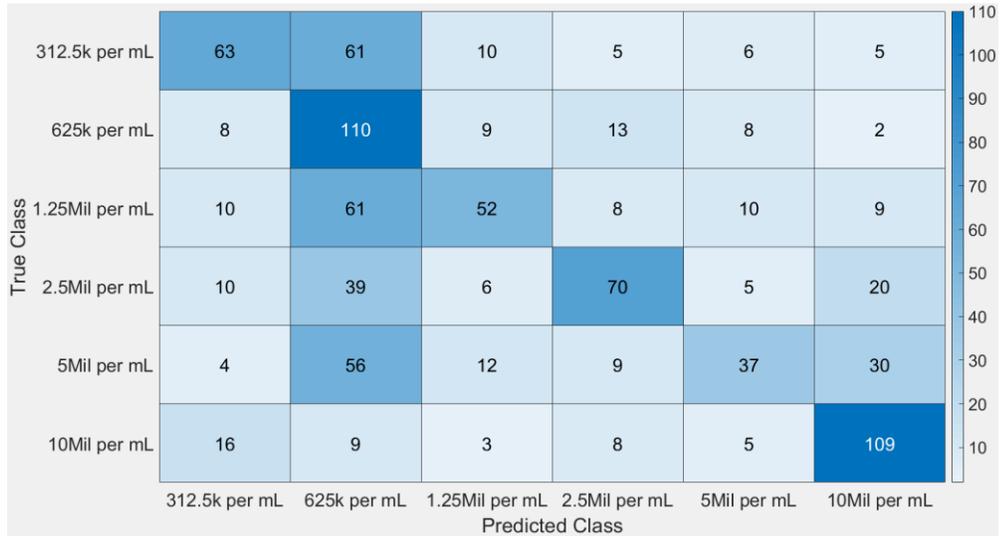
Figure 26: Confusion chart for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations acquired in static cuvette acoustic attenuation setup.

Table 8: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
312.5k_per_mL	0.42	0.88732	0.56757
625k_per_mL	0.73333	0.59425	0.32738
1.25Mil_per_mL	0.34667	0.90676	0.56522
2.5Mil_per_mL	0.46667	0.89614	0.61947
5Mil_per_mL	0.25	0.92237	0.52113
10Mil_per_mL	0.72667	0.83417	0.62286

For the models pertaining to the six Jurkat T-cell concentrations, the transfer learning model had a validation accuracy of 16.67%, while the scratch learning model achieved a validation accuracy of 51.78%.

In the attenuation coefficient measures, it was shown that when excluding the 0.3125 million (312.5 thousand) Jurkat T-cells per mL concentration as an outlier, the 10 million cells per mL concentration became significantly different from the 0.625 million (625 thousand) cells per mL concentration at a p-value of 0.040. Because of this, the transfer and scratch learning models were retrained for only the five Jurkat T-cell concentrations (0.625 to 10 million cells per mL) to see if the accuracy and loss would improve without the outlier. The outputs of the transfer learning and scratch learning models that were trained on these five Jurkat T-cell concentrations are shown in Figures 27 and 29, respectfully. Confusion charts and their metrics for each model type are shown in Figures 28 and 30, as well as Tables 9 and 10.
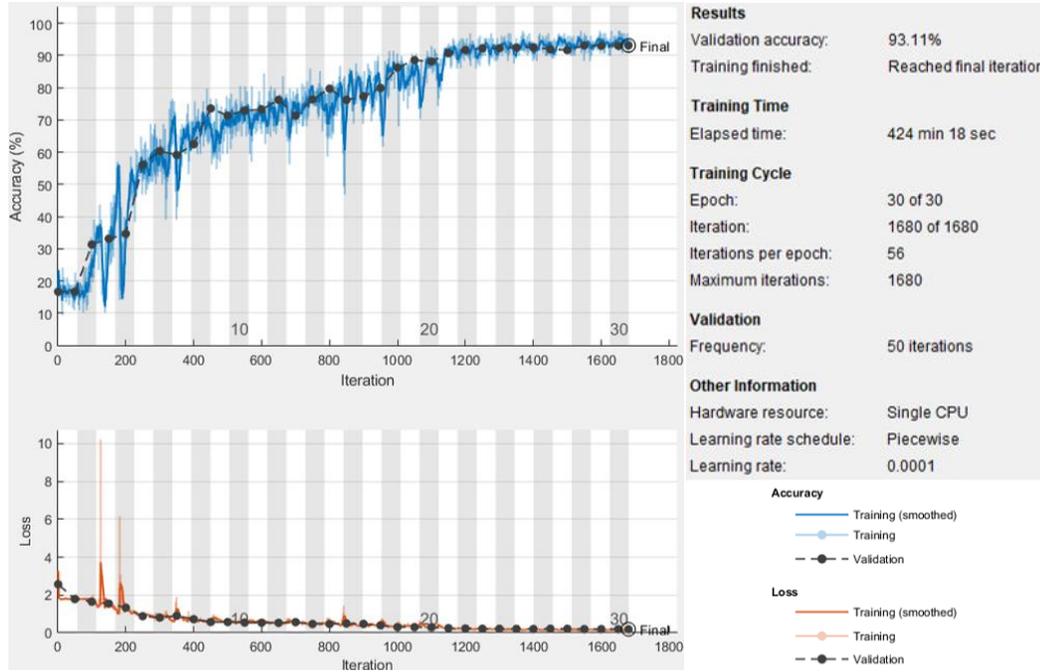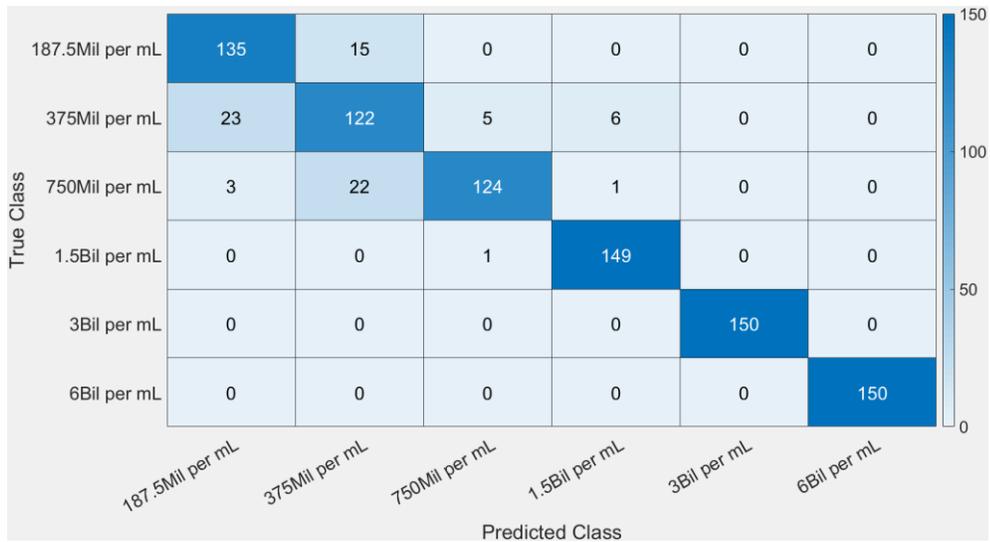
Figure 27: Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup. The validation accuracy was 20.00%.

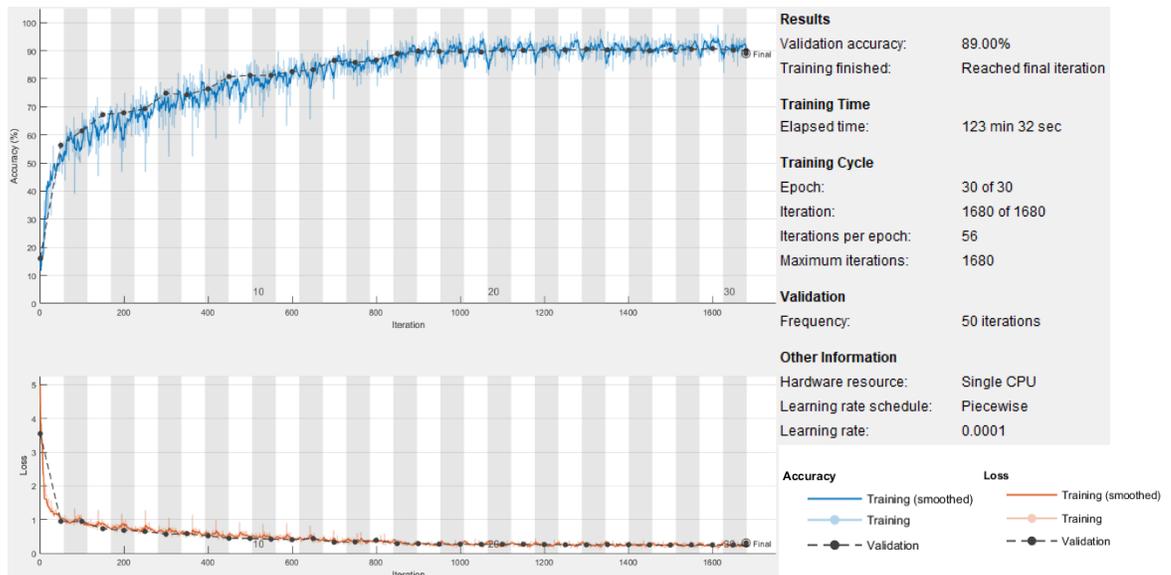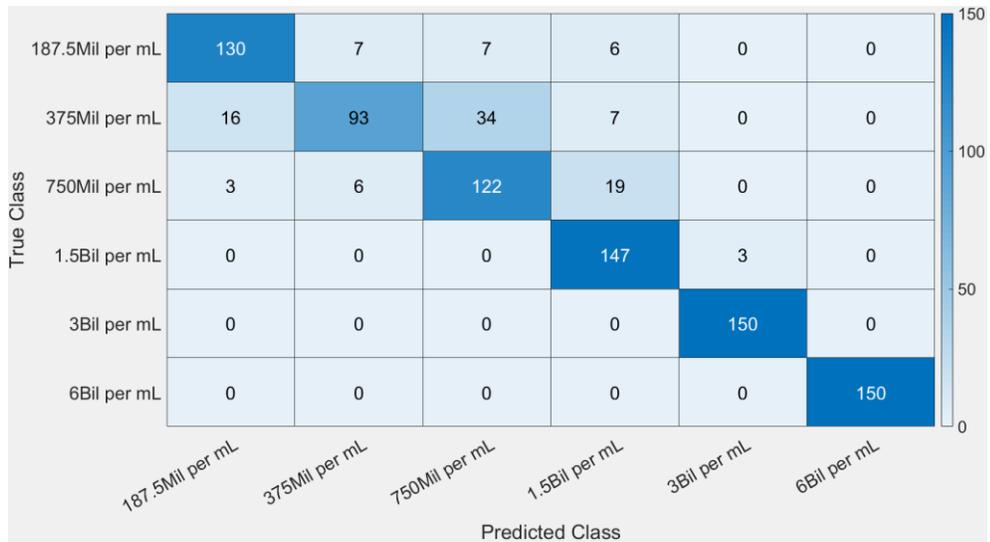Figure 28: Confusion chart for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup.

Table 9: Confusion chart metrics for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
625k_per_mL	0	1	NaN
1.25Mil_per_mL	1	0	0.2
2.5Mil_per_mL	0	1	NaN
5Mil_per_mL	0	1	NaN
10Mil_per_mL	0	1	NaN
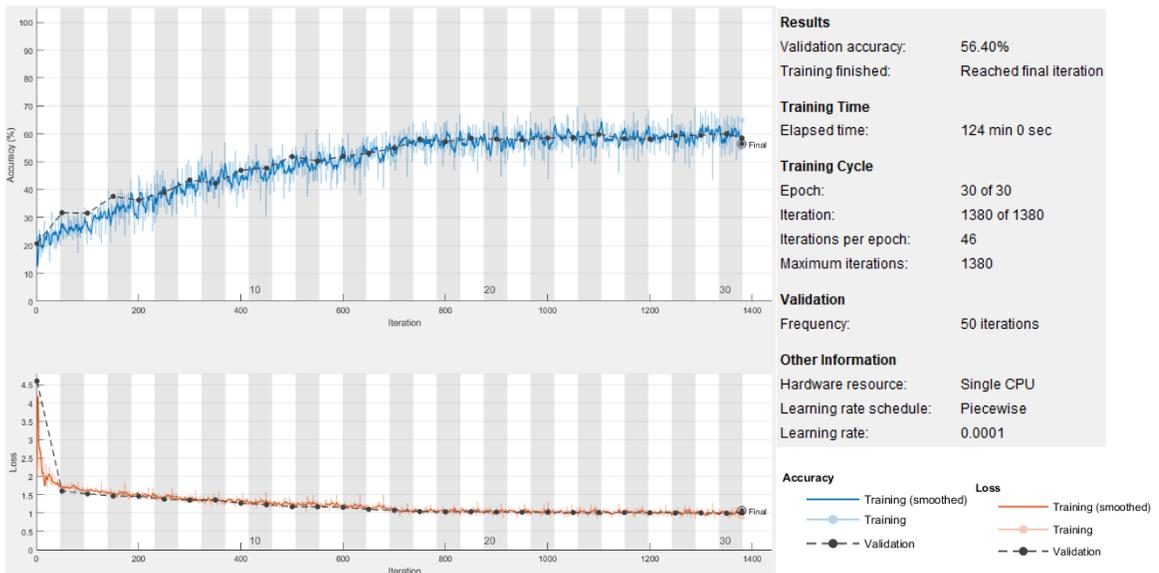
Figure 29: Scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup. The validation accuracy was 56.40%.

Figure 30: Confusion chart for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup.

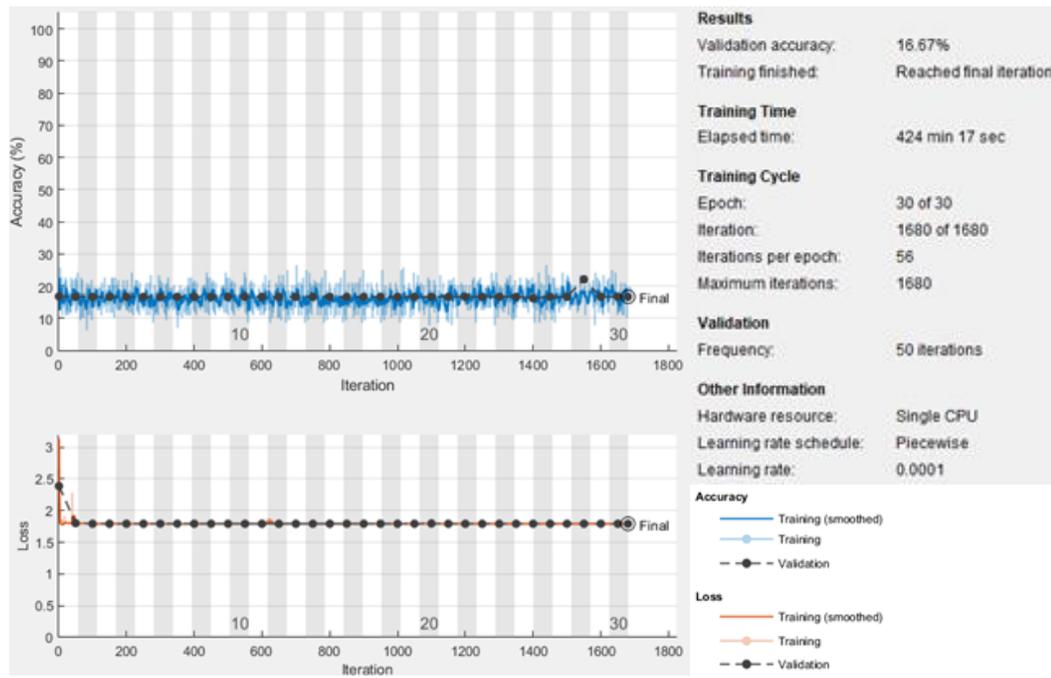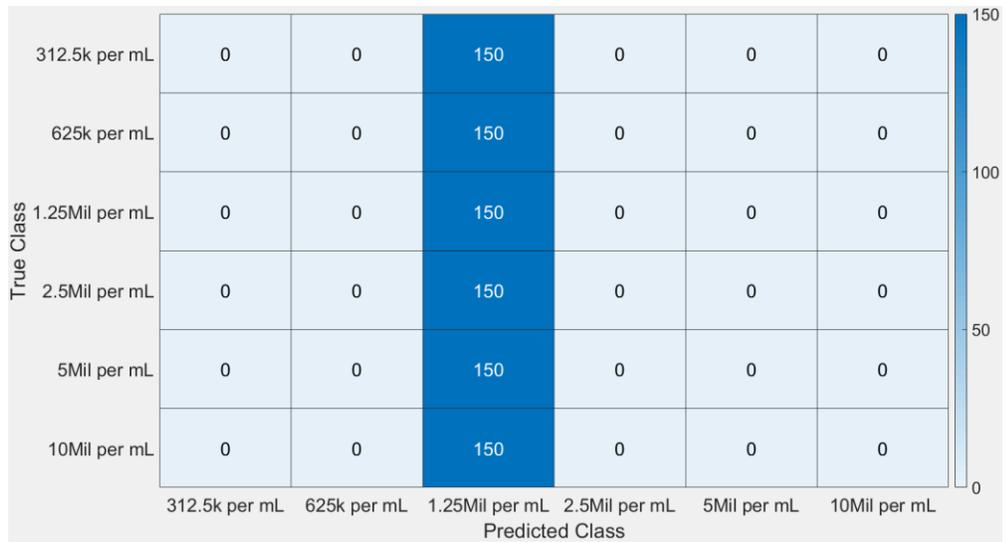Table 10: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL as an outlier) acquired in static cuvette acoustic attenuation setup.

	Sensitivity	Specificity	Precision
625k_per_mL	0.40667	0.89664	0.60396
1.25Mil_per_mL	0.41333	0.7954	0.4106
2.5Mil_per_mL	0.66667	0.81915	0.59524
5Mil_per_mL	0.58667	0.7191	0.41315
10Mil_per_mL	0.64667	0.93958	0.82906

For the models pertaining to the remaining five Jurkat T-cell concentrations, the transfer learning model had a validation accuracy of 20.00%, while the scratch learning model achieved a validation accuracy of 56.40%.

3. Acoustofluidic Attenuation Setup

The steps taken in this research were to be a proof of concept for utilizing machine learning for analysis of acoustic attenuation measurements for cell characterization that would be eventually used in an in-line acoustofluidic quality control module for cell processing of cancer immunotherapies. An acoustofluidic attenuation setup, shown in Figure 31, was developed at the University of Louisville Theranostic Ultrasound Laboratory that uses the same basic components as the static cuvette acoustic attenuation setup.

Figure 31: Rendering of acoustofluidic attenuation device as in SolidWorks (left) and two angles of 3D printed device with attached transducers and tubing (middle and right).

Just like the static cuvette device, the acoustofluidic attenuation device utilizes a transmitting ultrasound transducer driven by a function generator producing the same frequency sweep that was used in the static setup, and a receiving ultrasound transducer connected to an oscilloscope for attenuation waveform acquisition. The acoustofluidic device utilizes a peristaltic pump to achieve a continuous flow of the sample being measured to imitate a future in-line model and ensure the solution stays well mixed. The

solution fills up the attenuation chamber enclosed in the device. The path length of ultrasound waveforms through the solution in this chamber is the same as that in the static attenuation device (10mm), but with the cylindrical shape of the acoustofluidic chamber and the circular shape of the transducers, a larger volume of cells is contained within the ultrasound field, potentially increasing the signal-to-noise ratio to improve the accuracy of acoustic attenuation measurements.

Some experiments with this acoustofluidic attenuation device design were run using Jurkat T-Cell concentrations, since it was determined by the trials conducted with the static cuvette acoustic attenuation setup that the Jurkat T-cell concentrations were more difficult to distinguish. The acoustic waveform data collected with this device were converted into spectrograms using the same procedure as the signals acquired with the static setup, and those spectrograms were used for training and validation of transfer and scratch learning models. There was a total of 6,000 spectrograms generated (1,000 for each concentration) for the Jurkat T-cells.

a. Experimentally Measured Jurkat T-Cell Concentrations

The outputs of the transfer learning and scratch learning models that were trained on all six Jurkat T-cell concentrations measured in the acoustofluidic attenuation setup are shown in Figures 32 and 34, respectfully. Confusion charts and their metrics for each model type are shown in Figures 33 and 35, as well as Tables 11 and 12.
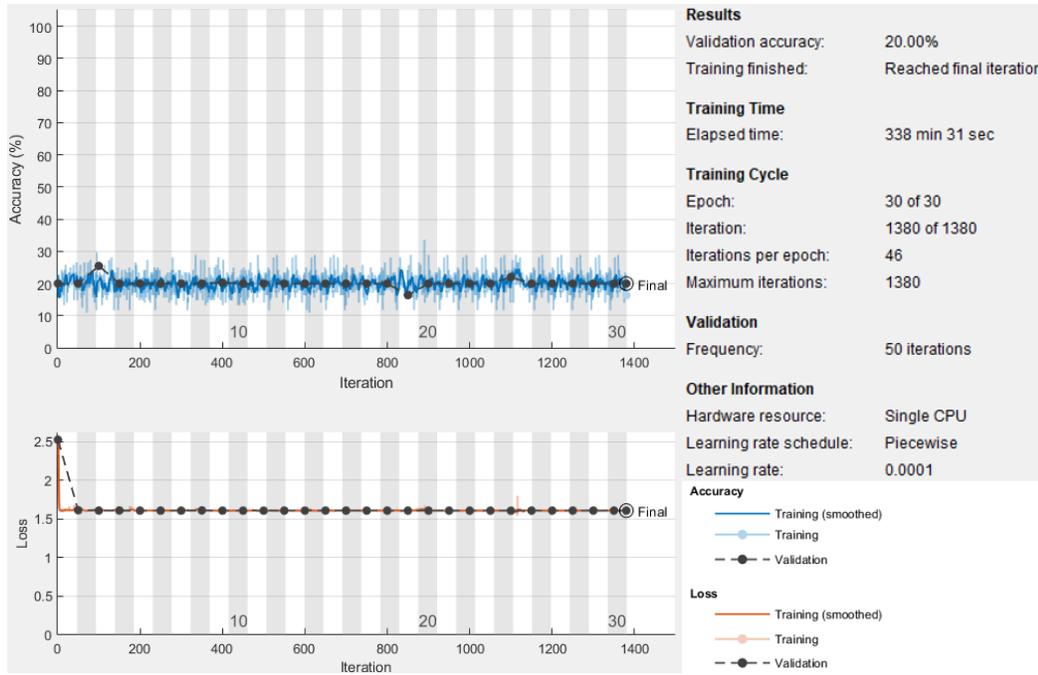
Figure 32: Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup. The validation accuracy was 16.67%.
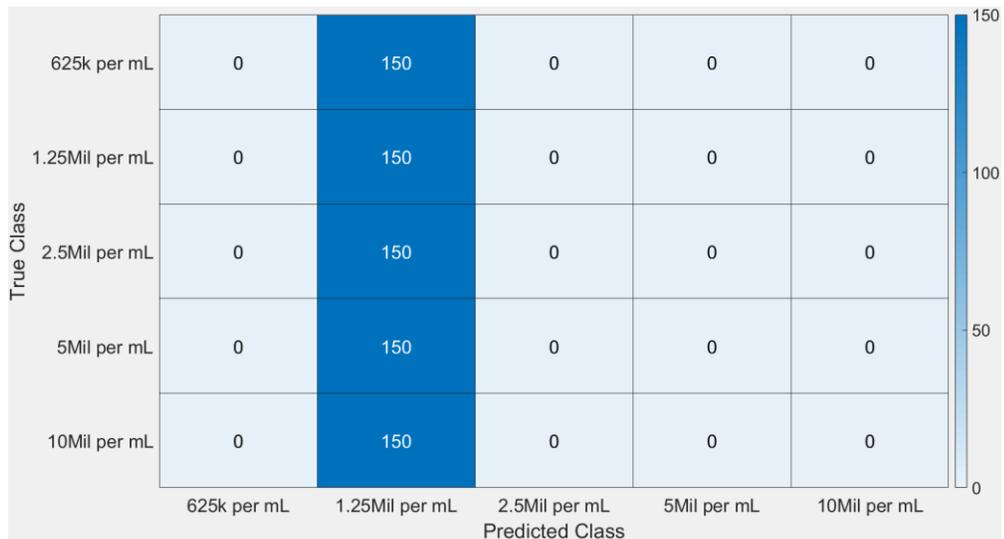

Figure 33: Confusion chart for Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup.

Table 11: Confusion chart metrics for Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup.

|	Sensitivity	Specificity	Precision
312.5k_per_mL	0	1	NaN
625k_per_mL	0	1	NaN
1.25Mil_per_mL	0	1	NaN
2.5Mil_per_mL	1	0	0.16667
5Mil_per_mL	0	1	NaN
10Mil_per_mL	0	1	NaN

Figure 34: Scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup. The validation accuracy was 56.50%.
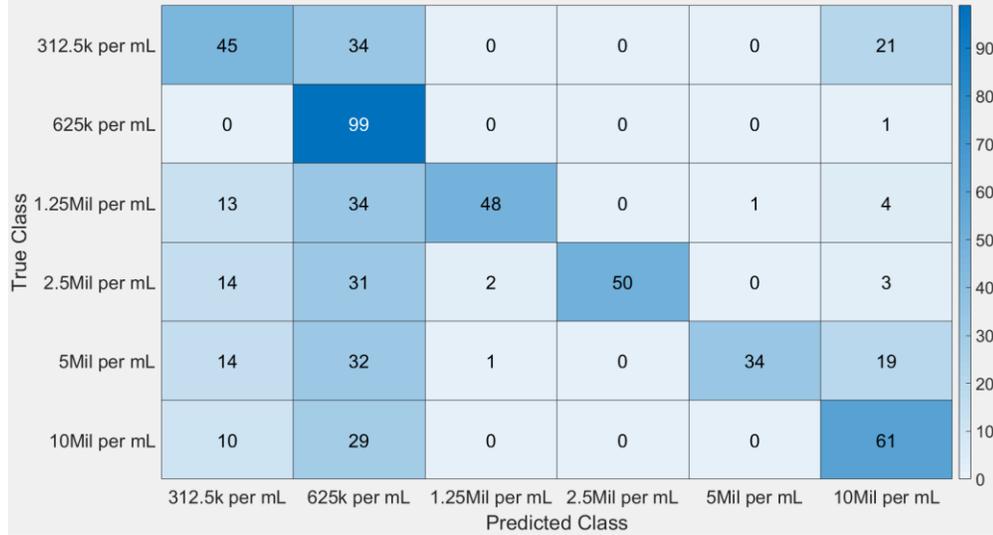
Figure 35: Confusion chart for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup.

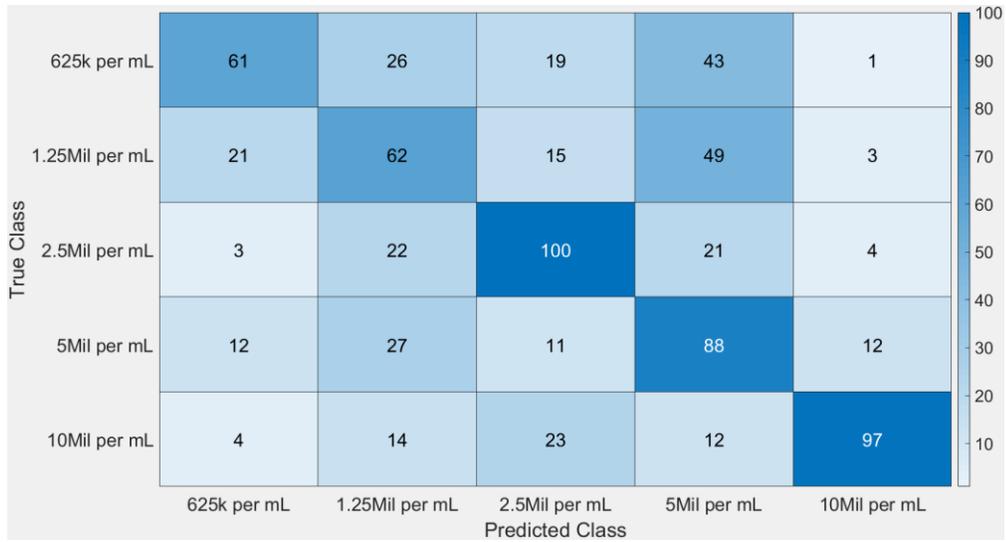Table 12: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for all Jurkat T-cell concentrations acquired in acoustofluidic attenuation setup.

	Sensitivity	Specificity	Precision
312.5k_per_mL	0.45	0.85131	0.46875
625k_per_mL	0.99	0.59799	0.38224
1.25Mil_per_mL	0.48	0.98973	0.94118
2.5Mil_per_mL	0.5	1	1
5Mil_per_mL	0.34	0.99671	0.97143
10Mil_per_mL	0.61	0.85185	0.55963

For the models pertaining to the six Jurkat T-cell concentrations measured in the acoustofluidic attenuation setup, the transfer learning model had a validation accuracy of 16.67%, while the scratch learning model achieved a validation accuracy of 56.50%.

Since the Jurkat T-cell concentrations measured in the static cuvette setup were analyzed a second time with the machine learning models, while excluding the 0.3125

million (312.5 thousand) Jurkat T-cells per mL concentration as an outlier, the same was done for the measurements acquired using the acoustofluidic attenuation setup.
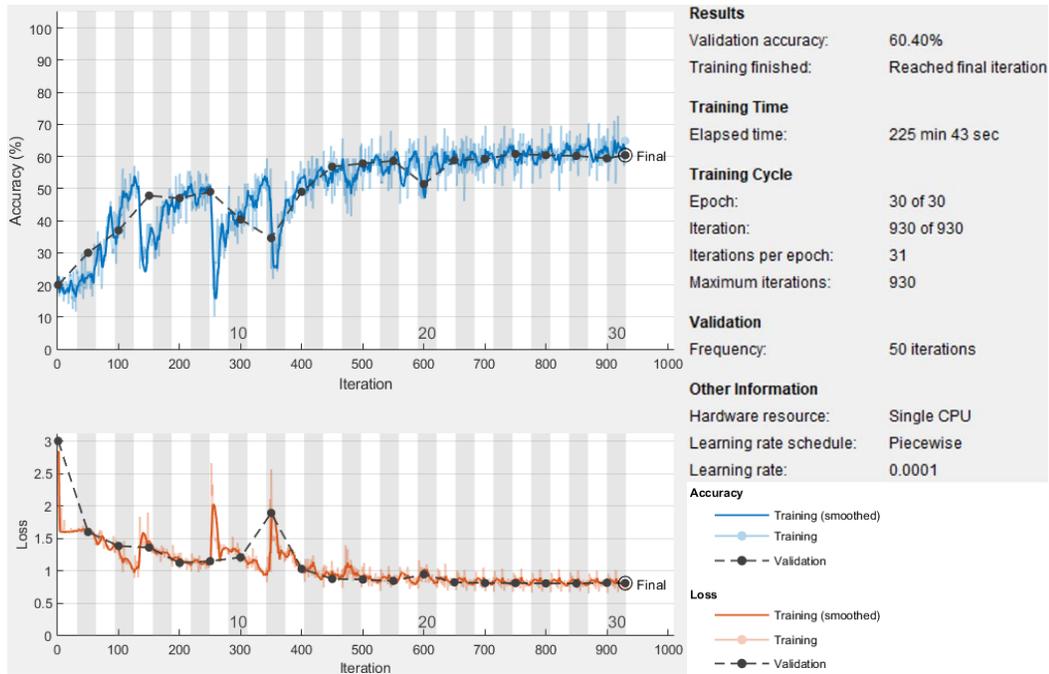
Figure 36: Transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup. The validation accuracy was 60.40%.
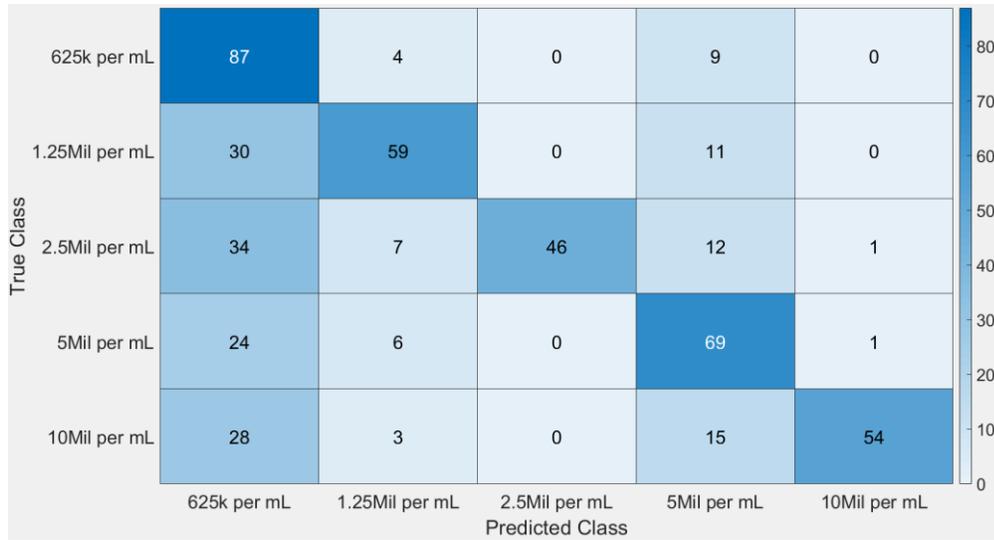
Figure 37: Confusion chart for transfer learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup.

Table 13: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup.

	Sensitivity	Specificity	Precision
625k_per_mL	0.87	0.66279	0.42857
1.25Mil_per_mL	0.59	0.92754	0.74684
2.5Mil_per_mL	0.46	1	1
5Mil_per_mL	0.69	0.83959	0.59483
10Mil_per_mL	0.54	0.9924	0.96429

40
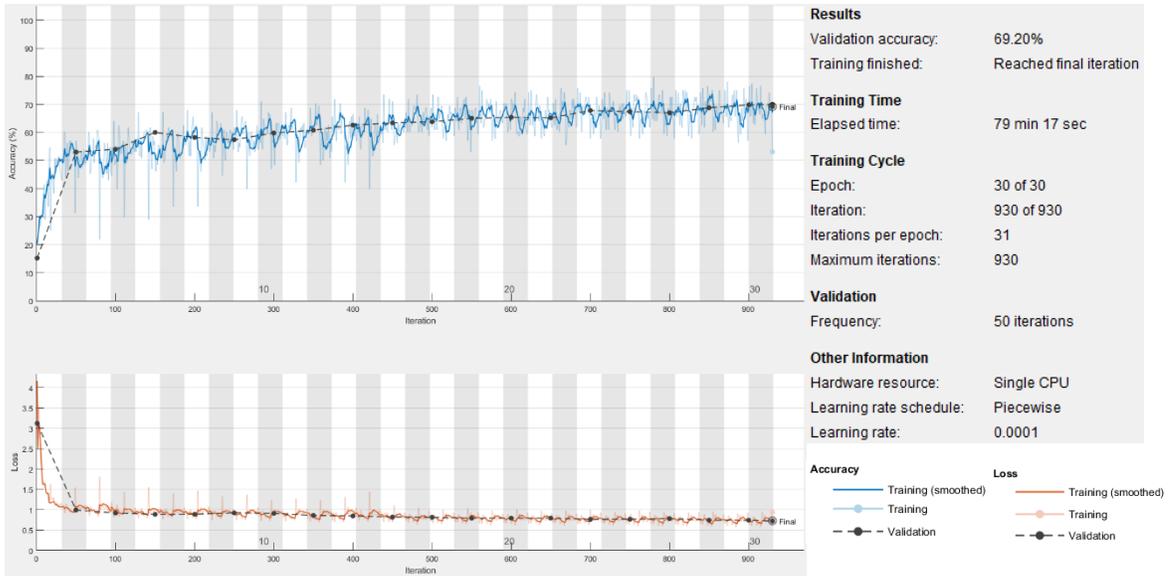
Figure 38: Scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup. The validation accuracy was 69.20%.
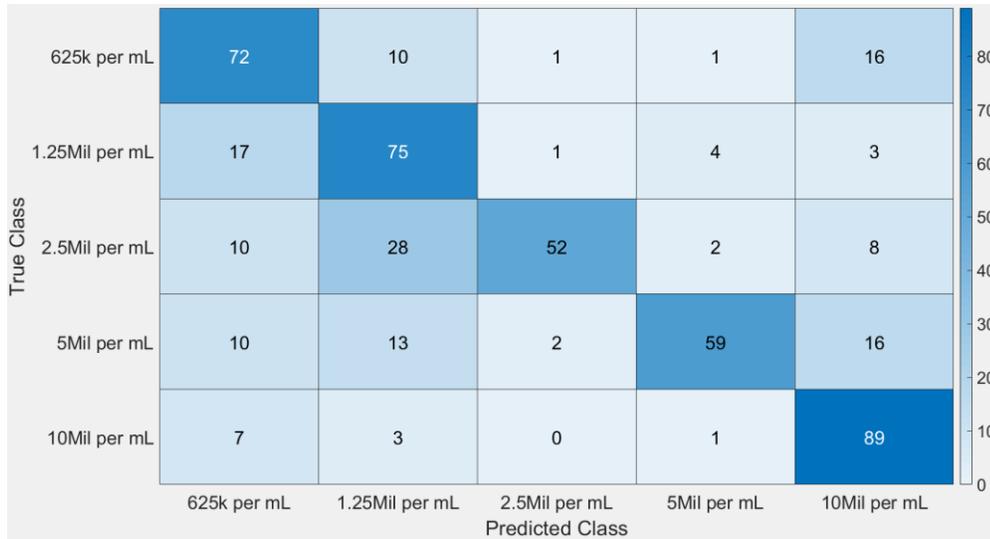
Figure 39: Confusion chart for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup.
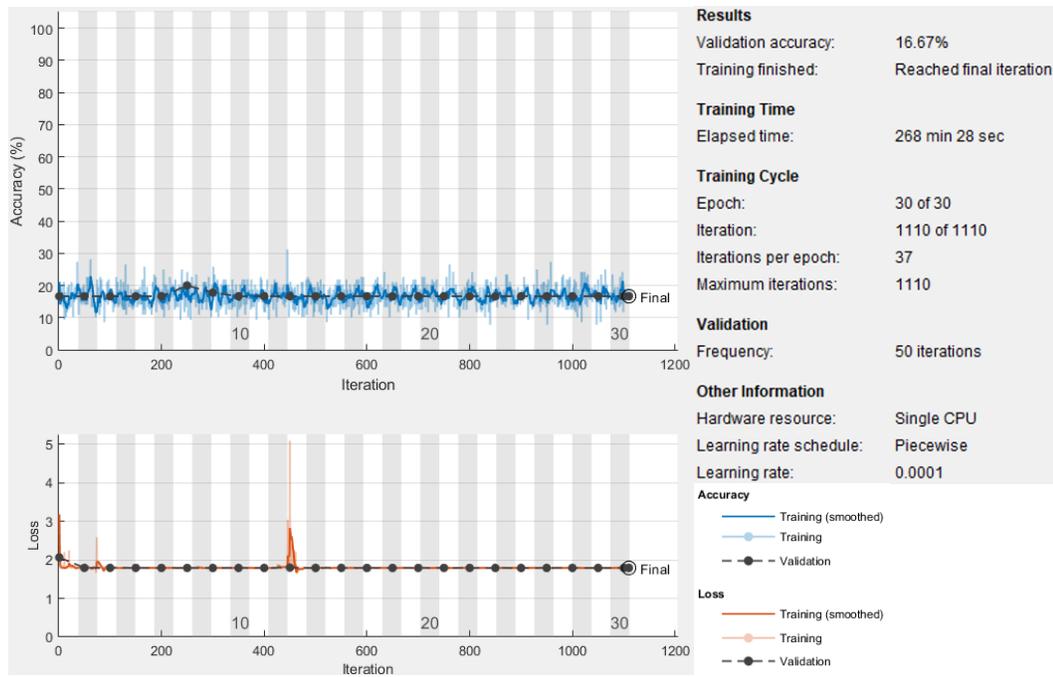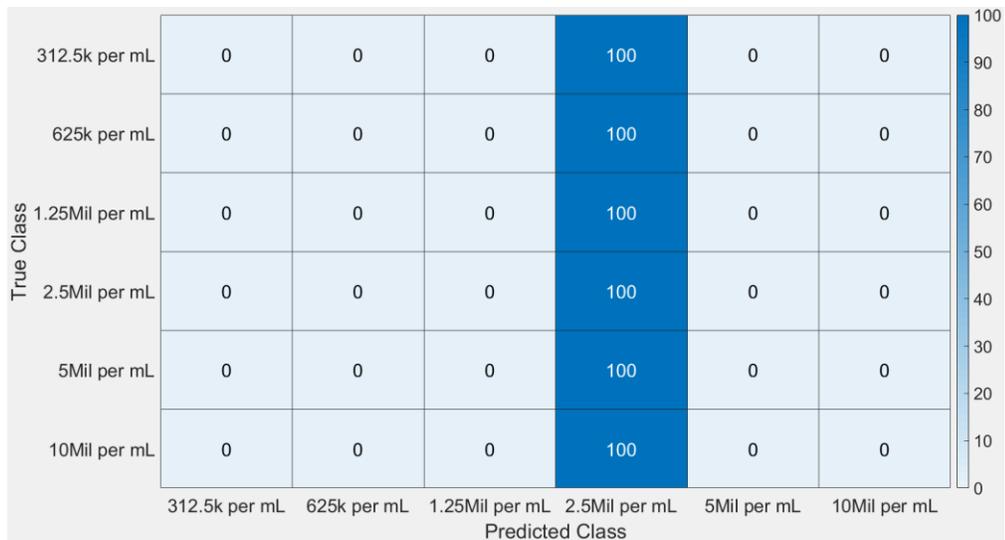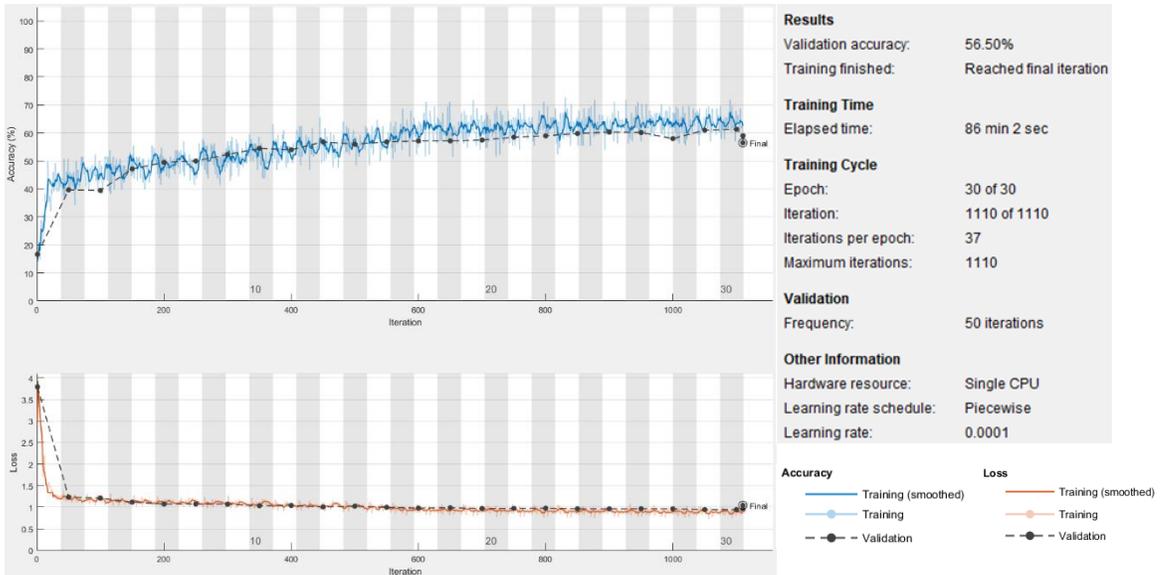
Table 14: Confusion chart metrics for scratch learning model training progress for spectrograms of experimentally measured acoustic attenuation data for Jurkat T-cell concentrations (omitting 0.3125 million, or 312.5 thousand, cells per mL) acquired in acoustofluidic attenuation setup.

	Sensitivity	Specificity	Precision
625k_per_mL	0.72	0.86207	0.62069
1.25Mil_per_mL	0.75	0.83436	0.5814
2.5Mil_per_mL	0.52	0.98662	0.92857
5Mil_per_mL	0.59	0.97297	0.8806
10Mil_per_mL	0.89	0.85714	0.67424

For the models pertaining to the remaining five Jurkat T-cell concentrations measured in the acoustofluidic attenuation setup, the transfer learning model had a validation accuracy of 60.40%, while the scratch learning model achieved a validation accuracy of 69.20%.

In terms of accuracy, transfer learning was more successful in predicting the correct concentrations of red blood cells; however, scratch learning was more successful in predicting the sizes of the theoretical microparticles and the concentrations of the Jurkat T-cells. It is believed that transfer learning worked better for the red blood cell concentrations because those measurements had a much higher signal-to-noise ratio, as demonstrated by the higher acoustic attenuation coefficient values shown in Figure 13. Looking at the attenuation coefficient measurements for those concentrations, there is a significant difference between the red blood cell concentrations. In contrast, the attenuation coefficient measurements for the Jurkat T-cell concentrations were more difficult to distinguish. This is most likely due to the low signal-to-noise ratio. Since concentrations for the red blood cells ranged from 187.5 million to 6 billion cells per mL, that dataset would have a higher signal-to-noise ratio, while the concentrations of the red blood cells ranged from 312.5 thousand to 10 million cells per mL.

IV.   CONCLUSIONS

The research conducted for this thesis demonstrates that machine learning analysis can be applied to acoustic signals for characterization of cells and particles. In a majority of the data sets, the scratch learning model achieved a higher validation accuracy than the transfer learning model. Transfer learning only had a higher accuracy for the red blood cell concentrations, but that was only by approximately 4%. These results show that transfer learning may work for data sets containing samples with a higher signal-to-noise ratio, while scratch learning will work better for data sets containing samples with a lower signal-to-noise ratio.

The research conducted for this thesis showed that having a larger volume of cells within the ultrasound field, as was the case in the acoustofluidic attenuation setup compared to the static cuvette setup, appears to increase the signal-to-noise ratio as well as the accuracy of the machine learning analysis. This research also shows that the scratch learning machine learning analysis method works better for a continuous-flow acoustic attenuation cell characterization device compared to a transfer learning analysis method. The scratch learning model in this thesis serves as a base for more advanced models to be developed in the future. The scratch learning model used in this thesis can to be further optimized to identify other cell characteristics, but this model has provided proof of concept that machine learning methods can be used in the analysis of acoustic attenuation measurements for characterization of cell properties.

V.   RECOMMENDATIONS

Multiple aspects of this research could be improved upon in order to achieve a higher accuracy for the machine learning models. The first topic to look into would be to re-evaluate the frequency spectrum of the transducers used. From looking at the data for the theoretical microparticle sizes spectrograms, it might be beneficial to investigate an increased broadband of frequencies. The theoretical microparticle sizes spectrograms were created with a frequency sweep from 1 to 25 MHz, while the frequency sweep used in the experimentally acquired measurements was only from approximately 3.0 to 3.6 MHz based on the fundamental resonance frequency of the transducers being utilized in the devices. New transducers might need to be implemented to the acoustic attenuation devices, but once an optimized frequency sweep has been determined, that could be applied to the acoustofluidic attenuation device.

Another aspect to investigate would be the parameters and architecture of the scratch learning model. Optimization is a major topic to improve the accuracy of the machine learning model. The parameters and architecture of scratch learning model used in this research were based on a model that did not have the same end-goal as this project. There are more than twenty parameters that could be investigated to improve the accuracy and reduce training and predicting times of the scratch learning model. The architecture of the model could also be investigated, such as introducing new layers or altering current layers, could also lead to improved efficiency.

The results from this thesis have shown that, after some optimization, machine learning techniques can be used for analysis of acoustic attenuation measurements, that can be implemented in an in-line quality control module for cell processing applications.

REFERENCES

[1] Adachi, H., Kawamura, Y., Nakagawa, K., Horisaki, R., Sato, I., Yamaguchi, S., Fujiu, K., Waki, K., Noji, H., & Ota, S. (2020). Use of ghost cytometry to differentiate cells with similar gross morphologic characteristics. *Cytometry Part A*, *97*(4), 415–422. https://doi.org/10.1002/cyto.a.23989

[2] *Advantages and disadvantages of machine learning language*. DataFlair. (2021, March 8). Retrieved May 2, 2022, from https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/

[3] Borgert, R. (2021). Improving outcomes and mitigating costs associated with car T-cell therapy. *The American Journal of Managed Care*, *27*(Suppl 13). https://doi.org/10.37765/ajmc.2021.88737

[4] Brownlee, J. (2020, August 14). *A tour of machine learning algorithms*. Machine Learning Mastery. Retrieved May 17, 2022, from https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

[5] Callahan, G. (2021, March 9). *What are the advantages of artificial intelligence?* Rev. Retrieved May 2, 2022, from https://www.rev.com/blog/what-are-the-advantages-of-artificial-intelligence

[6] Centner, C. S., Moore, J. T., Baxter, M. E., Long, Z. T., Miller, J. M., Kovatsenko, E. S., Xie, B., Menze, M. A., Berson, R. E., Bates, P. J., Yaddanapudi, K., & Kopechek, J. A. (2021). ACOUSTOFLUIDIC-mediated molecular delivery to human T cells with a three-dimensional-printed flow chamber. *The Journal of the Acoustical Society of America*, *150*(6), 4534–4547. https://doi.org/10.1121/10.0009054

[7] Dertat, A. (2017, November 13). *Applied deep learning - part 4: Convolutional Neural Networks*. Medium. Retrieved May 17, 2022, from https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#:~:text=The%20main%20advantage%20of%20CNN,CNN%20is%20also%20computationally%20efficient.

[8] Eyles, J. E., Vessillier, S., Jones, A., Stacey, G., Schneider, C. K., & Price, J. (2018). Cell therapy products: Focus on issues with manufacturing and quality control of chimeric antigen receptor t-cell therapies. *Journal of Chemical Technology & Biotechnology*, *94*(4), 1008–1016. https://doi.org/10.1002/jctb.5829

[9] Feins, S., Kong, W., Williams, E. F., Milone, M. C., & Fraietta, J. A. (2019). An introduction to chimeric antigen receptor (CAR) T-Cell Immunotherapy for

Human Cancer. *American Journal of Hematology*, *94*(S1).
https://doi.org/10.1002/ajh.25418

[10] Gebo, J. E., & Lau, A. F. (2020). Sterility testing for cellular therapies: What is the role of the Clinical Microbiology Laboratory? *Journal of Clinical Microbiology*, *58*(7). https://doi.org/10.1128/jcm.01492-19

[11] Kinnunen, M., Kauppila, A., Karmenyan, A., & Myllylä, R. (2011). Effect of the size and shape of a red blood cell on elastic light scattering properties at the single-cell level. *Biomedical Optics Express*, *2*(7), 1803.
https://doi.org/10.1364/boe.2.001803

[12] Kinsler, L. E., & Frey, A. R. (1962). *Fundamentals of Acoustics* (2nd ed.). John Wiley & Sons, Inc.

[13] Kopechek, J. A., Haworth, K. J., Raymond, J. L., Douglas Mast, T., Perrin, S. R., Klegerman, M. E., Huang, S., Porter, T. M., McPherson, D. D., & Holland, C. K. (2011). Acoustic characterization of echogenic liposomes: Frequency-dependent attenuation and Backscatter. *The Journal of the Acoustical Society of America*, *130*(5), 3472–3481. https://doi.org/10.1121/1.3626124

[14] Lee, J. H., Shin, J., & Realff, M. J. (2018). Machine learning: Overview of the recent progresses and implications for the Process Systems Engineering Field. *Computers & Chemical Engineering*, *114*, 111–121.
https://doi.org/10.1016/j.compchemeng.2017.10.008

[15] Ma, C., Zhang, Z., Luce, B., Pusateri, S., Xie, B., Rafiei, M. H., & Hu, N. (2020). Accelerated design and characterization of non-uniform cellular materials via a machine-learning based framework. *Npj Computational Materials*, *6*(1).
https://doi.org/10.1038/s41524-020-0309-6

[16] MathWorks. (n.d.). *Deep Learning with MATLAB*. MathWorks Training Services. Retrieved from https://matlabacademy.mathworks.com/details/deep-learning-with-matlab/mldl

[17] Mullard, A. (2021). FDA approves Fourth Car-T cell therapy. *Nature Reviews Drug Discovery*, *20*(3), 166–166. https://doi.org/10.1038/d41573-021-00031-9

[18] Ota, S., Sato, I., & Horisaki, R. (2020). Implementing machine learning methods for imaging flow cytometry. *Microscopy*, *69*(2), 61–68.
https://doi.org/10.1093/jmicro/dfaa005

[19] Santomasso, B., Bachier, C., Westin, J., Rezvani, K., & Shpall, E. J. (2019). The other side of car T-cell therapy: Cytokine release syndrome, neurologic toxicity, and financial burden. *American Society of Clinical Oncology Educational Book*, (39), 433–444. https://doi.org/10.1200/edbk_238691

[20] Tully, S., Feng, Z., Grindrod, K., McFarlane, T., Chan, K. K. W., & Wong, W. W. L. (2019). Impact of increasing wait times on overall mortality of chimeric antigen receptor T-cell therapy in large B-cell lymphoma: A discrete event simulation model. *JCO Clinical Cancer Informatics*, (3), 1–9. https://doi.org/10.1200/cci.19.00086

[21] Weiss, K., Khoshgoftaar, T. M., & Wang, D. D. (2016). A survey of Transfer Learning. *Journal of Big Data*, *3*(1). https://doi.org/10.1186/s40537-016-0043-6

[22] Yang, T., Peng, J., Shu, Z., Sekar, P. K., Li, S., & Gao, D. (2019). Determination of the membrane transport properties of Jurkat cells with a microfluidic device. *Micromachines*, *10*(12), 832. https://doi.org/10.3390/mi10120832

Acoustic Attenuation Data Acquisition Using Arbitrary Waveform Code:

```matlab
% Program used to collect attenuation data
% Made to work with function generator, oscilloscope, and attenuation
devices
% Last Updated by John Moore on 26 February 2021
% This program is set to use the cuvette attenuation system set to 3.3
MHz with PBS as reference
%**Will need to change CHAN1:SCALe and CHAN1:RANGe to optimize for
other frequencies and reference types**

%----- DS1202 Z-E Oscilloscope Data Collection Program -----
clear; clc; close all;
warning('off','instrument:fread:unsuccessfulRead');


groups = inputdlg('How many samples are you running?');
groups = str2num(groups{1});


ntimepoints = inputdlg('How long do you want to read the signals for
each sample (in minutes):');
ntimepoints = str2double(ntimepoints);


refs = 10;
refAmount = questdlg(sprintf('Would you like to run %d reference
measurements?',refs),'Reference Measurement Amount','YES','NO','NO');
switch refAmount
    case 'NO'
        input = inputdlg('Enter number of reference measurements you''d
like to conduct');
        refs = cellfun(@str2num,input);
end

%-- Find a VISA-USB object --
obj1 = instrfind('Type', 'visa-usb', 'RsrcName',
'USB0::0x1AB1::0x0517::DS1ZE214503265::0::INSTR', 'Tag','');
%%Find oscilloscope - Instrument Address Infor needs to be updated for
each specific instrument

obj2 = instrfind('Type', 'visa-usb', 'RsrcName',
'USB0::0x1AB1::0x0643::DG8A213101492::0::INSTR', 'Tag', '');
%%Find function generator - Instrument Address Infor needs to be
updated for each specific instrument

%-- Create the VISA-USB object if it does not exist --
%%otherwise use the object that was found
if isempty(obj1)
    obj1 = visa('NI',
'USB0::0x1AB1::0x0517::DS1ZE214503265::0::INSTR');
else
    fclose(obj1);
    obj1 = obj1(1);
```

```matlab
        end

    if isempty(obj2)
        obj2 = visa('NI', 'USB0::0x1AB1::0x0643::DG8A213101492::0::INSTR');
    else
        fclose(obj2);
        obj2 = obj2(1);
    end

    %-- Preset obj1 settings (before fopen)--

    obj1.InputBufferSize = 1280; %Sets input buffer size

    %-- Connect to instrument object, obj1 and obj2 --
    fopen(obj1); fopen(obj2);

    %-- Set funciton generator parameters
    cycles = 500;
    burst = questdlg('Would you like Burst on or off? (Off =
Continuous)','','ON','OFF','OFF');
    switch burst
        case 'ON'
            fcngen_parameters.frequency = 3300000;
            fcngen_parameters.Vpp = 20;
            fcngen_parameters.cycles = 500;
            prevParameters = questdlg(sprintf('Are these function generator
parameters correct? \nFreq = %d MHz \nVpp = %d Volts \nCycles =
%d',(fcngen_parameters.frequency/1e6),fcngen_parameters.Vpp,fcngen_para
meters.cycles),'Function Generator Parameters','YES','NO','NO');

            switch prevParameters
                case 'NO'
                    input = inputdlg('Enter frequency (MHz)');
                    fcngen_parameters.frequency = cellfun(@str2num,input) *
1e6;
                    fcngen_parameters.Vpp = inputdlg('Enter Vpp (Volts)');
                    fcngen_parameters.cycles = inputdlg('Enter number of
cycles');
            end

            freq = str2num(string(fcngen_parameters.frequency));
            Vpp = str2num(string(fcngen_parameters.Vpp));
            cycles = str2num(string(fcngen_parameters.cycles));

        case 'OFF'
            fcngen_parameters.frequency = 3300000;
            fcngen_parameters.Vpp = 20;
            prevParameters = questdlg(sprintf('Are these function generator
parameters correct? \nFreq = %d MHz \nVpp = %d
Volts',(fcngen_parameters.frequency/1e6),fcngen_parameters.Vpp),'Functi
on Generator Parameters','Yes','No','No');

            switch prevParameters
                case 'No'
                    input = inputdlg('Enter frequency (MHz)');
```

```matlab
                fcngen_parameters.frequency = cellfun(@str2num,input) *
1e6;
                fcngen_parameters.Vpp = inputdlg('Enter Vpp (Volts)');
        end

        freq = str2num(string(fcngen_parameters.frequency));
        Vpp = str2num(string(fcngen_parameters.Vpp));
end

fcngen_output = [':SOUR2:APPL:SIN
',num2str(freq),',',num2str(Vpp),',0,0'];fprintf(obj2, fcngen_output);
%%Enables the harmonic function of CH1 and set the fundamental waveform
(Sine) parameters:
%%3.3 MHz frequency, 20 Vpp amplitude, 0 Vdc offset, and 0° start phase
fprintf(obj2, fcngen_output);
fprintf(obj2, [':SOUR2:BURS ', burst]); %Enables burst function of CH1
% having burst set to "OFF" would make the signal continuous
fprintf(obj2, ':SOUR2:BURS:MODE TRIG'); %Sets burst type of CH1 to N
cycle
fprintf(obj2, [':SOUR2:BURS:NCYC ',num2str(cycles)]); %Sets # cycles in
the N cycle burst of CH1 to 5
fprintf(obj2, ':SOUR2:BURS:INT:PER 0.01') %Sets internal burst period
of the N cycle burst of CH1 to 0.01 s
fprintf(obj2, ':SOUR2:BURS:TRIG:SOUR INT'); %Sets trigger source of the
burst mode of CH1 to Internal
fprintf(obj2, ':SOUR2:BURS:TRIG:TRIGO POS'); %Sets edge type of trigger
output signal of CH1 to Rising edge
fprintf(obj2, ':SOUR2:BURS:TDEL 0.00'); %Sets burst delay of the N
cycle burst of CH1 to 0.0 s
fprintf(obj2, ':SOURce2:BURSt:IDLE FPT'); %Sets idle level position to
First Point


%--- Adjust acquisition parameters ---

fprintf(obj1, ':ACQuire:TYPE NORMal'); %Sets acquisition mode
%%Options include {NORMal|AVERages|PEAK|HRESolution}

%fprintf(obj1, ':ACQuire:AVERages 128'); %Sets number of averages under
averages acqustion mode
%%Range includes 2^n where n is an integer between 1 and 10

%--- Adjust scale ---
pause(1.0);

fprintf(obj1, ':TIMebase:MODE MAIN');
%%Options: {MAIN|XY|ROLL} where MAIN = YT

fprintf(obj1, ':TIMebase:MAIN:SCALe 0.00001'); %Sets the main timebase
scale to 200?s/div (time/div)
%%YT mode: 5ns/div to 50s/div in 1-2-5 step; Roll mode: 200ms/div to
50s/div in 1-2-5 step

fprintf(obj1, ':CHAN1:PROBe 1'); %Sets probe ratio of specified channel
%%Options: {0.01|0.02|0.05|0.1|0.2|0.5|1|2|5|10|20|50|
100|200|500|1000}
```

```matlab
fprintf(obj1, ':CHAN2:PROBe 1'); %Sets probe ratio of specified channel
%%Options: {0.01|0.02|0.05|0.1|0.2|0.5|1|2|5|10|20|50|
100|200|500|1000}


fprintf(obj1, ':CHANnel1:OFFSet 0.00'); %Sets vertical offset of
specified channel
%%Related to the current vertical scale and probe ratio
%%When the probe ratio is 1X, vertical scale?500mV/div: -100V to +100V
vertical scale<500mV/div: -2V to +2V
%%When the probe ratio is 10X, vertical scale?5V/div: -1000V to +1000V
vertical scale<5V/div: -20V to +20V


fprintf(obj1, ':CHANnel2:OFFSet 0.00'); %Sets vertical offset of
specified channel
%%Related to the current vertical scale and probe ratio
%%When the probe ratio is 1X, vertical scale?500mV/div: -100V to +100V
vertical scale<500mV/div: -2V to +2V
%%When the probe ratio is 10X, vertical scale?5V/div: -1000V to +1000V
vertical scale<5V/div: -20V to +20V


fprintf(obj1, ':CHAN1:SCALe 0.48'); %Sets vertical scale of specified
channel FOR 3.3MHz
%%Related to the current probe ratio
%%When the probe ratio is 1X: 1mV to 10V
%%When the probe ratio is 10X (default): 10mV to 100V


%fprintf(obj1, ':CHAN2:SCALe 0.01'); %Sets vertical scale of specified
channel
%%Related to the current probe ratio
%%When the probe ratio is 1X: 1mV to 10V
%%When the probe ratio is 10X (default): 10mV to 100V


fprintf(obj1, ':CHAN1:RANGe 3.84'); %Sets range of vertical scale of
specified channel FOR 3.3MHz
%%Related to the probe ratio
%%When the probe ratio is 1X: 8mV to 80V
%%When the probe ratio is 10X: 80mV to 800V


%fprintf(obj1, ':CHAN2:RANGe 0.08'); %Sets range of vertical scale of
specified channel
%%Related to the probe ratio
%%When the probe ratio is 1X: 8mV to 80V
%%When the probe ratio is 10X: 80mV to 800V


fprintf(obj1, ':TIMebase:MAIN:OFFSet 0.000081'); %Sets main timebase
offset


%--- Adjust channel and data intake settings---
fprintf(obj1, ':WAVeform:SOURce CHAN2'); %Sets channel from which the
waveform is read
%%Options: {CHANnel1|CHANnel2|MATH}


fprintf(obj1, ':WAVeform:MODE NORM'); %Sets reading mode
%%Options: {NORMal|MAXimum|RAW}
```

```matlab
fprintf(obj1, ':CHAN1:COUPling DC') %Sets coupling mode of selected
channel
%%Options: {AC|DC|GND}

fprintf(obj1, ':ACQuire:MDEPth 12000') %Sets memory depth (number of
waveform points that can be stored in a single trigger sample)
%%Units: points; memory depth = sample rate * waveform length (timebase
scale)
%%Options: Single channel: {AUTO|12000|
120000|1200000|12000000|24000000}.
%%Dual channels: {AUTO|6000|60000| 600000|6000000|12000000}.

%--- Adjust Trigger Settings  ---
fprintf(obj1, ':TRIGger:COUPling DC'); %Sets trigger coupling
%%Options: {AC|DC|LFReject|HFReject}

fprintf(obj1, ':STOP');

fprintf(obj1, ':TRIGger:SWE:AUTO'); %Sets trigger mode to auto

fprintf(obj1, ':RUN');

fprintf(obj1, ':TRIGger:MODE EDGE'); %Sets trigger mode
%%{EDGE|PULSe|RUNT|WIND|NEDG|SLOPe|VIDeo|
PATTern|DELay|TIMeout|DURation|SHOLd| RS232|IIC|SPI}

fprintf(obj1, ':TRIGger:EDGe:SOURce EXT'); %Sets trigger source
%%Options: {CHANnel1|CHANnel2|AC|EXT}

fprintf(obj1, ':TRIGger:EDGe:SLOPe POS'); %Sets edge type in edge
trigger
%%Options: {POSitive|NEGative|RFALl}

fprintf(obj1, ':TRIGger:EDGe:LEVel 1'); %Sets trigger level in edge
trigger
%%Range: (-5 x VerticalScale - OFFSet) to (5 x VerticalScale - OFFSet)

%-- Organization system for folders --

currentfolder = pwd; %Requests current file directory
selpath = uigetdir('Data files'); %Requests user to choose a file
directory
cd(selpath) %Changes file path to selected path(selpath)

d = dir(selpath); % Gets the folder contents
dfolders = d([d(:).isdir]); % Removes all files (isdir property is 0)
dfolders = dfolders(~ismember({dfolders(:).name},{'.','..'})); %
Removes '.' and '..'

% **********BEGIN LOOP FOR REFERENCE MEASUREMENTS ******************
for refCount = 1:refs

    foldername = strcat('Ref_Trial',(num2str(refCount)));
```

```matlab
    i=1;
    while i <= length(dfolders)
        invalidchar = ["\","/",":","?","*","<",">","|",'"'] ;
        invalidid = contains(foldername,invalidchar); %First check for
invalid characters

        while invalidid ==1
            invalidchar = ["\","/",":","?","*","<",">","|",'"'];
            invalidid = contains(foldername,invalidchar); %Loop until
special characters are removed

            if invalidid == 1
                foldername = inputdlg('Please input a unique test name
without invalid special characters such as    [ \ / : ? * < > | " :
]'); %Requests user to rename folder
                foldername =
char(strcat((foldername),'_Trial',(num2str(refCount))));
            end
        end
        checkfolder = dfolders(i).name;
        tf = strcmpi(checkfolder,foldername); %Checks if the chosen
folder name exists already
        if tf == 1
            foldername = inputdlg('The file name you selected already
exists. Please select a unique name'); %Requests user to rename folder
            foldername =
char(strcat((foldername),'_Trial',(num2str(refCount))));
            i = 0;
        end
        i = i+1;
    end

    %-- Data processing --

    %pause(0.2); % in case it is sluggish the first time...
    nframes = 26;
    fprintf(obj1, strcat(':FUNCtion:WRECord:FEND ',num2str(nframes)));
%Sets max # frames recorded to 26
    fprintf(obj1, ':FUNCtion:WRECord:FINTerval 0.001'); %Sets time
interval between frames in recording to 1 ms
    fprintf(obj1, ':FUNCtion:WRECord:ENABle 0'); %Enables waveform
recording ability
    fprintf(obj1, ':FUNCtion:WRECord:OPERate RUN'); %Enables ability to
RUN

    for j = 1:2
        if j == 1

            %-- Acquires Reference Data --%
            if refCount == 1
                m = msgbox(sprintf('Press OK to start acquiring
reference data for %s', foldername));
                uiwait(m);
            end
```

53

```matlab
            h = msgbox(sprintf('Acquiring reference data for %s',
foldername));

            cd(selpath);
            mkdir(strcat(foldername,'_attenuation_reference'))%Creates
folder with selected name
            datahome =
strcat(selpath,'\',foldername,'_attenuation_reference'); %Defines
location of data folder
            cd(datahome); %Sets current directory to data folder
            tic
            for k = 1:1
                fprintf(obj1, ':WAVeform:SOURce CHAN1'); %Sets channel
from which the waveform is read
                %%Options: {CHANnel1|CHANnel2|MATH}

                fprintf(obj1, ':WAVeform:MODE NORM'); %Sets reading
mode
                %%Options: {NORMal|MAXimum|RAW}

                fprintf(obj1, ':CHAN1:COUPling DC') %Sets coupling mode
of selected channel
                %%Options: {AC|DC|GND}

                fprintf(obj1, ':ACQuire:MDEPth 12000') %Sets memory
depth (number of waveform points that can be stored in a single trigger
sample)
                %%Units: points; memory depth = sample rate * waveform
length (timebase scale)
                %%Options: Single channel: {AUTO|12000|
120000|1200000|12000000|24000000}.
                %%Dual channels: {AUTO|6000|60000|
600000|6000000|12000000}.
                fprintf(obj1, ':CHANnel1:DISPlay ON');
                fprintf(obj1, ':CHANnel2:DISPlay OFF');
                pause(1.0);
                i = 1;
                while i < 101
                    %vertscal = query(obj1, ':CHAN1:SCALe');
                    fprintf(obj2, ':OUTP1 OFF'); %Turns on output for
CH1 for function generator
                    fprintf(obj2, ':OUTP2 ON'); %Turns on output for
CH2 for function generator
                    fprintf(obj2, ':SOUR2:BURS:TRIG'); %Triggers a
burst output immediately on CH1 for fungen
                    %fprintf(obj1, ':SINGle'); %Enables single playback
- Only plays once
                    fprintf(obj1, ':TRIGger:SWE:AUTO'); %Sets trigger
mode to auto
                    pause(0.4);

                    fprintf(obj1, ':wav:data?');
                    %yinc = (vertscal/25); %in NORM mode
                    %yorig = str2num(query(obj1,
':WAVeform:YORigin?'));
```

54

```matlab
                    %yref = str2num(query(obj1,
':WAVeform:YREFerence?'));

                    %Request the data
                    [data1,len1]= fread(obj1);

                    %subtract = yorig + yref
                    %newdata 0= ([data] - subtract)*yinc

                    %Obtain oscilliscope properties
                    fprintf(obj1,':MEASure:SOURce CHANnel1'); %Obtain
parameters from CHANNEL 1
                    scopeprop(i).Period = str2double(query(obj1,
':MEASure:ITEM? PERiod')); %Obtain period
                    scopeprop(i).tVmax = str2double(query(obj1,
':MEASure:ITEM? TVMAX')); %Obtain time of Vmax
                    scopeprop(i).tVmin = str2double(query(obj1,
':MEASure:ITEM? TVMIN')); %Obtain time of Vmin
                    scopeprop(i).Vmax = str2double(query(obj1,
':MEASure:ITEM? VMAX')); %Obtain max voltage (in Volts)
                    scopeprop(i).Vmin = str2double(query(obj1,
':MEASure:ITEM? VMIN')); %Obtain min voltage (in Volts)
                    scopeprop(i).SampleRate = str2double(query(obj1,
':ACQuire:SRATe?')); %Obtain sampling rate (Sa/s)

                    query(obj1, ':WAVeform:PREamble?'); %Queries
waveform parameters

%%<format>,<type>,<points>,<count>,<xincrement>,<xorigin>,<xreference>,
<yincrement>,<yorigin>,<yreference>

                    %Waveform display on MATLAB
                    yincr1 = query(obj1, ':WAVeform:YINCrement?');
                    yoffset1 = str2num(query(obj1,
':CHANnel1:OFFSet?'));
                    wave = str2num(yincr1)*(data1(12:(len1-1))-127);
                    waves(k,i,:) = double(wave');
                    %     figure(1);
                    %     plot(squeeze(waves(k,i,:)));
                    %     pause(0.005);
                    new_offset1 = yoffset1 - mean(waves(k,i,:));
                    fprintf(obj1, char(strcat(':CHANnel1:OFFSet ',{'
'},num2str(new_offset1)))); %Sets vertical offset of specified channel
                    pause(0.1)
                    i = i + 1;
                end

                figure(5)
                plot(squeeze(mean(waves(k,:,:),2)))

                %leng = 1200; %# of samples
                xincr = query(obj1, ':WAVeform:XINCrement?');
                Fs = (1/str2num(xincr));
                N = size(waves,3);
                y = mean(fft(squeeze(waves(k,:,:))'),2);
```

```matlab
                f = (0:N/2-1)*(Fs/N); %Frequency range
                pow = (abs(y(1:N/2)).^2)/N; %Power spectrum
                figure(6)
                plot(f/1e6,10*log10(pow))
                xlabel('\bfFrequency (MHz)');
                ylabel('\bfMagnitude');

                %f = Fs*(0:(leng/2))/leng;
                %F = (log10(mean(abs(fft(waves')),2)));
                %plot(f,F(1:(length(f))))
                elapsedtime(k) = toc;
                PkV(k) = max(pow(find(f > 0 & f < 1e7)));
            end

            wavesRef = waves;
            xincrRef = xincr;

            clear wave tf checkfolder currentfile currentfolder
invalidchar invalidid len;
            currentfile =
strcat('ScopeDataAttenuation_',datestr(now,'yyyy_mm_dd_HH_MM_SS_FFF'),'
.mat');
            save(currentfile);
            plot(elapsedtime,PkV,'-o');
            xlabel('Time Elapsed (seconds)')
            ylabel('Peak Frequency Value')
            clear waves scopeprop;
            clear PkV;
            %-- Reference Data Acquired --%
        end
    end

    delete(h);
    if refCount == refs
        m = msgbox('Reference measurements are done.');
        uiwait(m);
    end
end

% **********BEGIN LOOP FOR SAMPLE MEASUREMENTS ******************
for groupCount = 1:groups

    foldername = inputdlg('Unique Test Name for Sample Measurement:');
%Requests user to name folder
    ntrials = inputdlg('What trial are you testing for this
experimental group?');
    foldername = char(strcat((foldername),'_Trial',(ntrials)));

    i=1;
    while i <= length(dfolders)
        invalidchar = ["\","/",":","?","*","<",">","|",'"'] ;
        invalidid = contains(foldername,invalidchar); %First check for
invalid characters

        while invalidid ==1
```

```matlab
                invalidchar = ["\","/",":","?","*","<",">","|",'"'];
                invalidid = contains(foldername,invalidchar); %Loop until
special characters are removed

                if invalidid == 1
                    foldername = inputdlg('Please input a unique test name
without invalid special characters such as    [ \ / : ? * < > | " :
]'); %Requests user to rename folder
                    foldername =
char(strcat((foldername),'_Trial',(ntrials)));
                end
            end
            checkfolder = dfolders(i).name;
            tf = strcmpi(checkfolder,foldername); %Checks if the chosen
folder name exists already
            if tf == 1
                foldername = inputdlg('The file name you selected already
exists. Please select a unique name'); %Requests user to rename folder
                foldername = char(strcat((foldername),'_Trial',(ntrials)));
                i = 0;
            end
            i = i+1;
        end

    %-- Data processing --

    %pause(0.2); % in case it is sluggish the first time...
    nframes = 26;
    fprintf(obj1, strcat(':FUNCtion:WRECord:FEND ',num2str(nframes)));
%Sets max # frames recorded to 26
    fprintf(obj1, ':FUNCtion:WRECord:FINTerval 0.001'); %Sets time
interval between frames in recording to 1 ms
    fprintf(obj1, ':FUNCtion:WRECord:ENABle 0'); %Enables waveform
recording ability
    fprintf(obj1, ':FUNCtion:WRECord:OPERate RUN'); %Enables ability to
RUN

    for j = 1:2
        if j == 1

            %-- Acquiring Sample Data --%
            m = msgbox(sprintf('Press OK to start acquiring sample data
for %s', foldername));
            uiwait(m);
            h = msgbox(sprintf('Acquiring sample data for %s',
foldername));

            cd(selpath);
            mkdir(strcat(foldername,'_attenuation'))%Creates folder
with selected name
            datahome = strcat(selpath,'\',foldername,'_attenuation');
%Defines location of data folder
            cd(datahome); %Sets current directory to data folder
            tic
            for k = 1:ntimepoints
```

```matlab
                fprintf(obj1, ':WAVeform:SOURce CHAN1'); %Sets channel
from which the waveform is read
                %%Options: {CHANnel1|CHANnel2|MATH}

                fprintf(obj1, ':WAVeform:MODE NORM'); %Sets reading
mode
                %%Options: {NORMal|MAXimum|RAW}

                fprintf(obj1, ':CHAN1:COUPling DC') %Sets coupling mode
of selected channel
                %%Options: {AC|DC|GND}

                fprintf(obj1, ':ACQuire:MDEPth 12000') %Sets memory
depth (number of waveform points that can be stored in a single trigger
sample)
                %%Units: points; memory depth = sample rate * waveform
length (timebase scale)
                %%Options: Single channel: {AUTO|12000|
120000|1200000|12000000|24000000}.
                %%Dual channels: {AUTO|6000|60000|
600000|6000000|12000000}.
                fprintf(obj1, ':CHANnel1:DISPlay ON');
                fprintf(obj1, ':CHANnel2:DISPlay OFF');
                pause(1.0);
                i = 1;
                while i < 101
                    %vertscal = query(obj1, ':CHAN1:SCALe');

                    fprintf(obj2, ':OUTP1 OFF'); %Turns on output for
CH1 for function generator
                    fprintf(obj2, ':OUTP2 ON'); %Turns on output for
CH2 for function generator
                    fprintf(obj2, ':SOUR2:BURS:TRIG'); %Triggers a
burst output immediately on CH1 for fungen
                    %fprintf(obj1, ':SINGle'); %Enables single playback
- Only plays once
                    fprintf(obj1, ':TRIGger:SWE:AUTO'); %Sets trigger
mode to auto
                    pause(0.4);

                    fprintf(obj1, ':wav:data?');
                    %yinc = (vertscal/25); %in NORM mode
                    %yorig = str2num(query(obj1,
':WAVeform:YORigin?'));
                    %yref = str2num(query(obj1,
':WAVeform:YREFerence?'));

                    %Request the data
                    [data1,len1]= fread(obj1);

                    %subtract = yorig + yref
                    %newdata 0= ([data] - subtract)*yinc

                    %Obtain oscilliscope properties
```

```matlab
                    fprintf(obj1,':MEASure:SOURce CHANnel1'); %Obtain
parameters from CHANNEL 1
                    scopeprop(i).Period = str2double(query(obj1,
':MEASure:ITEM? PERiod')); %Obtain period
                    scopeprop(i).tVmax = str2double(query(obj1,
':MEASure:ITEM? TVMAX')); %Obtain time of Vmax
                    scopeprop(i).tVmin = str2double(query(obj1,
':MEASure:ITEM? TVMIN')); %Obtain time of Vmin
                    scopeprop(i).Vmax = str2double(query(obj1,
':MEASure:ITEM? VMAX')); %Obtain max voltage (in Volts)
                    scopeprop(i).Vmin = str2double(query(obj1,
':MEASure:ITEM? VMIN')); %Obtain min voltage (in Volts)
                    scopeprop(i).SampleRate = str2double(query(obj1,
':ACQuire:SRATe?')); %Obtain sampling rate (Sa/s)

                    query(obj1, ':WAVeform:PREamble?'); %Queries
waveform parameters

%%<format>,<type>,<points>,<count>,<xincrement>,<xorigin>,<xreference>,
<yincrement>,<yorigin>,<yreference>

                    %Waveform display on MATLAB
                    yincr1 = query(obj1, ':WAVeform:YINCrement?');
                    yoffset1 = str2num(query(obj1,
':CHANnel1:OFFSet?'));
                    wave = str2num(yincr1)*(data1(12:(len1-1))-127);
                    waves(k,i,:) = double(wave');
                    %     figure(1);
                    %     plot(squeeze(waves(k,i,:)));
                    %     pause(0.005);
                    new_offset1 = yoffset1 - mean(waves(k,i,:));
                    fprintf(obj1, char(strcat(':CHANnel1:OFFSet ',{'
'},num2str(new_offset1)))); %Sets vertical offset of specified channel
                    pause(0.1)
                    i = i + 1;
                end

                figure(5)
                plot(squeeze(mean(waves(k,:,:),2)))

                %leng = 1200; %# of samples
                xincr = query(obj1, ':WAVeform:XINCrement?');
                Fs = (1/str2num(xincr));
                N = size(waves,3);
                y = mean(fft(squeeze(waves(k,:,:))'),2);
                f = (0:N/2-1)*(Fs/N); %Frequency range
                pow = (abs(y(1:N/2)).^2)/N; %Power spectrum
                figure(6)
                plot(f/1e6,10*log10(pow))
                xlabel('\bfFrequency (MHz)');
                ylabel('\bfMagnitude');

                %f = Fs*(0:(leng/2))/leng;
                %F = (log10(mean(abs(fft(waves')),2)));
                %plot(f,F(1:(length(f))))
                elapsedtime(k) = toc;
```

```matlab
                PkV(k) = max(pow(find(f > 0 & f < 1e7)));
            end

            wavesSample = waves;
            xincrSample = xincr;

            clear wave tf checkfolder currentfile currentfolder
invalidchar invalidid len;
            currentfile =
strcat('ScopeDataAttenuation_',datestr(now,'yyyy_mm_dd_HH_MM_SS_FFF'),'
.mat');
            save(currentfile);
            plot(elapsedtime,PkV,'-o');
            xlabel('Time Elapsed (seconds)')
            ylabel('Peak Magnitude')
        end
    end

    delete(h);
    if groupCount == groups
        m = msgbox('Samples are done.');
        uiwait(m);
    else
        groupnum = iptnum2ordinal(groupCount);
        m = msgbox(sprintf('Processing for sample ''%s'' is finished.
\nPress OK when ready to start your next sample.',foldername));
        uiwait(m);
        close all;
    end
end

%-- Close the VISA object --
fprintf(obj1, ':FUNCtion:WRECord:ENABle 0'); %Enables waveform
recording ability
fclose(obj1);
delete(obj1);
fclose(obj2);
delete(obj2);
% clear obj1 tf checkfolder currentfile d currentfolder dfolders
foldername i invalidchar invalidid len selpath;
clear obj1 tf checkfolder currentfile d currentfolder dfolders
foldername invalidchar invalidid len selpath;
```

CURRICULUM VITA

John Moore received his Bachelors of Science in Bioengineering from the University of Louisville in May 2021. He was a recipient of the Grawemeyer Scholarship due to his work in high school where he demonstrated an interest in undergraduate research. In the fall of 2019, he began his first co-op rotation at the University of Louisville Theranostic Ultrasound Laboratory. He worked there for his other two rotations, and during that time he contributed to multiple peer-reviewed publications and abstracts. For his Master's thesis, he continued his work with Dr. Jonathan Kopechek at the Theranostic Ultrasound Laboratory, where he developed both static and acoustofluidic ultrasound chambers to be utilized in his thesis research involving machine learning analysis of acoustic attenuation signals for cellular characterization. John will graduate with his Masters in Bioengineering from the University of Louisville in May 2022, and plans to continue his research at the Theranostic Ultrasound Laboratory during the pursuit of his Doctor of Philosophy degree.