

TACTILE AND MULTISPECTRAL BIMODAL IMAGING FOR BREAST CANCER RISK ASSESSMENT

A Dissertation
Submitted to
the Temple University Graduate Board

in Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

by
Vira Oleksyuk
August, 2021

Examining Committee Members:

Dr. Chang-hee Won, Advisory Chair, Dept. of Electrical and Computer Engineering
Dr. Joseph Picone, Dept. of Electrical and Computer Engineering
Dr. Iyad Obeid, Dept. of Electrical and Computer Engineering
Dr. Nancy Pleshko, Dept. of Bioengineering
Dr. Xiaojiang Du, External Member, Dept. of Computer and Information Sciences

ABSTRACT

American Cancer Society estimates that in 2021 nearly 300,000 women in the United States will be diagnosed with invasive breast cancer, and about 43,600 women will die from breast cancer. While many have access to health care and cancer screening, women from rural or underdeveloped communities often have limited access. Therefore, there is a need for an inexpensive and easy-to-use breast cancer identification device, which can be employed in small clinics to provide support to primary care physicians.

This work aims to develop a method to characterize breast tumors and tissue using non-invasive imaging modalities. The proposed bimodal imaging system has tactile and multispectral imaging capabilities. Tactile imaging modality characterizes tumors by estimating their depth, size, and stiffness, along with the Tactile Index. Multispectral imaging modality identifies breast asymmetry, texture, and inflammation changes, together with the Spectral Index. These indices are combined with the BCRAT Index, the risk score developed by the National Institute of Health, to form the Multimodal Index for personalized breast cancer risk assessment.

In this study, we will describe the development of the bimodal imaging system. We will present the algorithms for tactile and multispectral modalities. Tactile and Multispectral Profile Diagrams are developed to capture broad imaging signals in a compact and application-specific way. A Tactile Profile Diagram is a pictorial representation of the relative depth, size, and stiffness of the imaged tumor. A Multispectral Profile Diagram is a representative pattern image for breast tissue superficial optical properties. To classify the profile diagrams, we employ the Convolutional Neural Network deep learning method. We will describe the results of the experiments conducted using tissue-mimicking phantoms and human *in-vivo* experiments. The results demonstrate the ability of the method to classify and quantify tumor and tissue characteristics. Finally, we describe the method to calculate Multimodal Index for the malignancy risk assessment via tactile and multispectral imaging modalities and the risk probability based on the health records.

ACKNOWLEDGMENTS

The first person I would like to thank is my academic advisor, Dr. Chang-hee Won, who relentlessly helped me to build my research skills. Dr. Won engaged me in exciting projects, taught me how to find and develop research ideas, trusted my abilities, and gave me the freedom to come up with creative solutions. In addition, Dr. Won supported my successful NSF Graduate Research Fellowship application, sent me to research conferences, and presented unique opportunities to participate in business commercialization efforts of Tactile Imaging Probe. I am also grateful to Dr. Won for leaving his office door open, the notion that no problem is unsolvable but needs a careful approach, constant encouragement, and stable financial support through the years of graduate school.

I thank my Ph.D. dissertation committee members, Dr. Joseph Picone, Dr. Iyad Obeid, Dr. Nancy Pleshko, and Dr. Xiaojiang Du, for their time and kind support, valuable advice, and improvements to this dissertation work. They are an enormous source of knowledge and research expertise that I was very fortunate to access. In addition, I want to thank Dr. Picone for his empowering mentorship and numerous pieces of advice since the first time we met in 2010, for encouraging me to do research and supporting my interest to join Dr. Won's lab. Thank you, Dr. Picone, for showing me the opportunities and sharing your valuable time and experience.

I was privileged to learn from outstanding university professors and school teachers in the United States and Ukraine through the years. I am thankful to all for their passion and tremendous energy to teach, for their persistence, support, and their influence on students beyond academics, which can never be erased.

I could not have achieved this without my family and friends' unconditional support, constant encouragement, and faith in my abilities. I thank my beloved parents, Yaroslav and Lidiya Oleksyuk, and my grandparents. They are always exceptional role models since my early years, and they nurtured my passion for learning. Thank you to my dear husband, Andriy Petryshyn. He is the person I could entirely rely on and gain my strength from. His love and support helped me daily and were vital to my success. I also would like to thank my kids, Bohdan and Solomia. They are my weakness and strength at the same time and

a great inspiration. Thank you to my friends at CSNAP lab, who were the source of joyful energy and support. We had many great years together, learning and evolving. Special mention is for Firdous Saleheen and Sung In Choi. Firdous is thoughtful, intelligent, and fun to be around. He helped and advised me on multiple occasions. Despite us coming from different cultures, I am surprised at how we think alike on a broad range of topics. Thank you to Sung In for the precious time we spent together and the kind philosophical conversations, which helped me overcome some tough times.

This work would not be possible without collaborators and financial supporters. Thank you to Max Lin (ECE), Helen Freitas (BioE), Dr. Rouzbeh Tehrani (CE), and Dr. Gangadhar Andaluri (CE) for help, laboratory training, and resourcefulness. Thank you to Dr. Jie Yin (ME) and Dr. Qiuting Zhang for the aid with simulation modeling. My gratitude to Dr. Chetan Patil and Mohammed Arefin (BioE) for kindly sharing their expertise on tissue phantoms and near-infrared imaging. I appreciate helpful discussions on deep learning topics and shared valuable resources from Vinit Shah (ECE). I am also sincerely thankful to Nadeepuram Ranganathan and Barry Brucker from Avix Medical Inc. for providing opportunities, financial support, and business expertise. My extended gratitude is to Dr. Dina Caroline, Dr. Suzanne Pascarella, and Dr. Renee Kendzierski (TUH) for their unmeasurable support with clinical trials. Thank you to the National Science Foundation, the College of Engineering, the Department of Defense, the Pennsylvania Department of Health, and Edmund Optics for supporting my research work through generous fellowships and grants.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	ix
LIST OF TABLES	xii
CHAPTER	1
1 INTRODUCTION	1
1.1 Contributions	1
1.2 Research Goals	2
1.3 Peer-reviewed Publications	3
1.4 Dissertation Outline	5
2 BACKGROUND	6
2.1 Breast Diseases	6
2.1.1 Normal Breast Physiology	6
2.1.2 Benign Breast Diseases	7
2.1.3 Malignant Breast Diseases	8
2.2 Breast Disease Management	9
2.3 Breast Imaging Techniques	11
2.4 Summary	14
3 TACTILE IMAGING PROBE	15
3.1 Tactile Imaging Modality	15
3.2 Sensing Principle and Hardware Design	16
3.3 Acquisition Software Design and Implementation	18
3.4 Mechanical Properties Estimation Algorithm	21
3.4.1 Numerical Simulation for TIP Sensing Element	21
3.4.2 Tumor Size Estimation: 3D Interpolation	24
3.4.3 Tumor Region Stiffness Estimation	24
3.5 Semi-automatic Mechanical Properties Estimation GUI	26
3.6 Tactile Profile Diagram	28
3.6.1 Tactile Profile Diagram Construction	30
3.6.2 Mechanical Properties Estimation Algorithm from Tactile Profile Diagram	32
TPD Segmentation	32
Tumor Size Estimation	33
Tumor Region Stiffness Estimation	33
3.7 Classification Methods	34

3.7.1	Overview of Machine Learning Classification Methods	35
3.7.2	Convolutional Neural Network (CNN) Classification	36
	Convolution	38
	Detection	39
	Pooling	41
	Model Compilation	41
3.8	Classification of Tactile Profile Diagrams	44
3.8.1	Depth Classification Model	45
3.8.2	Size Classification Model	47
3.8.3	Stiffness Classification Model	47
3.9	Summary	49
4	TACTILE IMAGING PROBE EXPERIMENT	52
4.1	Phantom Development	52
4.1.1	Review of Phantoms that Mimic Mechanical Properties of Tissues	52
4.1.2	Proposed Phantom Design	53
4.2	Phantom Experiment	55
4.2.1	3D Interpolation for Size Estimation Using Tactile Images	55
4.2.2	Size and Stiffness Estimation Using Tactile Images	56
4.3	<i>In-vivo</i> Human Experiment Using Tactile Images	62
4.3.1	Size and Stiffness Estimation	62
4.3.2	Breast Tumor Risk Score	64
4.4	Tactile Profile Diagrams Using Phantom Data	67
4.4.1	Construction of Tactile Profile Diagrams	67
4.4.2	Size and Stiffness Estimation from Tactile Profile Diagrams	67
4.4.3	Classification of Tactile Profile Diagrams	71
	Depth Classification Model	72
	Size Classification Model	73
	Stiffness Classification Model	75
	Test Classification Results	76
4.5	Tactile Profile Diagrams Using <i>In-vivo</i> Human Data	77
4.5.1	Size and Stiffness Estimation from Tactile Profile Diagrams	78
4.5.2	Classification of Tactile Profile Diagrams	78
4.6	TPD Classification as Preliminary Step for Size Calculation	78
4.7	Summary	79
5	MULTISPECTRAL IMAGING PROBE	83
5.1	Overview of the Modality	83
5.2	Hardware Design and Imaging Principles	84
5.3	Acquisition Software	90
5.4	Imaging Wavelengths Selection	91
5.5	Optical Properties Estimation with Multispectral Imaging Probe	92
5.5.1	Image Pre-processing	92
5.5.2	Multispectral Profile Diagram	93

5.5.3	Multispectral Profile Diagrams for Differential Tissue Properties Estimation	97
5.5.4	Segmentation of Differential Multispectral Profile Diagrams	99
5.6	Classification of Multispectral Profile Diagrams	99
5.6.1	Asymmetry Classification Model	101
5.6.2	Texture Classification Model	102
5.6.3	Inflammation Classification Model	105
5.7	Summary	107
6	MULTISPECTRAL IMAGING PROBE EXPERIMENTS	108
6.1	Optical Property Phantom	108
6.1.1	Review of Optical Property Phantoms	108
6.1.2	Phantom Development	110
6.2	Wavelengths Selection Results	113
6.3	Differential Multispectral Profile Diagrams Using Phantom Data	118
6.3.1	Construction of Differential Multispectral Profile Diagrams	120
6.3.2	Classification of Differential Multispectral Profile Diagrams	121
Asymmetry Classification Model	121	
Texture Classification Model	123	
Inflammation Classification Model	124	
6.3.3	Test Classification Results	125
6.4	Summary	129
7	MULTIMODAL BREAST CANCER RISK ASSESSMENT	130
7.1	Review of Cancer Risk Calculations	130
7.2	Tactile Index	131
7.3	Spectral Index	131
7.4	NCI Breast Cancer Risk Assessment Tool	132
7.5	Multimodal Index	132
7.6	Multimodal Index Results	134
7.6.1	Bimodal Imaging Phantom	135
7.6.2	Multimodal Index Calculation Dataset	136
7.6.3	Multimodal Index Calculation Results	137
7.7	Summary	142
8	CONCLUSIONS AND FUTURE WORK	143
8.1	Conclusions	143
8.2	Future Work	144
	REFERENCES	155
	APPENDICES	155
	A HARDWARE CHARACTERISTICS OF TIP PROTOTYPES	156
	B TIP MICROCONTROLLER CIRCUIT IMPLEMENTATION	157

C	TIP LEDs ALIGNMENT PROCEDURE	165
D	YOUNG'S MODULUS TEST OF PDMS SAMPLES WITH INSTRON 5944	174
E	TPD CNN MODELS SOFTWARE IMPLEMENTATION	179
F	MPD CNN MODELS SOFTWARE IMPLEMENTATION	199

LIST OF FIGURES

2.1	Breast anatomy	7
3.1	TIP principle and hardware design	17
3.2	TIP controller circuit	17
3.3	TIP prototype connected to TIP GUI	18
3.4	TIP GUI setup	19
3.5	TIP GUI capturing capabilities	20
3.6	Example of the images saved during one TIP acquisition session	20
3.7	TIP GUI results tab	21
3.8	ABAQUS simulation setup	22
3.9	ABAQUS simulation results	23
3.10	Example of a 3D interpolation model	25
3.11	TIP sensing element deformation	25
3.12	GUI visualizations and calculations	27
3.13	Automatic detection of compression ranges	28
3.14	Example of a speech signal representation via a spectrogram	29
3.15	Tactile Profile Diagram generation	32
3.16	Tactile Profile Diagram segmentation	33
3.17	Tumor size calculation from a TPD	34
3.18	Deep learning representation [1]	37
3.19	Convolution operation via cross-correlation	39
3.20	Common activation functions	41
3.21	<i>Max pooling</i> operation	42
3.22	TPD classification steps	45
3.23	TPD depth classification model (TPDModelDepth)	46
3.24	Depth classification model training details	46
3.25	TPD size classification model (TPDModelSize)	47
3.26	Size classification model training details	48
3.27	TPD stiffness classification model (TPDModelStiffness)	49
3.28	Stiffness classification model training details	50
4.1	Breast phantom for TIP experiments	54
4.2	Phantom tumor samples	54
4.3	PDMS samples from Instron test	55
4.4	3D Interpolating models for the used PDMS	58
4.5	Breast phantom for tactile experiments	61
4.6	ROC curve to determine the threshold	65
4.7	Tactile Profile Diagram results	68
4.8	Linear size interpolation model results	70
4.9	Test phantom components and their placement during the TIP experiments	70
4.10	Samples for size and stiffness calculation from TPDs	70

4.11	Stiffness Index for shallow and deep inclusions	72
4.12	Accuracy and loss during training of TPDMoDelDepth over 50 epochs	73
4.13	Examples of tumors depth classification using TPDMoDelDepth	74
4.14	Accuracy and loss during training of TPDMoDelSize over 50 epochs	74
4.15	Examples for tumors size classification using TPDMoDelSize	75
4.16	Accuracy and loss during training of TPDMoDelStiffness over 50 epochs . . .	76
4.17	Examples of tumors stiffness classification using TPDMoDelStiffness	76
5.1	Hyperspectral image vs. bitmap image [2]	84
5.2	Light reflectance, scattering, and transmission within a tissue sample	85
5.3	Absorption spectra of different tissue chromophores [3]	85
5.4	Hyperspectral camera acquisition system (Retiga EXi F-M-12-C)	87
5.5	TIP camera (IDS UI-3240CP-NIR) and lens	87
5.6	Comparative characteristic of the quantum efficiency curve for IDS UI-3240CP-NIR vs. Retiga EXi F-M-12-C	87
5.7	Two imaging systems setups	88
5.8	Multispectral Imaging Probe components and experimental setup	89
5.9	Multispectral Imaging Probe graphical user interface	90
5.10	Spectra of the key tissue absorbents in NIR window	91
5.11	Construction of a Multispectral Profile Diagram	95
5.12	An example of Multispectral Profile Diagram	96
5.13	Multispectral Profile Diagram data points	97
5.14	Formation of a Differential Multispectral Profile Diagram	98
5.15	Image filtering in the spatial domain	99
5.16	Two examples of Differential Multispectral Profile Diagram texture-enhancement	100
5.17	Inflammation segmentation on a Differential Multispectral Profile Diagram	100
5.18	Classification of Differential Multispectral Profile Diagrams	101
5.19	Asymmetry classification model structure (MPDMoDelAsymmetry)	102
5.20	Asymmetry classification model details	103
5.21	Texture classification model structure (MPDMoDelTexture)	103
5.22	Texture classification model details	104
5.23	Inflammation classification model structure (MPDMoDelInflammation)	105
5.24	Inflammation classification model details	106
6.1	MIP phantom design	111
6.2	Description and quantity of MIP samples	111
6.3	Skin tone and blood amount variations in MIP phantoms	112
6.4	MIP phantom test patterns and depths	113
6.5	MIP asymmetry parameter implementation	114
6.6	Wavelengths selection experiment samples	114
6.7	Reflectance and absorption/scattering spectra for the experimental samples .	115
6.8	Absorbance/scattering spectra for water, lipids, and PDMS samples	116
6.9	Absorbance/scattering spectra for the blood samples	117
6.10	Selected wavelengths for MIP	118
6.11	Examples of Differential Multispectral Profile Diagrams	119

6.12	Examples of Differential MPDs from two phantom samples	120
6.13	Asymmetry model training accuracy and loss	122
6.14	Asymmetry classification examples using MPDModelAsymmetry	122
6.15	Texture model training accuracy and loss	123
6.16	Texture classification examples using MPDModelTexture	124
6.17	Inflammation model training accuracy and loss	125
6.18	Inflammation classification examples using MPDModelInflammation	125
6.19	Independent classification test samples	126
7.1	The BCRAT calculator questions [4]	133
7.2	Example of the NCI BCRAT calculator results [4]	133
7.3	Multimodal Index development	134
7.4	Bimodal imaging phantom with optical and mechanical property	135
7.5	Bimodal imaging phantom	136
7.6	Multimodal Index calculation results	141
8.1	Future developments in Multimodal Index methodology	145
B.1	TIP Microcontroller circuit implementation	157
C.1	LED and spacers configuration schematics in TIP design	166
C.2	Alignment implementation	167

LIST OF TABLES

4.1	TIP size estimation interpolation parameters	57
4.2	Size and Deformation Index estimation at shallow tumor depths	59
4.3	Size and Deformation Index estimation at deep tumor depths	60
4.4	The <i>in-vivo</i> test results for estimating size and deformation index using TIP	63
4.5	Risk Score based classification of tumors using TIP output	66
4.6	Size interpolation development samples	69
4.7	Properties of the phantom components	70
4.8	Size and Stiffness Index calculation for the phantom dataset	71
4.9	CNN classification results for the phantom data	77
4.10	Results for human data size and stiffness estimation	81
4.11	CNN classification results for the human dataset	81
4.12	CNN classification results for the selected samples	82
4.13	Size calculation results for the selected samples	82
5.1	Comparative characteristics of the two cameras (IDS UI-3240CP-NIR vs. Retiga EXi F-M-12-C)	88
6.1	Independent classification test samples description	127
6.2	CNN classification results for the independent test dataset	128
7.1	Tactile Index calculation results	138
7.2	Spectral Index calculation results	139
7.3	Multimodal Index calculation results	140
A.1	Comparative hardware characteristics for three generations of TIP prototypes	156
C.1	List of materials and tools	165
D.1	PDMS Young's Modulus results	175

CHAPTER 1

INTRODUCTION

In this first chapter, we outline the contributions of the dissertation work with the research goals. Next, we list the peer-reviewed publications. Finally, we present an outline of the dissertation.

1.1 Contributions

There are eight main contributions in this dissertation.

- The combination of tactile imaging and multispectral imaging is proposed to aid doctors in breast cancer diagnostic decisions.
- Tactile Imaging Probe's hardware, firmware, and software are developed to facilitate tactile imaging.
- A novel method to analyze tactile imaging data by constructing Tactile Profile Diagrams is proposed. These pattern images with the encoded tumor depth, size, and stiffness information from sets of tactile images are developed and analyzed.
- Convolutional Neural Network classification models are built to extract information about tumors' depth, size, and stiffness from Tactile Profile Diagrams.
- Experimental evaluations of the developed methods are completed on the custom made breast tissue mimicking phantoms, and on *in-vivo* human data.
- Multispectral Imaging Probe's hardware and software are designed and implemented to support multispectral imaging.
- A novel method to analyze multispectral imaging data by constructing Multispectral Profile Diagrams is developed. The pattern images with the encoded tissue optical differential properties information from sets of multispectral images are developed and analyzed.

- Convolutional Neural Network classification models are proposed to gather information about affected breast tissue asymmetry, texture, and inflammation size from Multispectral Profile Diagrams. The models are trained and validated.
- A bimodal breast tissue and tumor phantom design is described. The custom phantom mimics the mechanical and optical properties of the human breast tissue. The phantom was fabricated and tested.
- A method to combine mechanical and optical properties estimations from the dual modality imaging with additional patient's health information into a single individualized Multimodal Index for breast cancer risk assessment is proposed and tested.

1.2 Research Goals

The main goal of this research is to develop a bimodal imaging system for identifying breast cancer risk. The two modes of the imaging system are tactile imaging and multispectral imaging. We develop Tactile Imaging Probe's hardware and software, as well as its algorithms, to measure the mechanical properties, such as the tumor size, stiffness, and depth within the breast tissue. We develop Multispectral Imaging Probe and its algorithms to characterize superficial breast tissue properties, such as inflammation, texture, and asymmetric changes. Finally, we propose a method to calculate Multimodal Index of the imaged breast tumor and tissue using the characterization indices obtained from the two imaging modalities and the patient's health information.

1.3 Peer-reviewed Publications

Here is the list of peer-reviewed publications:

1. **V. Oleksyuk**, D. Caroline, S. Pascarella, R. Kendzierski, and C.-H. Won, "Tissue Inclusion Characterization using Tactile Profile Diagrams," Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2021) (*Submitted for review*).
2. S. Choi, **V. Oleksyuk**, and C.-H. Won, "Breast Tumor Malignancy Classification using Smartphone Compression-induced Sensing System and Deformation Index Ratio," *Annual International Conferences of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Montreal, Canada: Jul. 2020.
3. **V. Oleksyuk**, R. Rajan, F. Saleheen, D. Caroline, S. Pascarella and C.-H. Won, "Risk score based pre-screening of breast tumor using compression induced sensing system," *Sensors Journal*, IEEE, Vol.18 (10), pp. 4038-4045, 2018.
4. F. Saleheen, **V. Oleksyuk**, and C.-H. Won, "Itchy skin region detection using hyperspectral imaging," Proc. SPIE Defense-Commercial Sensing, Paper 10656-7, Baltimore, MD: USA, 2018.
5. **V. Oleksyuk**, F. Saleheen, D. Caroline, S. Pascarella and C.-H. Won, "Classification of breast masses using tactile imaging system and machine learning algorithms," IEEE Signal Processing in Medicine and Biology Symposium, Philadelphia, PA: USA, Jan 2017.
6. **V. Oleksyuk**, F. Saleheen, D. Caroline, S. Pascarella, C.-H. Won. "KNN classification of tactile imaging data," 2017 International Symposium on Innovation in Information Technology and Application (ISIITA17), Danang: Vietnam, Jan. 2017.
7. C.-H. Won, J. Goldstein, **V. Oleksyuk**, D. Caroline, S. Pascarella. "Tumor size and elasticity estimation using smartphone-based compression-induced scope," 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 17), JeJu Island: S. Korea, July 2017, pp. 4106-4109.

8. F. Saleheen, Z. Wang, W. Moser, **V. Oleksyuk**, J. Picone, C.-H. Won, "Effectiveness of virtual open laboratory teaching assistant for circuits laboratory," 123rd ASEE Annual Conference, New Orleans, LA: USA, June 2016.
9. **V. Oleksyuk**, F. Saleheen, W. Moser, and C.-H. Won, "Tactile imaging sensor for mechanical properties quantification of breast tumor," IEEE Biomedical Circuits and Systems Conference (BioCAS), Atlanta, GA: USA, Oct. 2015.
10. A. Sahu, F. Saleheen, **V. Oleksyuk**, C. McGoverin, N. Pleshko, A. Harati, J. Picone and C.-H. Won, "Characterization of Mammary Tumors Using Noninvasive Tactile and Hyperspectral Imaging Sensors," *Sensors Journal*, IEEE, Vol.14 (10), pp. 3337-3344, 2014.
11. A. Sahu, F. Saleheen, **V. Oleksyuk**, Y. Chen, and C.-H. Won, "Tactile and hyperspectral imaging sensors for mammary tumor characterization," *Sensors*, 2013 IEEE, Baltimore, MD: USA, Nov 2013.
12. F. Saleheen, **V. Oleksyuk**, A. Sahu, and C.-H. Won, "Non-invasive mechanical properties estimation of embedded objects using tactile imaging sensor," Proc. SPIE 8719, Smart Biomedical and Physiological Sensor Technology X, Baltimore, MD: USA, May 2013.
13. F. Saleheen, A. Sahu, **V. Oleksyuk**, and C.-H. Won, "Normal force estimation using tactile imaging sensor," Bioengineering Conference (NEBEC), 2012 38th Annual Northeast, Philadelphia, PA: USA, Mar. 2012, pp. 119-120.

1.4 Dissertation Outline

This dissertation consists of eight chapters.

Chapter 2 presents the background information on normal and abnormal breast physiology. This chapter also gives a review of current breast cancer screening techniques and the general overview of tactile and multispectral imaging.

Chapter 3 describes the Tactile Imaging Probe. The hardware design and acquisition software are presented in detail. We explain size and stiffness estimation algorithms designed to work on sets of tactile images. Then we present the automatic data pre-processing software. Next, we describe the development of the Tactile Profile Diagrams as a method to capture the dynamic properties of the signal. We also present an overview of different types of classification methods and outline algorithms for tumor depth, size, and stiffness estimation using Tactile Profile Diagrams and Convolutional Neural Network.

Chapter 4 presents the results from the Tactile Imaging Probe experiments conducted using silicone breast tissue mimicking phantoms and using *in-vivo* human study. We also show how estimations of depth, size, and stiffness of a tumor with CNN can improve the size calculation.

Chapter 5 describes the multispectral imaging modality. We present the hardware design and the acquisition software. Then, we describe the image pre-processing techniques to improve data quality. Later, we outline methods for the Multispectral Profile Diagram development, the optical properties estimation from the diagrams using deep learning classification, and the Spectral Index developed to characterize optical tissue properties. Multispectral modality is effective in detecting inflammatory breast cancer manifestations.

Chapter 6 shows the results from the Multispectral Imaging Probe experiments. The design of a bimodal tissue mimicking phantom is given, and experiments are described.

Chapter 7 outlines the method for Multimodal Index calculation for breast cancer risk assessment based on the Tactile Index, Spectral Index, and the patient's personal data (BCRAT Index). This chapter also presents the results from Multimodal Index calculation using phantom data.

Chapter 8 presents conclusions and future research.

CHAPTER 2

BACKGROUND

In this chapter, we review human breast physiology, breast diseases, and breast cancer screening techniques and tools. We also review tactile imaging and multispectral imaging in biomedical applications.

2.1 Breast Diseases

There is a broad spectrum of breast diseases, some of which are benign, and some are malignant or cancerous. Depending on the diagnosis, a patient gets a specific procedure and treatment plan to follow [5]. It is crucial to know breast anatomy and physiology to understand breast cancer.

2.1.1 Normal Breast Physiology

The breast consists of multiple tissues with structural and supportive functions [6]. Fig. 2.1 shows the breast composition. The main components of the breast are skin, superficial fascia, and breast parenchyma. Skin is the protective layer that is covering the breast. It includes nipple and areola complex. The deep layer of skin, dermis, merges with the superficial fascia, which in turn transitions to the breast parenchyma. The breast parenchyma consists of several tissues, such as fibrous stroma, glandular epithelium, fat, and supporting structures [6]. The parenchyma is built of segments called lobes in a radial arrangement. The lobes are divided into smaller structures, lobules, and milk collecting ducts converging within the nipple region [6, 7]. The breast undergoes physiological changes over a lifespan, causing the change in the breast composition. For example, the proportion of glandular tissue to fat tissue decreases with age [6], and it lowers the density of the breasts.

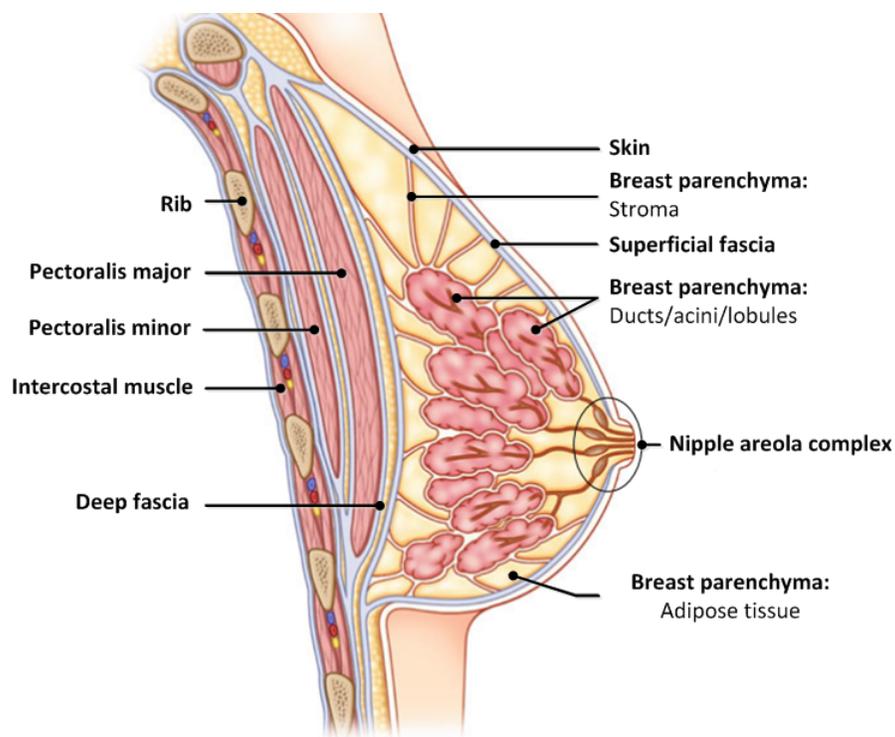


Figure 2.1: Breast anatomy

2.1.2 Benign Breast Diseases

Benign breast diseases are breast diseases that cause pain, with or without masses or nipple discharge, but they are not cancerous. The vast majority of women (90%), who require attention from a health care specialist, have benign conditions [8].

There are several main causes of benign breast conditions: fibrocystic breast disease, benign breast tumors, and breast inflammation or mastitis [5]. The benign conditions are not cancerous by definition, yet they may elevate chances for patients to develop breast cancer in the future. Doctors use a detailed patient's health history review and a careful physical examination to diagnose a breast disease [5].

Fibrocystic breast disease is caused by hormonal changes. The breast feels lumpy and painful. The lumps are the fluid-filled lobules, which are also called cysts. They are more common in younger women [8]. More than half of all pre-menopausal women go through Fibrocystic change [5].

Benign breast masses are mostly caused by fibroadenoma [5]. The lumps are composed

of fibrous and epithelial tissues and are very mobile during a breast examination [8].

Mastitis is an infection of breast tissues, which is common for lactating women and women with large breasts. Mastitis manifestations are pain, fever, swelling, none, single or multiple breast masses, and an abscess-like inflammation [6]. Mastitis has a rapid onset, yet short term if treated properly [9].

2.1.3 Malignant Breast Diseases

Many different malignant or cancerous breast diseases are known. Their ability to invade surrounding tissues differentiates them from benign conditions. The disease manifestations for each patient are defined by the type of cancer and the time of the diagnosis. Possible symptoms of breast cancer include sudden breast mass development, discharge, breast skin color or texture changes, nipple shape changes, or itch development [5].

The most common cancerous breast tumors are invasive ductal and lobular adenocarcinomas, which are cancers of ductal or lobular glandular tissues, respectively [5].

Ductal carcinoma *in situ* and lobular carcinoma *in situ* are the conditions where tumors did not cross the inner membranes and invade other surrounding tissues. These conditions carry an increased risk for a patient to develop cancer [5].

Inflammatory breast cancer (IBC) is rare (about 2% of breast cancers) but a very aggressive type of breast cancer, which often develops in women at a relatively young age [10, 11]. IBC is characterized by a very rapid onset of symptoms (from days to up to 3-6 months) [9, 12] and is unilateral in most cases [10, 13].

IBC symptoms are similar to mastitis symptoms; however, unlike mastitis, IBC is non-infectious in nature. While the local breast tissue temperature may be increased [9, 12, 14], it does not cause fever or the elevated number of white cell count in the blood [12].

A palpable tumor may or may not be present in patients with IBC [9, 12]. Most frequently, IBC develops without an underlying tumor [15].

The breast absolute or relative asymmetry is higher for IBC patients [16]. Nipple changes, such as flattening, crusting, or retraction, are common in IBC patients [9, 14].

IBC is also characterized by skin texture changes. The shortening in the length of suspensory ligaments of Cooper, which is a part of stroma (Fig. 2.1), causes stiffening of

the breast and changes in breast skin texture. Breast skin is characterized by thickening and distortion in 80% of patients with IBC [12]. The texture changes cause the breast skin to look like an orange peel, which is commonly referred to by doctors as *peau d'orange* appearance [17]. *Peau d'orange* of the breast, bruising [14], ridging [9, 10, 12, 14], and ulceration [14] are all common texture changes for breast cancer patients.

While IBC symptoms are very similar to mastitis symptoms, in IBC patients, the area of the affected reddened skin (inflammation area) usually involves the entire breast [7]. The color of inflammation ranges from pink to violet [18, 19, 20, 21], which may be presented in patches [14].

2.2 Breast Disease Management

Breast masses are the most common breast disease manifestation, where the majority of them being benign [7]. Breast masses can be detected by patients during self breast examination (SBE), by medical practitioners during a clinical breast examination (CBE), or during a routine screening mammogram [5]. In modern practice, SBE and CBE are not the primary tools for a cancer diagnosis; however, they are very important supplemental screening techniques to mammography, ultrasound, or MRI imaging. CBE is used to identify the affected tissues and even to find 15% of the malignant tumors undetected during mammography [6].

Regular breast cancer screenings are recommended for asymptomatic women [7]. Women with an increased risk for developing breast cancer should follow a more frequent evaluation procedure schedule than the one developed for the regular public [5]. The patient's health history plays a role in breast cancer risk assessment. The American Cancer Society specifies many types of well-established breast cancer risk factors, which are the advanced age, family history of breast cancer, the inheritance of some gene mutations, dense breast tissue, a benign breast condition, early menarche, late menopause age, chest radiation, unhealthy weight, alcohol consumption, no physical activity, no children, not breastfeeding, hormonal therapy, hormonal birth control, and breast implants [22].

The National Institute of Health developed an interactive online tool for breast cancer

risk assessment based on the Gail model, which can be used by patients and clinicians [4]. The Gail model was created in 1989 by Dr. Mitchell Gail and his colleagues from NIH based on the large case study of 280,000 women. It is a statistical breast cancer risk assessment algorithm, which gives 5-year and 10-year probabilistic estimates of a woman developing breast cancer compared with the average risk. The model was initially developed for white women; since then, the model was supplemented and improved to work for Afro-American, Asian, and Hispanic women as well. The Gail model is instrumental in estimating highly individual breast cancer risk and in improving the breast cancer screening guidelines [23].

When a tumor is found, physicians complete a detailed assessment of the tumor and the affected breast, evaluate for the possibility of malignancy, and aim to give an accurate diagnosis [7]. To do so, most often, they use mammography and ultrasound imaging to visualize the breast tissues and suspicious masses. Mammography is a powerful breast cancer screening tool, which helps detect microcalcifications, commonly caused by cancer. Ultrasound technology aids medical doctors in differentiating cysts from solid tumors. The patients with cysts usually do not need further evaluations, yet the patients with solid masses are scheduled to undergo a biopsy procedure and exclude malignancy. Other supplemental tests are available to support the diagnosis [5].

To date, there are many prognostic factors identified for breast cancer patients. However, the most significant of them are the presence and the amount of metastasis in axillary lymph nodes and the tumor size. These two factors are used to stage the tumor, and therefore, they are the ones that define the treatment plan for the patient. The more metastatic lymph nodes are found, and the larger tumor is, the more ominous survival prognosis is given to the patient by doctors [6].

Doctors should exclude the possibility of inflammatory breast cancer (IBC) in patients with symptoms similar to mastitis [7]. However, there are no standard diagnostic criteria for IBC available at this time [10, 12]. Clinicians diagnose IBC in several steps: a visual observation of symptoms in the suspicious breast, a review of the patient's medical history, completion of a physical examination, and a pathological evaluation of a skin biopsy samples and/or needle or core biopsy samples for invasive carcinoma [9, 10]. An inflammation

(erythema) area has to occupy at least one-third of the breast for IBC diagnosis [9, 12]. The detection of erythema in African Americans may be challenging because redness is less pronounced in this group [10]. The timing of the IBC diagnostic is very important. If a patient was treated for mastitis for more than 7-14 days with no results, IBC may be suspected [9, 18]. There is a need to detect IBC early.

2.3 Breast Imaging Techniques

Multiple imaging technologies are developed for breast cancer screening purposes, such as screening mammography, ultrasound, magnetic resonance imaging, computer tomography, ultrasound elastography, thermography, optical imaging, and others [7, 24]. The goal for screening is to detect the disease early and to give the best chance of survival and cure to the patient. Positive screening result follows by additional tests to confirm malignancy [7].

Here we provide a brief overview of different breast imaging modalities.

Mammography

Mammography is a gold standard method for breast cancer screening [7, 24]. It relies on low-dose X-rays to image the breast placed between a source and detector plates. Microcalcification in the breast, which can indicate malignancy, looks brighter on the mammogram, whereas the breast tissues look darker. However, it is challenging to detect early-stage malignancy, especially in women with dense breasts. Therefore, additional screening tests are used as supplements to obtain better results for cancer detection. Recently, a digital 3D mammography was developed to characterize breast tissues better. The modality uses the functional principles of mammography yet is modified to acquire three-dimensional views of the breast.

Computer-aided detection (CAD) is a set of machine learning algorithms developed to support doctors with the identification of small tumors, which has now become a standard feature in modern mammography machines. Besides all of its advantages, CAD is associated with a high false positive rate and may mislead radiologists [24].

Ultrasound

Ultrasound (US) is the first choice for the supplemental to mammography diagnostic

modality and can also be used as a stand-alone test in some cases. It is used to image patients, which had inconclusive mammogram results due to high breast density or other factors [7].

The US method uses high frequency sound waves to image the tissues. A US transducer sends and senses the reflected inside the tissue waves to construct the ultrasound image. The images are generated in real-time, while the transducer is moved over the region. The cons of the method are the high rate of false positives and false negatives during screening.

One of the notable advancements in US technology is the development of ultrasound breast elastography, which can characterize tissue elasticity. There are two types of US elastography: free-hand elastography and shear wave elastography. The first relies on the light compression of the transducer on the tissue during acquisition, which creates a displacement field for strain estimation. The second method relies on the generated additional shear waves in transducers, which are used for strain estimation as well. Stiffness information is displayed as a range of colors, and the elastography image is superimposed over the corresponding US image [25].

Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) uses a single proton (a hydrogen nucleus) to image tissues. It is possible because different tissue types are composed of molecules with different amounts of hydrogen nuclei. MRI is using the magnetic properties of the nuclei to analyze the tissues. The patient is placed in a magnetic field, and then radio frequency waves are applied to acquire an image sequence.

The pros of the method include high contrast of the images, no radiation concerns to patients, the ability to image both breasts together, and the ability to image dense breasts efficiently. The cons of the method are the high cost of the imaging [7], high false-positive rate, and microcalcifications that may not be shown correctly on images [24].

Positron Emission Tomography

Positron emission tomography (PET) is a nuclear medicine imaging technique, which uses gamma rays to image tissues and visualize their functional characteristics. To distinguish benign from malignant tumor images, PET modality uses the differences in levels of glucose metabolism. The relatively higher level of metabolism in the tissue will indicate

malignancy.

PET is a useful supplemental method for detecting and staging breast cancer. It also can be coupled with MRI and mammography to overcome the issue with breast density and the menopausal status of a woman. There are also negative aspects of this method. The PET experiment cost is high, the resolution of the images is low, and patients are exposed to radiation [24].

Thermography

Thermography imaging of the breast is sometimes also called digital infrared imaging. It is designed to use the infrared light spectrum for breast cancer detection [7]. Cancerous and pre-cancerous tissues are characterized by increased growth and higher metabolism than surrounding normal tissues. When the tissue becomes cancerous, it starts to draw more blood supply. These changes are causing the increased temperature in the affected areas, which can be detected by thermography. Thermography is a promising screening modality; however, its performance suffers from high false positive rate, high sensitivity to the test conditions, and image interpretation difficulties [24].

Diffuse Optical Imaging

Diffuse Optical Imaging (DOI) uses the near infrared wavelength band (650 nm - 1000 nm) to find tumors within a breast. The two types of DOI: diffuse optical tomography and diffuse optical mammography, use the same principle yet different wavelengths for imaging. The pros of the method are that it is non-invasive and not ionizing. It can give a quantitative assessment of the imaged tissues for hemoglobin saturation and concentration. The cons of the method are the difficulty to accurately measure biochemical components, to reconstruct an image, and the use of contrast mechanisms [24].

Electrical Impedance-based Imaging

The human body is a collection of tissues with different permeability properties. Research shows that cancerous tissues have lower impedance than normal tissues [24]. Two imaging methods are developed based on that property: electrical impedance tomography (EIT) and electrical impedance scanning (EIS). Multiple electrodes are placed on the breast to detect the impedance level samples, and then images are reconstructed from the values and analyzed [24]. The pros of the imaging method include its non-ionizing nature, the

possibility of continuous patient monitoring, and the chance of cost reduction for the diagnostic. The cons of the methods are the comparatively low resolution of the impedance methods to computer tomography or MRI images, challenging image reconstruction algorithms, and complex hardware implementation [26].

Computer Tomography

Computer tomography (CT) uses X-rays to capture planar images or slices of the body. CT algorithms can recreate three-dimensional images from the taken slices, which can help to localize the tumor. The combination of CT imaging with PET is used to define the stage for metastatic cancers because it is able to assess the location of the tumor and the metabolism level of it [24].

CT method has multiple advantages with high quality of anatomical details visualization, yet uses 70 times higher level of ionizing radiation than a chest x-ray to complete a test [27]. The multiple uses of CT can increase the risk of cancer [28]. In addition, approximately 50% of the CT imaging procedures require to use of the intravenous iodine contrast agent, to which patients may develop from mild to fatal allergic reactions [29].

2.4 Summary

This chapter presents an overview of breast physiology and the typical manifestations of benign and malignant breast diseases. Next, the breast disease management strategies are outlined. Finally, we gave the list and a short description of the available breast cancer imaging modalities.

Despite all of the advancements in imaging technology, no imaging method stands out for breast cancer pre-screening. Most imaging modalities are for diagnostic purposes. There is a strong need for a multimodal imaging method for breast cancer pre-screening.

CHAPTER 3

TACTILE IMAGING PROBE

In this chapter, we introduce Tactile Imaging Probe (TIP) for breast tumor characterization, its principle, and hardware design. We will also describe the acquisition software and its capabilities. Next, we outline the algorithm and the graphical user interface for the size and stiffness estimation. We describe the method to capture the tactile properties of the tissue under compression by constructing Tactile Profile Diagrams, the pattern images with tissue tactile information. We describe the method to estimate the size and stiffness of the embedded tumors directly from Tactile Profile Diagrams. Finally, we introduce Convolutional Neural Network models to classify Tactile Profile Diagrams based on the depth, size, and stiffness.

3.1 Tactile Imaging Modality

Research shows that cancerous masses tend to be stiffer than benign ones [30, 31], and in some studies, malignant tumors are found to be tenfold stiffer than normal breast tissue through compression experiments [32]. It is also shown that mechanical imaging can classify breast tumors as benign or malignant [33]. Moreover, the size of the detected tumor defines its grade during diagnosis and defines the future treatment plan for the patient [6].

We developed Tactile Imaging Probe (TIP) and its algorithms to mimic human touch sensation and to non-invasively measure the size and stiffness of embedded tumors [34, 35]. We also introduced the malignant/benign classification method to support doctors.

Tactile imaging modality is an inexpensive, non-invasive, and non-ionizing method for breast tumors characterization. This method provides information about the tumor's mechanical properties for pre-screening purposes.

3.2 Sensing Principle and Hardware Design

The Tactile Imaging Probe (TIP) is designed to mimic human touch sensation. From tactile images, TIP algorithms quantify the size and stiffness of the imaged tissues and tumors.

Fig. 3.1 shows sensing principle and hardware design of Tactile Imaging Probe. TIP uses a transparent and highly flexible polymer as a sensing element, which is made from Polydimethyl siloxane (PDMS). We insert visible light to scatter within the sensing element using four light emitting diodes (LEDs) and the principle of total internal reflection [36]. When the sensing element is compressed against tissue with a tumor, its contact surface is deformed. This deformation causes the scattered within the polymer light to escape and to be captured by the lens-camera unit. We use a UI-3240CP-NIR monochrome CMOS Camera (IDS Imaging Development Systems GmbH, Obersulm, Germany), and a 5 Megapixel 12 mm lens (Schneider, Rhineland-Palatinate, Germany). The size of a tactile image is 1024 pixel \times 1280 pixel. The depth of the pixel is 8 bit, which makes the pixel value range from 0 to 255. The image acquisition rate is 20 images per second [37]. Attachment A and Attachment C provide supplemental information on TIP hardware and firmware development.

The compression force is measured with an FC22 Force Sensor (TE Connectivity, USA) attached on top of the camera body. The typical number of images in one acquisition session is 50.

The controller unit consists of a microcontroller (SparkFun Pro Micro 5V/16 MHz, SparkFun Electronics, Niwot, Colorado, USA) and its circuitry to manage the brightness of the LEDs, as well as the force-camera synchronization. Fig. 3.2 shows the design of the controller circuit [38]. The controller circuit is positioned outside of the TIP main body to decrease the hand-held probe's weight. Attachment B additionally describes the implementation and setup of the controller circuit.

The image data, together with the corresponding compression force data, are continuously transferred to the laptop during acquisitions and saved to its hard drive using TIP acquisition software. Finally, we use TIP calculation software to analyze the images and the force information and to quantify the mechanical properties of the embedded tumors.

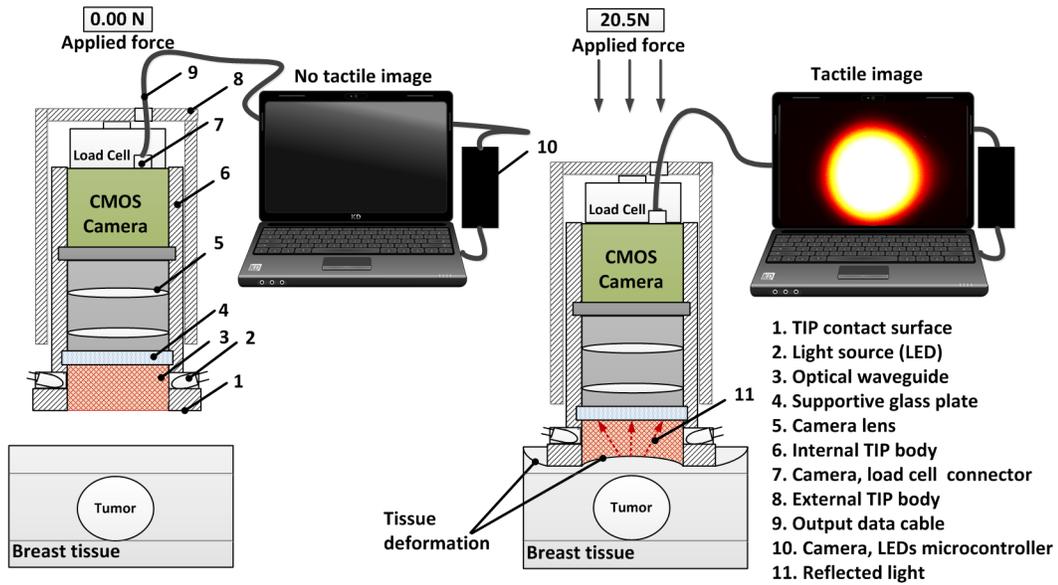


Figure 3.1: TIP principle and hardware design

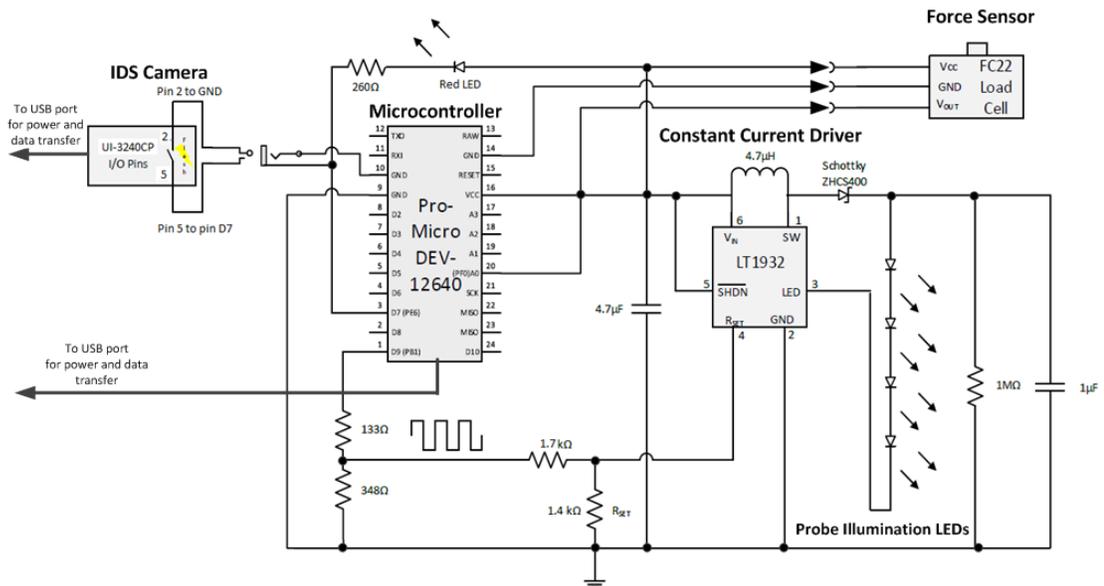


Figure 3.2: TIP controller circuit

3.3 Acquisition Software Design and Implementation

Acquisition software is used to control the acquisition through the controller unit, and to transfer the data from the camera and the force sensor to the laptop. Fig. 3.3 shows the TIP prototype connected to the laptop with the acquisition software graphical user interface (GUI) on the screen.

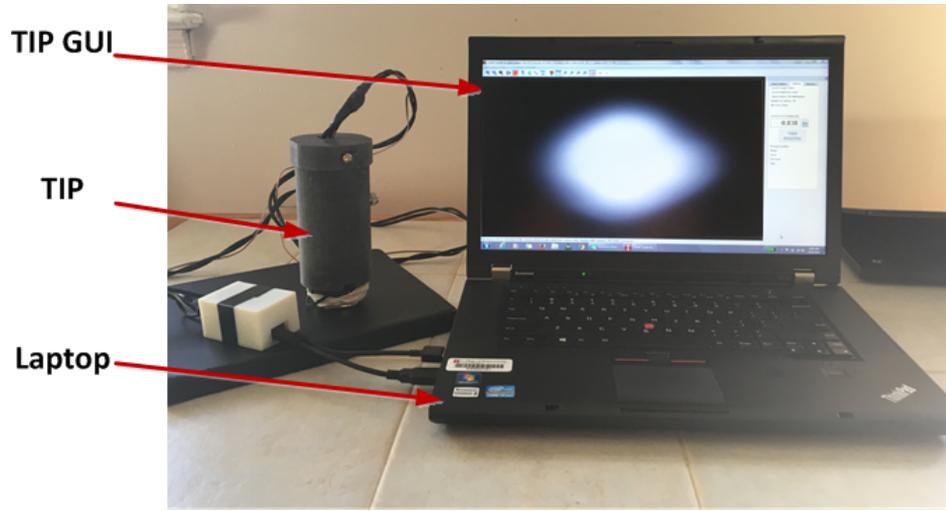


Figure 3.3: TIP prototype connected to TIP GUI

The software is based on the "ueyedemo" project bundled with the IDS uEye Linux SDK. The GUI is developed with Qt Creator and Qt 4.8, uses MinGW C++ Compiler and OpenCV 2.4.13 library. The initial version of the GUI was developed by our group based on the application script examples given by IDS Imaging Development Systems GmbH [38]. We developed it further to satisfy the operational requirements from doctors and to make the overall operation procedure easier.

The TIP acquisition software operation has three sequential steps: setup, capture, and results. The setup step allows the parameter to be set for the camera, controller, and force sensor, as well as the input of the information about the imaging session and the tumor. Fig. 3.4 shows the TIP GUI setup.

After the setup is complete, we move to the capture step. Information about the current sample and the acquisition session number are displayed. Dynamic force information is provided on the screen in real-time as a value in Newtons and a color bar. The Zero button



Figure 3.4: TIP GUI setup

can be used to offset the force sensor. When the Start button is pressed, a number of tactile images with the corresponding forces are captured and saved to the session folder on the hard drive. The number of images in one session can be set from 1 to 270. The system takes 20 images per second. The applied force level is visualized on the side panel as a dynamic status bar. The bar area covers the range of forces from 0 to 50 N. Also, colors are assigned to different levels of compression force. The compression of up to 5 N is not sufficient and will give the yellow color of the force status bar. Compressions between 5 N and 45 N will show the green color of the force status bar. This range is the typical acquisition range. Compressions over 45 N will be indicated in red. These forces are considered to be excessive.

Fig. 3.5 shows the TIP GUI capturing capabilities. After images from one compression session are saved, the software creates an additional folder with the false colored images for better visualization of the results.

Fig. 3.6 shows the folders with 50 tactile images and the corresponding force information from one compression session. Each image file name includes time and information about the acquisition. The image file names are in the following format:

Case#<number>_TumorEstimationModel_TumorLocation_Compression#_YYYY-MM-DD_<Time in ms>_TIP_im#<image number>_<force in N>N.bmp.

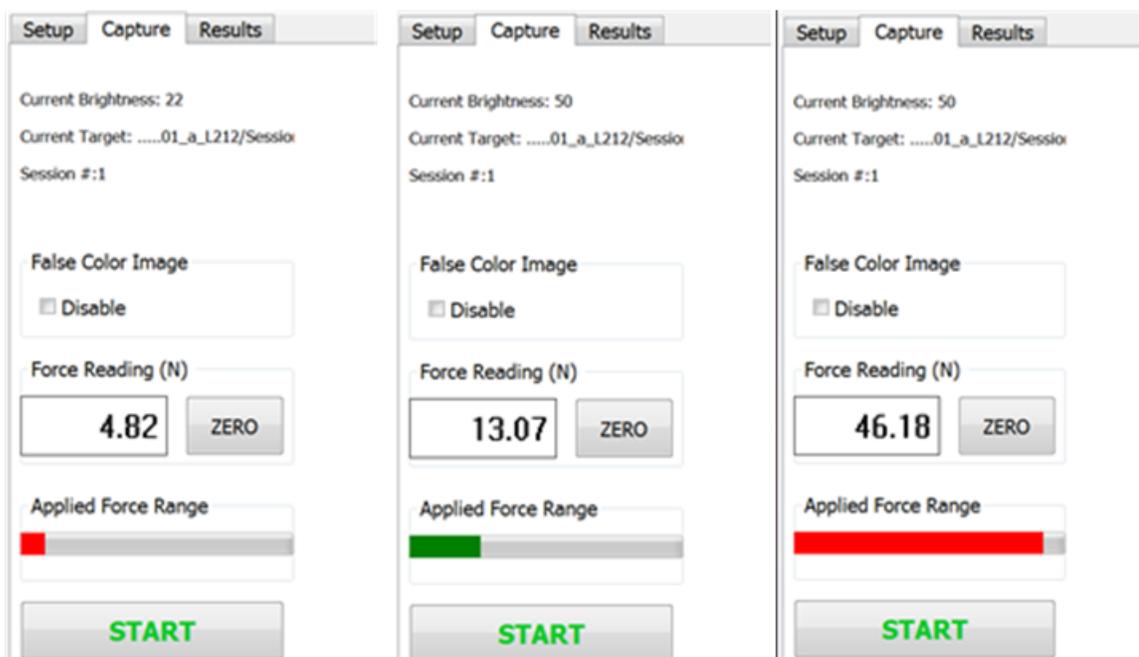


Figure 3.5: TIP GUI capturing capabilities

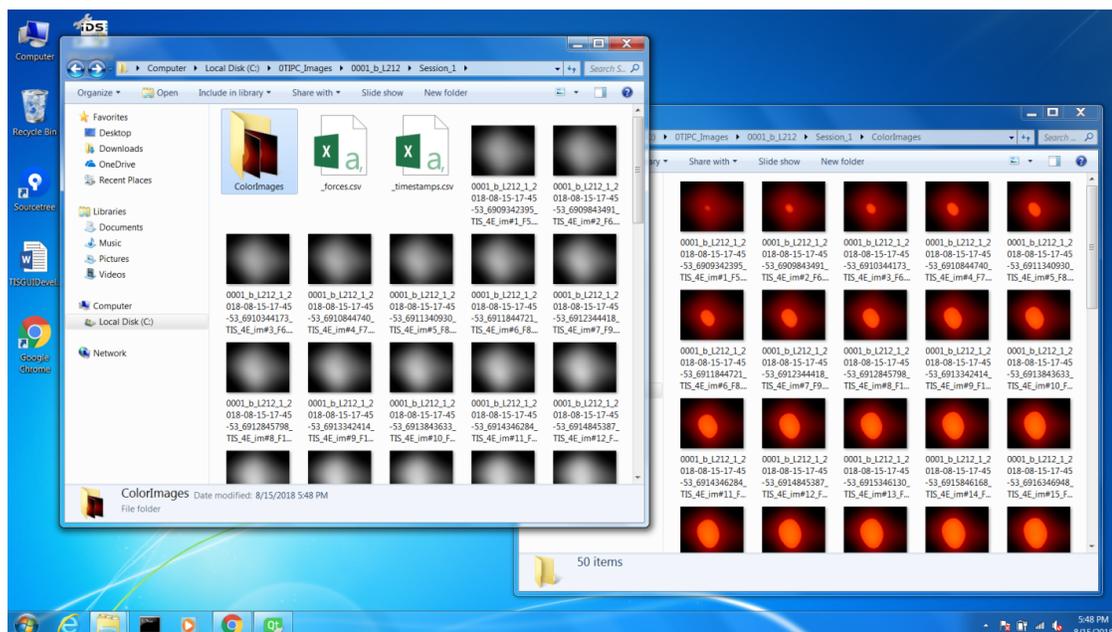


Figure 3.6: Example of the images saved during one TIP acquisition session

The final step is the results section. It is designed to present the operator with the preliminary calculations of size and stiffness of the tumor, the confidence in the calculations, as well as the preliminary malignancy score estimation in the range from 0 (benign) to 5 (malignant). This step should help the operator to evaluate the quality of the completed acquisition, and to decide if additional compressions are needed. Fig. 3.7 shows the TIP GUI Results Tab.

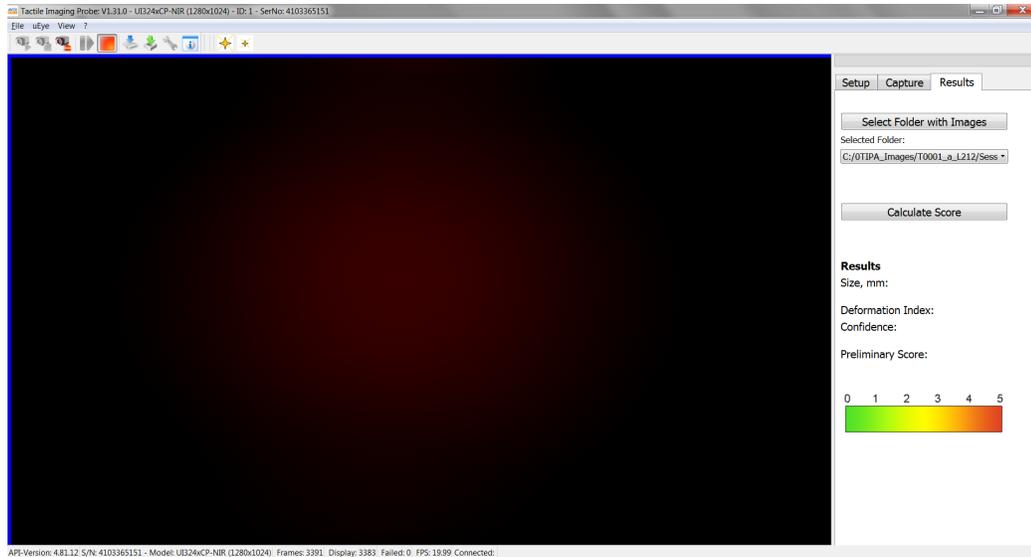


Figure 3.7: TIP GUI results tab

3.4 Mechanical Properties Estimation Algorithm

The mechanical properties estimation algorithm evaluates the size and stiffness of the tumors by analyzing the tactile imaging sets with the corresponding applied force information from TIP acquisition sessions [34, 35].

3.4.1 Numerical Simulation for TIP Sensing Element

We completed a numerical simulation for the TIP sensing element to understand how the contact surface of the sensor deforms during its compression against an embedded tumor. We used ABAQUS simulation software (ABAQUS Inc., RI, USA) to create the model.

To complete the simulation, we specified the components of TIP and multiple samples of tissue with tumors, their dimensions, and stiffness. We set up the sensing element dimensions ($23 \text{ mm} \times 23 \text{ mm} \times 12 \text{ mm}$) and specified its stiffness as Young's modulus value (27 kPa). We also specified the thickness of the glass plate (3 mm), which supports the sensing element's internal side. Next, we specified a range of breast tissue and tumor properties based on our breast mimicking phantom characteristics, which will be discussed in Chapter 4. We varied the size of the tumor from 5 mm to 30 mm and the stiffness of the tumor from 15 kPa to 465 kPa. We also varied the depth of the tumor within the tissue from 2 mm to 20 mm (the depth layer). Finally, we specified the range of applied force to TIP (5 N - 30 N), which caused the deformation of the polymer in contact with the tissue sample. Fig. 3.8 outlines our ABAQUS model setup.

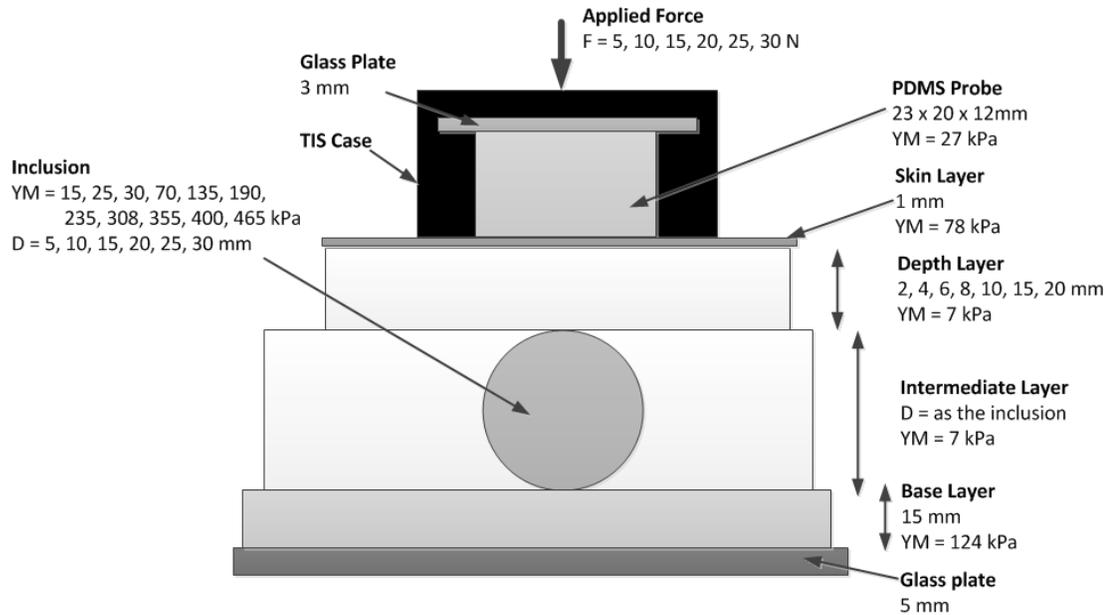


Figure 3.8: ABAQUS simulation setup

The results from the numerical modeling are summarized in Fig. 3.9. The three graphs describe the deformation effect from varying tumor size, stiffness, and depth, respectively, for the selected 26 N applied force. Each of these parameters has a unique effect on the formed tactile image, so it is possible to develop algorithms to estimate these properties from the sets of TIP images.

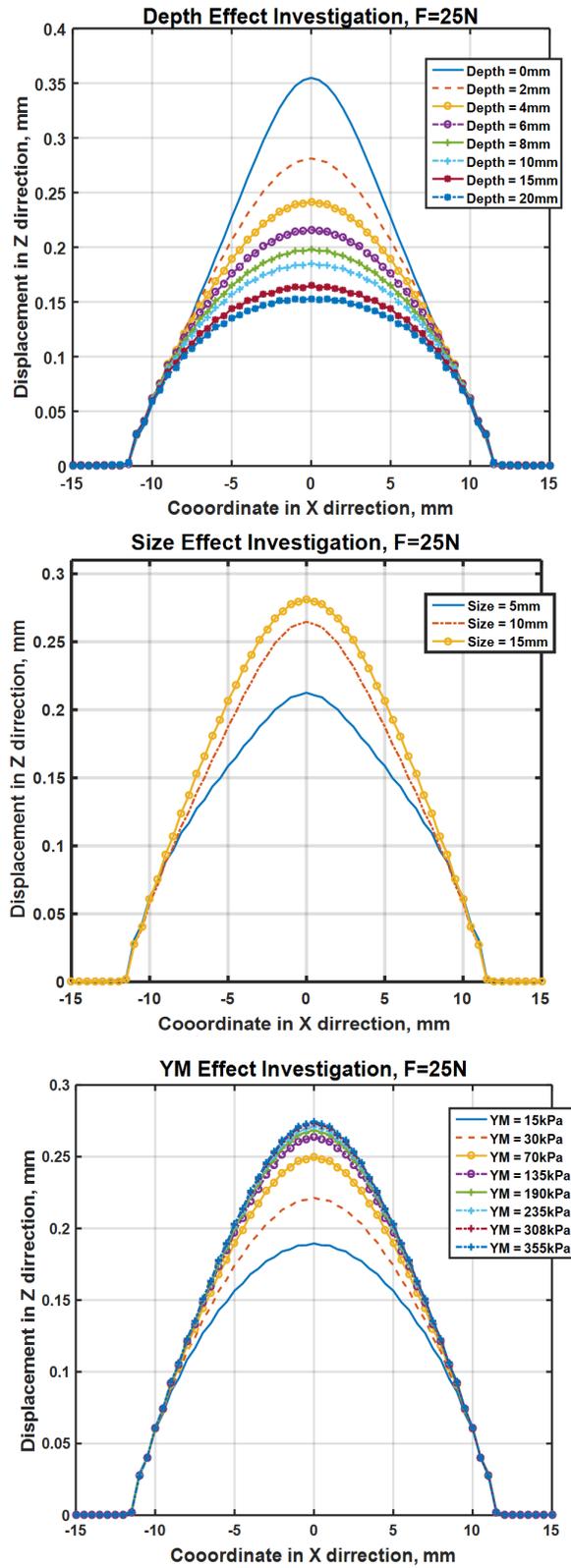


Figure 3.9: ABAQUS simulation results

3.4.2 Tumor Size Estimation: 3D Interpolation

To estimate the size of a tumor from a set of tactile images, we developed a three dimensional (3D) interpolation models [34]. The 3D interpolation method relates the size of the tumor as diameter measure, D , applied forces in compression, \vec{f} , and the sum of pixel intensities value on the corresponding tactile images, \vec{I}_p . We built several interpolation models to adjust the method for different tumor sizes and depths. To use the modeled surfaces for size calculation, we employ Equation (3.1). To prepare for the calculation, we collect applied force data, \vec{f} , from the force sensor, calculate the sum of pixel intensities from the corresponding tactile images, and use the estimated depth information from the TIP operator. Then we use Equation (3.1) to calculate the size of the tumor.

$$D(\vec{f}, \vec{I}_p) = \sum_{i=0}^{i=n} \sum_{j=0}^{j=m} p_{ij} f_i I_{pj}. \quad (3.1)$$

The size estimation model coefficients, p_{ij} , define the shape of the modeled surface. Indices n and m in Equation (3.1) denote the order of the polynomial for the size estimation. We developed a third order polynomial surface for the 3D interpolation (i.e., $n=3$, $m=1$), which fits our empirical data the most. We also build four 3D models for four different imaging scenarios. The four models are designed for large and deep inclusions, large and shallow inclusions, small and deep inclusions, and small and shallow inclusions. Fig. 3.10 shows an example of the model for large ($D > 12$ mm) and deep (> 10 mm) tumors. The size and depth thresholds depend on the application.

3.4.3 Tumor Region Stiffness Estimation

The extend at which the sensing element gets deformed during TIP compression against the embedded tumor within the tissue is called the deformation index. The amount of deformation depends on the tumor's size and stiffness, as well as on the stiffness properties of the surrounding tissue. When the conditions of the depth, size, and applied force are fixed, the stiffer tumor deforms the surface of the sensing element more than the softer tumor. Fig. 3.11 shows the surface deformation of the TIP sensing element in compression against stiffer inclusion vs. a softer inclusion.

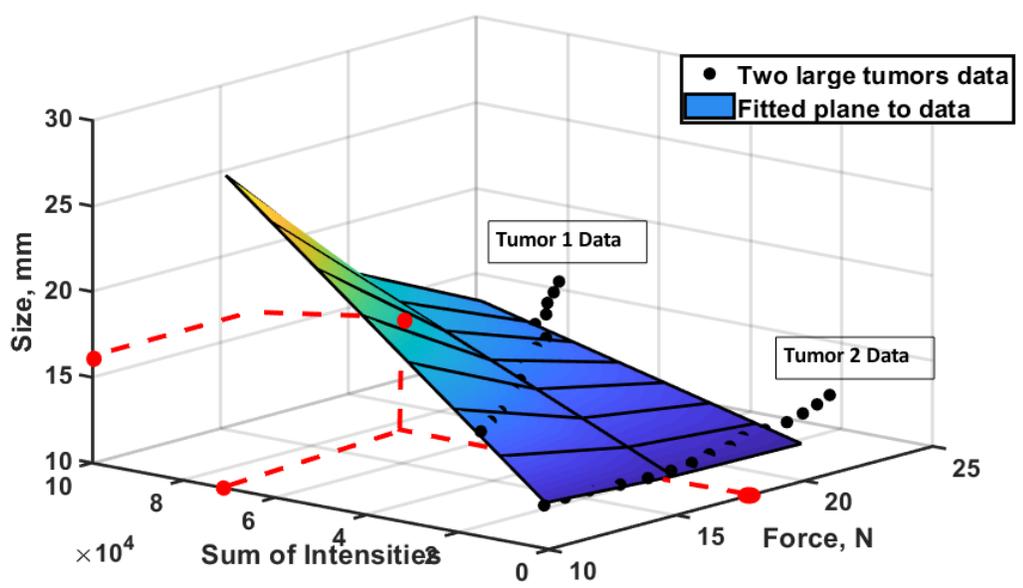


Figure 3.10: Example of a 3D interpolation model

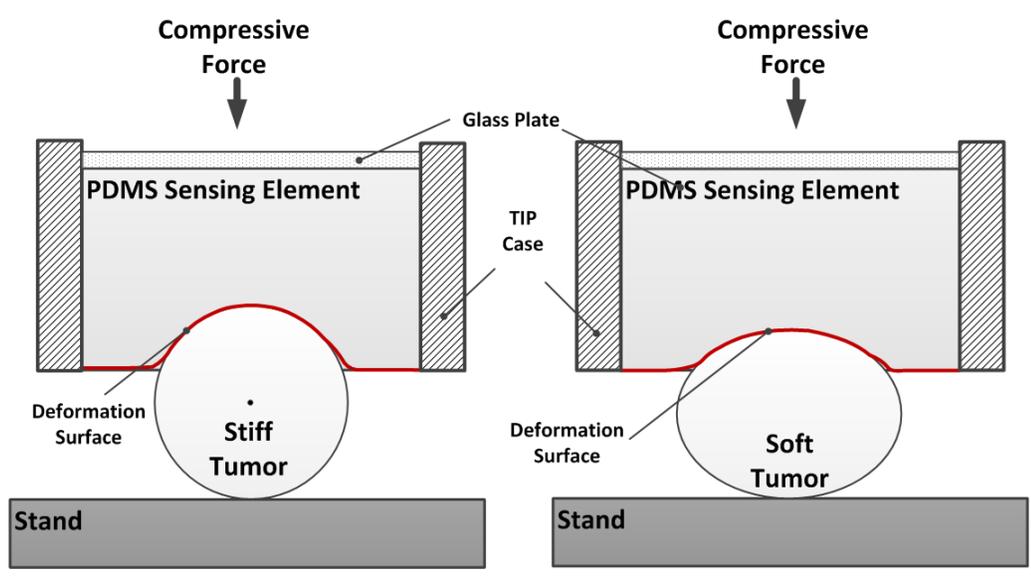


Figure 3.11: TIP sensing element deformation

Every image is defined as a multiple of 8-bit grayscale numbers (0 to 255) with the size of 1024×1280 pixel. Consider that the reference force f_{ref} is applied to the contact region (phantom), which produces a reference image M_{ref} , which is 1024×1280 matrix. The i^{th} tactile image, M_i , is obtained after applying a force f_i . The change in the pixel intensities can be represented as (3.2).

$$\Delta M = M_i - M_{ref}, \quad (3.2)$$

here $i = 1, 2, 3 \dots$.

The change in the force values corresponding to change in pixel intensity can be calculated as given below:

$$\Delta f_i = f_i - f_{ref}, \quad (3.3)$$

where $i = 1, 2, 3 \dots$.

The deformation index, DI , is the slope value of the graph plotted with the sum of pixel intensities ΔM_i vs. the change in the applied force Δf_i , which is calculated using (3.4).

$$DI_i = \frac{\sum_{l=1}^n \sum_{k=1}^m \Delta M_i^{l,k}}{\Delta f_i}, \quad (3.4)$$

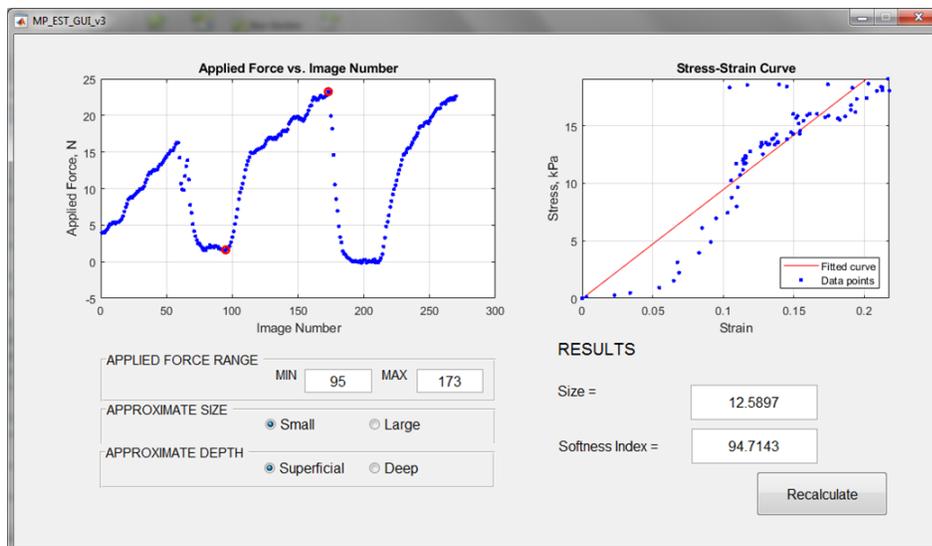
where the i^{th} tactile image has l rows and k columns.

3.5 Semi-automatic Mechanical Properties Estimation GUI

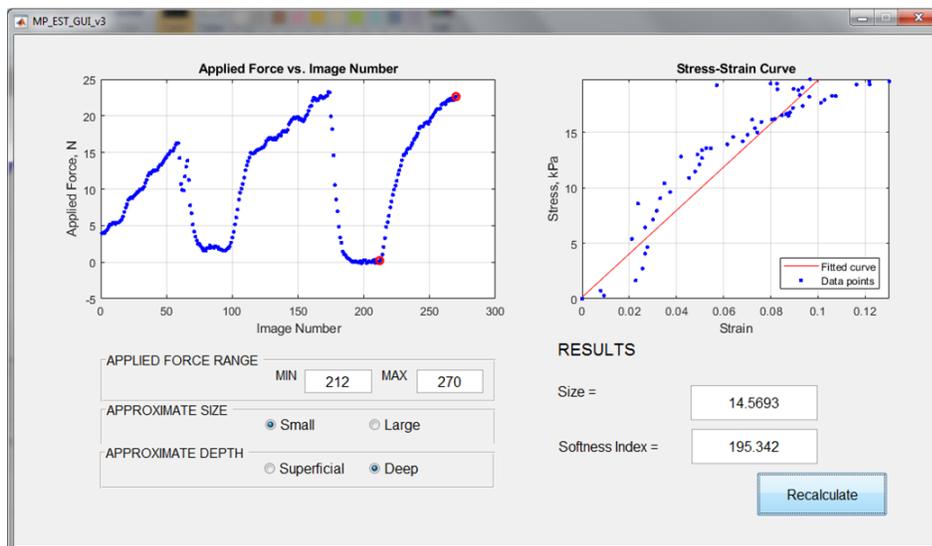
We developed a semi-automatic mechanical properties estimation graphical user interface using GUIDE application in MATLAB named MP_EST_GUI. TIP operator can use MP_EST_GUI to load one set of tactile images, eliminate outlier samples within the set, visualize the compression force data, automatically search for the best compression subset to be analyzed, plot the stress-strain characteristic for the selected subset, and calculates preliminary size and stiffness of the tumor. The GUI also allows the operator to define the subset manually and to make the estimations.

Fig. 3.12 shows MP_EST_GUI results for an imaging session with 270 images. There were three compressions within that acquisition session. In Fig. 3.12(a), the GUI automatically selected the best compression for analysis out of the three, which is between the

two red points on the left graph. Fig. 3.12(b) presents MP_EST_GUI results for the same imaging session, yet when we manually selected another subset for analysis and changed the depth estimation. The results show changes in size and stiffness estimation.



a) Automatic data selection



b) Manual data selection

Figure 3.12: GUI visualizations and calculations

The best compression identification step is completed by calculating the gradient of the force data array and searching for its peaks shown for a compression session in Fig. 3.13. The peaks indicate the end of each compression. Data to the left of a gradient peak location and before the next peak location correspond to an applied compression. If there were no

peaks found, it means only one compression was applied. Following, the algorithm selects the maximum and minimum points from each compression range. The decision on which compression to use during the analysis depends on the compression size in Newton. The largest compression has to be selected because it will give the best estimation performance. The algorithm will verify the compression length, which has to be at least 20 data points.

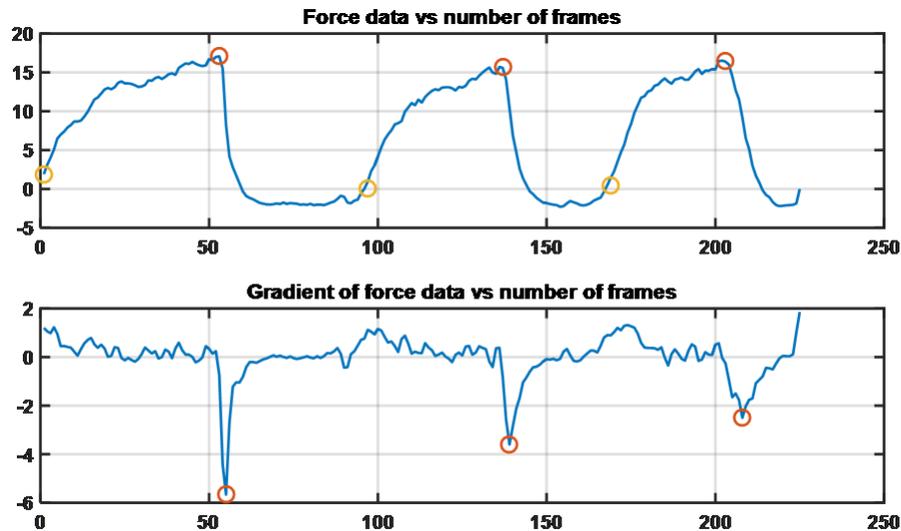


Figure 3.13: Automatic detection of compression ranges

The GUI is very useful for processing multiple TIP sessions, where we want to decrease the calculation time yet obtain quality results. The operator is able to process data in an interactive manner and visually verify the tactile imaging data.

3.6 Tactile Profile Diagram

Clinical data is highly multimodal due to the variety of sensors and diagnostic devices. The data can be one- or multi-dimensional; it can be stationary or dynamic, depending on how it was generated. Therefore, different methods are applied to the analysis of medical data. Time series data is one of the most popular data in the field. One patient in a hospital can generate a large amount of such data. The dynamic data can produce more useful information about the patient's condition than the static data; however, due to the computational limitations, not all of it will be analyzed and used for classification.

Clinical time series data analysis was explored by many researchers [39, 40]. Recently, researchers directed their attention to the processing of the series of data with deep learning tools by visualizing time series data via pattern images [41, 42]. The reconstructed patterns from time-series data are given names such as Gramian Angular Summation/Difference Fields and Markov Transition Field [43].

One of the fields, which explores the dynamic properties of time-series signals for a long time, is the speech processing field. In speech processing and recognition, it does not make much sense not to process sounds over a wide range of frequencies and time [44]. Spectrograms are the way of visualization speech signals [45, 46]. A spectrogram is a visual representation of the time-varying frequencies of an acoustic signal. It reflects on acoustic energy stored within the signal over a period of time [45]. Spectrogram can be presented as grayscale or a color plot Fig. 3.14. Human hearing relies on signals from the cochlea of the inner ear. These signals are functionally similar to how spectrograms are developed [47], so it makes sense to represent audio signals via spectrograms [44].

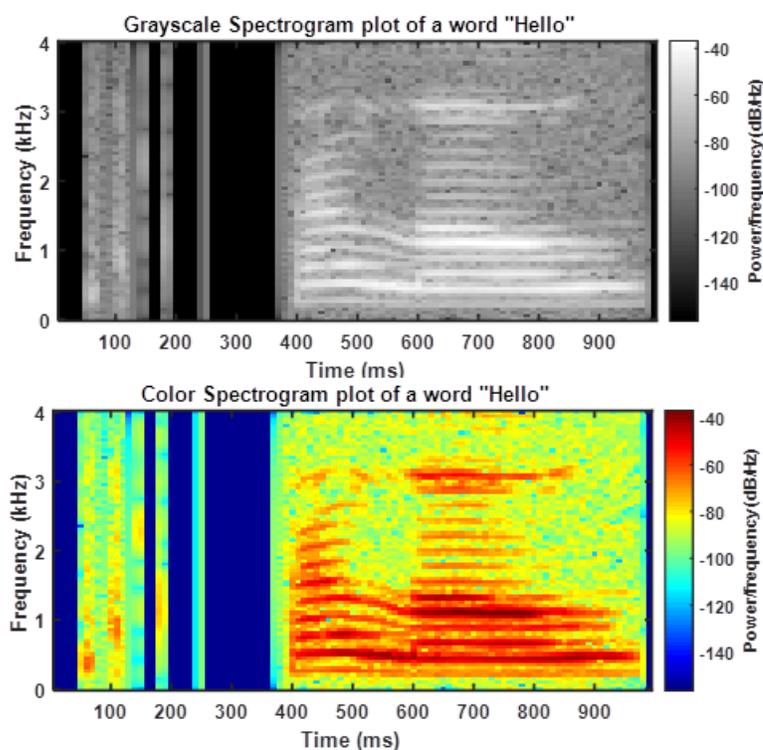


Figure 3.14: Example of a speech signal representation via a spectrogram

Human touch sensation is based on signals from tactile sensors throughout our body, which are processed in the parietal lobe of our brain [48]. The tactile signals are processed in our brain very similarly to the visual signals [49]. Tactile Imaging Probe (TIP) is developed to measure the tactile properties of touched tissues [34, 50]. So far, we have described how TIP and its algorithms estimate the size and stiffness of embedded tumors from the series of tactile images.

Tactile images are dynamic time series data that contain a large amount of information about the imaged tissue. We can extract the application-specific information (tumor's stiffness and size) from the data by creating pattern images for each tactile imaging set.

We want to use the dynamic tactile information as a temporal signal from TIP acquisition sessions to improve the tumor characterization. With that goal in mind, we propose a method of combining sets of multiple tactile images into one representative diagram, called the Tactile Profile Diagram (TPD).

TPD is a complex tactile image, which fuses the information from a set of 50 single tactile images and gives a possibility of using the dynamic information from tactile imaging with TIP.

3.6.1 Tactile Profile Diagram Construction

One raw TIP image characterizes the stiffness distribution of a region for an instantaneous value of the compression force [34]. Here we present a method to construct a Tactile Profile Diagram (TPD) as a representative pattern image from a set of TIP images.

TPD is a compact pictorial representation of the relative stiffness and size of the imaged tumor. TPDs should be constructed using imaging data from the same range of applied forces during TIP acquisition. The range of forces in our experiments is from 15 N to 45 N. The method to create a TPD from a set of tactile images is as follows.

We acquire a set of images with TIP by compressing it on the region with a tumor. The number of raw tactile images, $I_i(x, y)$, in a set is $i = 1, 2, \dots, N$ (in our experiments $N = 50$). Variables x and y are the horizontal and the vertical coordinates respectively of a pixel within an image I_i . We apply an averaging filter with a non-overlapping sliding window of size 10 pixels \times 10 pixels to each $I_i(x, y)$ to reduce the level of white noise within the

images, decrease image size without the loss of tactile information, and improve the speed of computations. The pixel values in the created reduced image, $R_i(m, n)$, corresponding to the average values of pixel intensities in the window at each step. The number of reduced images in the set, N , is the same as the number of original images ($N = 50$). The size of the reduced images is 103 pixels (m) \times 128 pixels (n). The reduced images are the more compact copies of TIP raw images.

Each tactile image, I_i , has a corresponding compression force value. The \vec{f} is the vector of forces of size $N \times 1$. Similar to the pre-load step during Instron tests [51], we select a reference force, f_{ref} , and its corresponding reference reduced image, $R_{ref}(m, n)$, from the set to account for the imperfections at the test tissue surface. Empirically, we chose 5 N as f_{ref} . To obtain the vector of the change in compression force, Δf_i , we subtract the reference force value from the force vector, following 3.3.

Subsequently, we complete pixel-wise subtraction of the reference reduced image, $R_{ref}(m, n)$, from all reduced images, $R_i(m, n)$, in the set to create an image set ΔR_i , which represents the change in tissue deformation under compression. These images describe the deformation of the silicon sensing element of the TIP.

$$\Delta R_i(m, n) = R_i(m, n) - R_{ref}(m, n), \quad (3.5)$$

Next, we find the maximum intensity value, ΔR_{max} , present in the ΔR_i images set. Then we subtract each pixel value in ΔR_i images from ΔR_{max} to get the change of deformation images, ΔW_i , of the tissue with inclusion, and not of the sensing probe's material.

$$\Delta W_i(m, n) = \Delta R_{max} - \Delta R_i(m, n), \quad (3.6)$$

To construct a Tactile Profile Diagram, we calculate Young's modulus index values, $YMI(m, n)$, which is the stiffness estimation parameter for each pixel location in ΔW_i (Fig. 3.15). We mimic the definition of Young's modulus and calculate YMI in a pixel location (m, n) as a slope of the compression force over an area (the compression change Δf_i) vs. the deformation change due to the compression (the change in deformation ΔW_i for the tissue with inclusion). To calculate the slope in each pixel location (m, n), N data points will be available from a TIP set. Finally, a TPD is obtained by plotting $YMI(m, n)$

for $m = 1 \dots 103$ and $n = 1 \dots 128$ (Fig. 3.15). If Δf_i is divided by the contact area and ΔW_i is correlated with the strain, then $YMI(m, n)$ is directly related to the young's modulus of the tissue at any given point at (m, n) .

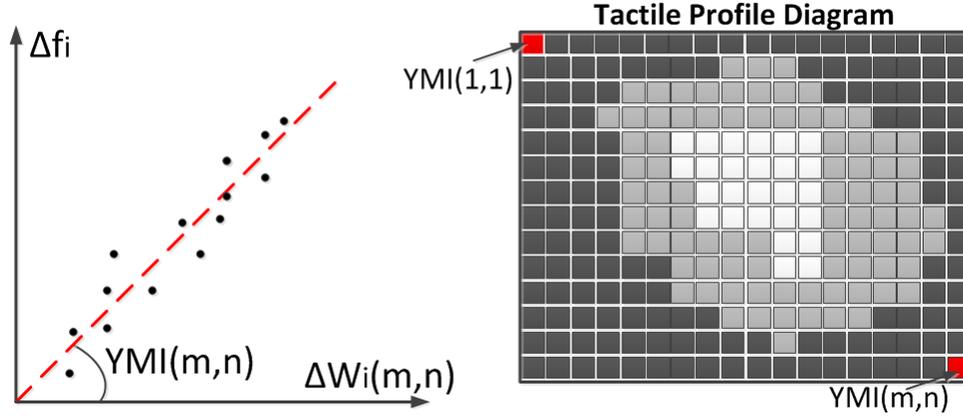


Figure 3.15: Tactile Profile Diagram generation

3.6.2 Mechanical Properties Estimation Algorithm from Tactile Profile Diagram

To use Tactile Profile Diagrams for mechanical property estimation, the diagrams have to be segmented due to the high level of hardware-related noise in their perimeter. Then size and stiffness can be estimated from TPDs directly using the methods described in this section.

TPD Segmentation

Similar to the Canny edge detection method [52], we utilize the directional gradient of TPD to find a region of interest for calculations. Because an image gradient is the directional change of the pixel intensity in an image, one can use it to detect intensity changes associated with the inclusion region vs. the changes due to the perimeter noise. We use MATLAB 2019a function $imgradient(YMI, 'sobel')$ to create a gradient magnitude image, $G_{mag}(m, n)$, and a gradient direction image, $G_{dir}(m, n)$, from a TPD [53]. Next, we create a composite image, $G(m, n)$, from a difference between $G_{mag}(m, n)$ and $G_{dir}(m, n)$ using $imfuse(Gmag, Gdir, 'diff')$. We semi-manually select the region of interest in the center of $G(m, n)$ image by following the edges of the circular region in Matlab using $drawassisted$

function. The selected region corresponds to the inclusion indentation into the probe during the TIP test. After applying the created with *drawassisted* function mask to $G_{magS}(m,n)$ image, the segmented $G_{magS}(m,n)$ has zero-valued pixels outside of the selected region, and non-zero pixels inside it. The segmentation method steps are shown in Fig. 3.16.

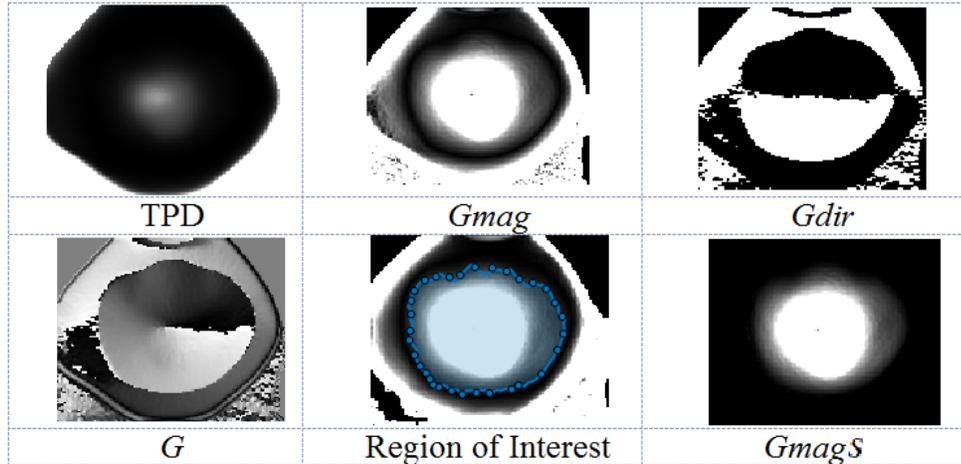


Figure 3.16: Tactile Profile Diagram segmentation

Tumor Size Estimation

The tumor's size is estimated by summing the number of non-zero pixels within the segmented image $G_{magS}(m,n)$. Next, this value is related to the tumor's size with the use of a linear regression model. Use (3.7) to find the size of the inclusion, D , in mm.

$$D = aN_p + b, \quad (3.7)$$

where N_p is the number of nonzero pixels in the segmented area, $a = 0.0019$ and $b = 7.1028$ are the empirical coefficients. The coefficients are developed by linear interpolation to the data points of known inclusion sizes vs. N_p from the corresponding $G_{magS}(m,n)$. Fig. 3.17 shows steps for size calculation.

Tumor Region Stiffness Estimation

Stiffer the inclusion, the larger is the change in TPD pixel intensities, and the greater are the pixel intensities of the segmented gradient magnitude image G_{magS} . Therefore, we define

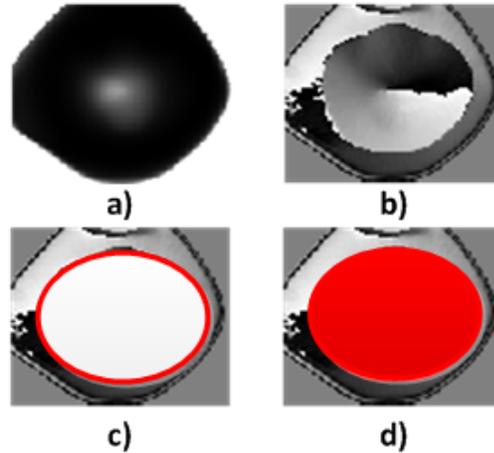


Figure 3.17: Tumor size calculation from a TPD

a) Load a TPD; b) calculate $G_{mag}(x, y)$; c) detect edges of the tumor; d) calculate the number of pixels within the selected region inside the edges

Stiffness Index, SI , as the measure of the tumor's stiffness based on the collection of local stiffness measurements (YMI), where YMI is directly related to the Young's modulus of the tissue at any given point on the corresponding TPD 3.6.1. The calculation of SI follows (3.8).

$$SI = \frac{\sum_m \sum_n G_{magS}(m, n)}{N_p}, \quad (3.8)$$

Therefore, in one number, SI represents the sum of all gradient magnitudes divided by the number of nonzero pixels. Stiffer the inclusion, the intensity changes will be larger, and SI will be larger.

3.7 Classification Methods

In this section, we develop classification methods to extract desired information about the imaged tumor from Tactile Profile Diagrams. We will use the results of the classifications to calculate the Tactile Index. Additionally, classification results will improve tumors' mechanical property calculation using Tactile Imaging Probe algorithms described earlier in this chapter.

Here we provide a method to apply pattern recognition techniques, such as Convolutional Neural Network, to Tactile Profile Diagrams. Using the method, we estimate the depth, size, and stiffness of tumors from TPDs. We discuss in length the method and the designed classification models.

3.7.1 Overview of Machine Learning Classification Methods

Artificial intelligence (AI) is applied to a wide variety of research topics [1]. It is used in engineering, medicine, and science to improve people's life.

The challenge presented to AI is to solve the problems, which humans can perform, yet found difficult to explain how it was done. The computers' ability to mimic human vision and speech are examples of such problems. The way to enable computers to perform such tasks is to let them learn from the smaller and less complex tasks, which can be easily described [1].

A complex task will include many layers of simpler concepts built and defined one over another. This idea of multiple layers is translated into the concept of AI deep learning (DL). A person uses years and decades of experience and knowledge to complete everyday tasks. This knowledge differs from person to person. It is difficult to express knowledge in a formal way, and it is difficult to teach it to computers. The attempts to define knowledge about the world proved to be very challenging [54]. On the other hand, machine learning (ML) is a modern tool in AI, which allows computers to learn concepts from data directly.

There are three main types of ML algorithms, depending on the way how they learn: supervised, unsupervised, and reinforced learning. A supervised ML algorithm learns from a data set with labels (annotations), which means the model has input and output for training. Examples of supervise learning algorithms include support vector machines (SVM), k-nearest neighbors (KNN), and decision trees. An unsupervised algorithm learns the feature representation from the data without labels, which means that the model is given only an input. Examples of unsupervised learning algorithms include the principal component analysis (PCA), and k-means clustering. The reinforcement learning algorithm interacts with the environment and learns through a rewards system [1, 55].

Simple ML algorithms, such as logistic regression or naive Bayes, are very dependent

on the data representation. The data is represented as a set of features. So the choice of features influence the results of the algorithms and is somewhat subjective. While it may be easy to create a set of features for one application, it may be challenging for others. It is possible to use ML to discover the feature representation for a given task. This concept is called representation learning, and it currently outperforms manual feature learning [1].

Some of the algorithms were inspired by our knowledge about living systems. For example, artificial neural network algorithms were designed to mirror brain structure and functions to learn. The network systems are the combinations of layers, where each layer is composed of multiple nodes (neurons). The neurons transmit information ("fire") if the given condition is met [55]. Autoencoders are successful neural network algorithms, which rely on representation learning. One of the main goals in the design of these algorithms is to be able to explain the factors of variation within the data, yet it is not always easy to do.

In recent years, this problem with explaining variations within data is solved by introducing deep learning. Deep learning algorithms were inspired by neural networks, and are able to represent complex concepts as sets of simple representations presented in multiple layers. Fig. 3.18 explains the way how deep learning represents an image of a person in simple concepts. The DL system learned edges in the image first, then the corners and contours were learned based on the edges. Next, the object parts were learned based on the previous knowledge of corners and edges. Finally, the object on the picture was classified as a person.

3.7.2 *Convolutional Neural Network (CNN) Classification*

Convolutional Neural Network (CNN) is a special type of neural network designed to work with the data presented in grids: time-series (1D data), and images (2D data) [1, 56]. CNN is a neural network that uses convolution operation instead of matrix multiplication in at least one of its layers [1].

The structure of the network was inspired by the visual cortex hierarchy and presented by Fukushima [57]. Since then, it was applied for visual object recognition and image classification tasks with good performance [58].

One CNN layer usually contains three steps to complete. The first step is convolution,

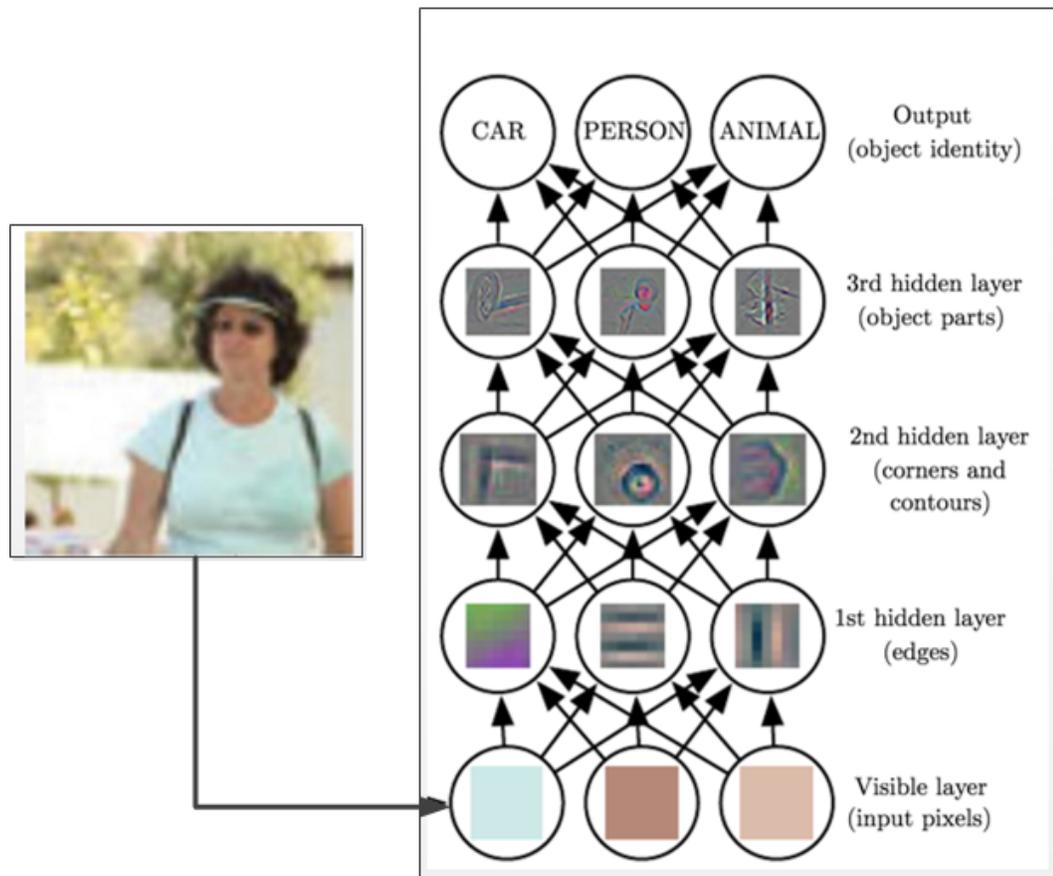


Figure 3.18: Deep learning representation [1]

which produces a set of linear activations. Followed by the detection (perception) step, where the linear activations are combined with nonlinear activation functions. Finally, the pooling step makes the output invariant to small translations [1].

Convolution

To explain the idea behind convolution, let us introduce a sensor system for an airplane location. This example was inspired by Goodfellow's introduction to the convolution operation [1]. The sensor system will have the output $z(t)$, which is the location of the airplane in time t . To have the value of x at any time t , z and t have to be real values.

To reduce the amount of noise in the sensor output, we compute a weighted average of several output values, assigning higher weights to the newer readings. To implement that, we apply a weight function $w(a)$ at each time instance with a representing the age of the signal measurement. The calculated s function (3.9) will correspond to the clean sensor's output and is called convolution. Asterisk sign ($*$) is a common notation for the convolution operation.

$$s(t) = \int z(a)w(t-a)da = (z*w)(t). \quad (3.9)$$

The convolution operation is used not only to implement the weighted average, yet it is very useful for many other applications. In the study area of convolutional networks, $z(t)$ is called the input, w is called the kernel, and the feature map is the output.

Because computers are not capable of processing continuous data and the sensor output are discrete readings in time, we need to define the discrete convolution function (3.10).

$$s(t) = \sum_{a=-\infty}^{\infty} z(a)w(t-a)da = (z*w)(t). \quad (3.10)$$

To apply convolution to an image I , we need to use a two-dimensional kernel W as well. The two-dimensional convolution will look as follows.

$$S(i, j) = \sum_m \sum_n I(m, n)W(i-m, j-n) = (I*W)(i, j). \quad (3.11)$$

The commutative property of convolution allows us to write the following.

$$S(i, j) = \sum_m \sum_n I(i-m, j-n)W(m, n) = (W*I)(i, j). \quad (3.12)$$

Very often, in Convolutional Neural Network implementation, cross-correlation is used instead of convolution. Cross-correlation is the same as convolution but does not involve the flipping of the kernel [1].

$$S(i, j) = \sum_m \sum_n I(i+m, j+n)W(m, n) = (W * I)(i, j). \quad (3.13)$$

Fig. 3.19 demonstrates the convolution operation via cross-correlation [1].

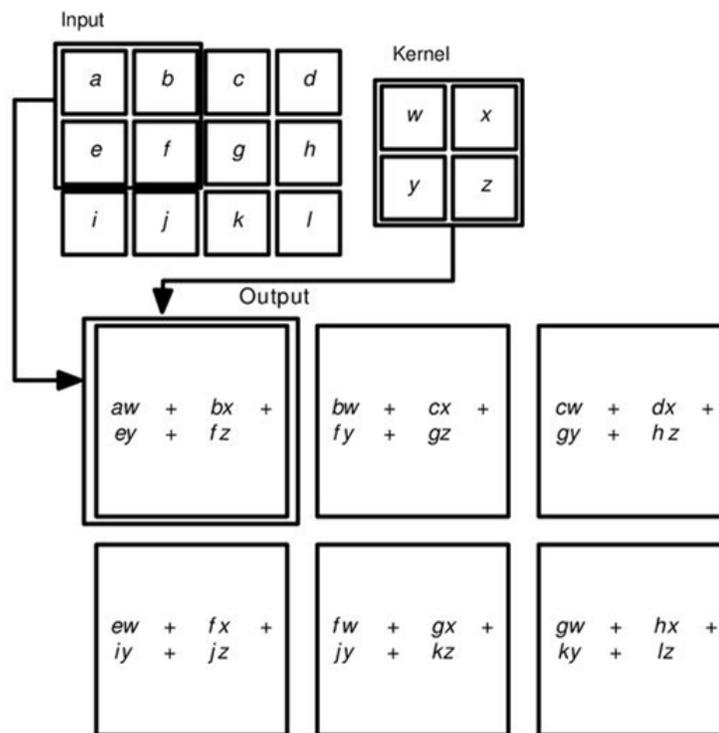


Figure 3.19: Convolution operation via cross-correlation

Applying convolution to an image creates a map of features of that image. Some types of input image transformations, such as scale or rotation, will not be adequately handled by convolution [1]. Therefore, the augmentation of training data can improve the performance of a CNN model.

Detection

Detection is done by using a detection algorithm on the output of a convolution step. The detection functions are also called activations.

Perceptron. The most simple detector is the perceptron. It has two possible outputs: yes (1) or no (0), and can be defined as follows [55].

$$f(x) = \begin{cases} 1 & wx + b > 0 \\ 0 & \textit{otherwise} \end{cases} \quad (3.14)$$

where x is input signal, w are weights, b is bias, and wx is the dot product $\sum_{j=1}^m w_j x_j$.

Unfortunately, the perceptron algorithm has serious flaws. It does not allow for a neuron to learn gradually, only with sharp changes. Therefore other activation functions were introduced.

Sigmoid. The sigmoid activation function defined as

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (3.15)$$

The function is continuous and gives the neuron ability to learn in tiny steps [55].

Hyperbolic Tangent. Similar to sigmoid function is the tanh activation, which is defined as

$$f(x) = \tanh(x). \quad (3.16)$$

ReLU. Another activation function is called a rectified linear unit (*ReLU*). It gained great popularity in recent years due to its practical compatibility with different applications. *ReLU* is defined as

$$f(x) = \max(0, x). \quad (3.17)$$

Leaky ReLU. The leaky rectified linear unit (*Leaky ReLU*) is a modification of *ReLU*, yet *Leaky ReLU* allows small learning when the neuron is not activated. *Leaky ReLU* is defined as

$$f(x) = \begin{cases} x & x > 0 \\ ax & \textit{otherwise} \end{cases} \quad (3.18)$$

where a is the slope parameter.

ELU. Another modifications to the linear rectified unit function is the exponential linear unit (*ELU*) [59].

$$f(x) = \begin{cases} x & x > 0 \\ a(e^x - 1) & \textit{otherwise} \end{cases} \quad (3.19)$$

where a is a hyper-parameter to be tuned.

All of the discussed activation functions are plotted in Fig. 3.20.

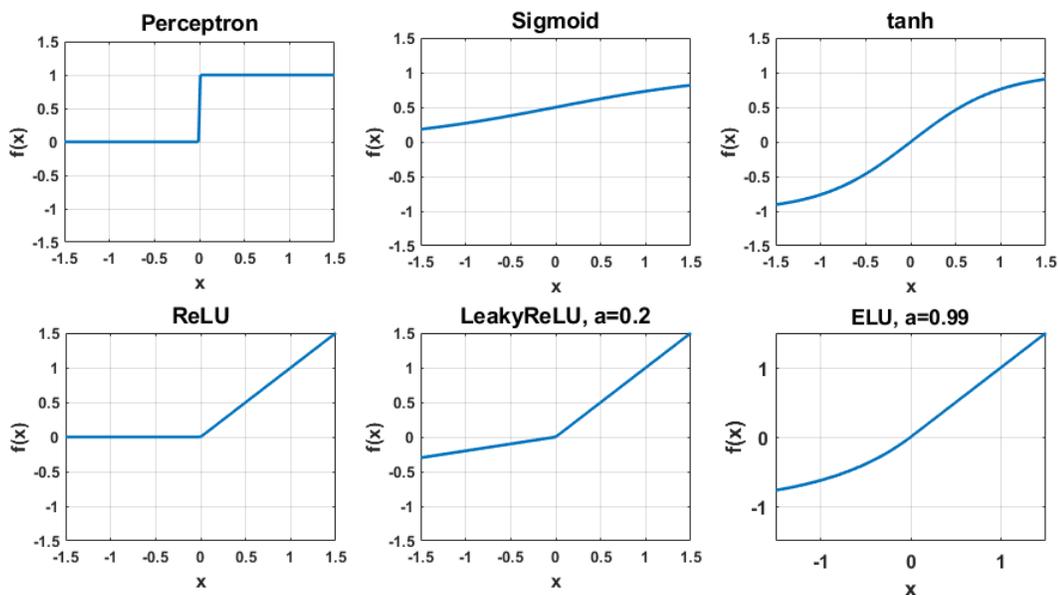


Figure 3.20: Common activation functions

Pooling

Pooling is an essential part of a CNN model, which was inspired by the complex structure of brain visual cortex cells [58]. Small translational variations (shifts of pixels) in the input image can be compensated by using pooling. It allows us to find a particular feature on the image without caring much where precisely the feature is on the image [1].

A pooling replaces the feature map pixels with the generalized neighboring pixel statistics [1]. There are many ways to implement pooling, yet the most common is the *Max pooling*, which replaces the pixel value on the feature map with the maximum pixel value of its neighboring pixels [60]. Fig. 3.21 illustrates how the *Max pooling* works [61].

Model Compilation

There are three main steps required to compile a CNN model [55]. All three of them are briefly summarized here.

Image Matrix				Max Pool	
2	1	3	1	2	4
1	0	1	4	7	9
0	6	9	5		
7	1	4	1		

Figure 3.21: *Max pooling* operation

Optimization

The optimizer is an algorithm used to update the model training weights. The most common algorithm used for optimization is the gradient descent algorithm [55]. The most commonly used optimizers are *Adam* and *Adamax* optimizers introduced by Kingma [62], as well as RMSprop [55].

A large amount of training data is necessary for the good generalization of deep learning. To learn, a model searches for the combination of parameters that minimizes the objective or loss function for the given training dataset. Because the amount of data is very large, the model training involves drawing random subsets (batches) from the training set and learning the parameters from one batch at a time. One training epoch is completed when the model finished learning in batches from the available training data [63]. In addition, deep learning methods heavily rely on the fast version of the gradient descent algorithm - the stochastic gradient descent algorithm (SGD), to be able to process a large amount of data [1].

Objective function

The objective functions or loss functions are used by optimizers with the goal of minimization of the loss during the weights update [55]. The loss function should be meaningful to the application because it will define how well the model will perform on the data during tests [63]. The most commonly used objective functions are as follows.

MSE optimizer uses the mean square error calculation between the prediction and the

true value. The function is expressed as following [55].

$$MSE = \frac{1}{n} \sum_{i=1}^n (\gamma - Y)^2. \quad (3.20)$$

where γ is a vector of n predictions, and Y is the vector of n observations.

The binary cross-entropy optimizer employs the logarithmic function and is used in binary label prediction [55].

The categorical cross-entropy optimizer is a logarithmic loss for multiple classes. The optimization is used for the multiclass classification and is described as follows [55].

$$L_i = \sum_j t_{i,j} \log(p_{i,j}) \quad (3.21)$$

where $t_{i,j}$ is the target, and $p_{i,j}$ are predictions.

Model Evaluation

After a CNN model is compiled, it has to be evaluated. There are several common evaluation metrics: recall, precision, and accuracy [55].

Recall is the number of correct positive predictions divided by the number of all samples that had to be positive.

Precision is the number of correct positive predictions divided by the number of positive predictions by the classifier.

Accuracy is defined as the total number of the correct predictions to the total number of the given predictions.

Evaluation of a CNN model should not be done on the same data it was trained on, yet on a separate validation set. If the model is trained and evaluated on the same data, the model will become overfit. It means the model will learn the training data very well and will classify it with high accuracy, yet will have a poor performance on the data not from the training set [63]. During multiple epochs of learning, the model begins to tune to the validation data as well. Therefore, as a final step after tuning the model, it needs to be tested on the independent dataset, which does not include data from training or validation [63].

It is important to avoid overfitting. On an accuracy *vs.* epochs plot, an overfit model will show a linear increase of training accuracy until nearly 100%, while the validation accuracy

will stop improving at some point. On a model loss *vs.* epochs plot, the overfit model's validation loss reaches some value and then stops decreasing or even starts to increase, while the training loss keeps linearly decrease to approach 0 [63].

The common methods to prevent overfitting of a model are data augmentation, the addition of a Dropout layer before Dense layers, or the use of regularization techniques [63]. Data augmentation techniques increase the number of representative samples in a dataset by generating new samples from the available data. If the initial dataset is small, data augmentation may not be sufficient. In this case, the addition of a Dropout layer before Dense layers may decrease overfit. A Dropout layer allows to randomly drop the number of training parameters before the classification step. Regularization can be added during model training to limit the number of filters per convolution layer or the depth of the network itself [63].

3.8 Classification of Tactile Profile Diagrams

Classification of Tactile Profile Diagrams, which are the one-image representations of multiple TIP images, is designed to extract the tumor's depth, size, and stiffness information. TPDs carry the compact fused information about tissue inclusion mechanical property from multiple TIP images acquired under different compression forces. The acquisition force information is incorporated into the TPD image in an application-meaningful way. Classification of TPDs can help to characterize the mechanical properties of tumors and to distinguish malignant from benign ones [33].

TPDs are different than typical images. They are rough estimates of tumors in the form of pattern images. We use a special type of deep learning, Convolutional Neural Network, which is used for image pattern recognition and classification for classification of Tactile Profile Diagrams.

In this work, we will use three CNN models to classify imaged tumors depth (deep or shallow), size (small, medium, or large), and stiffness (soft or stiff). These classification results in the form of class probabilities will be used for Tactile Imaging Probe Index calculation for the breast cancer risk assessment and to aid tumors' size calculations with TIP

algorithms [34]. Fig. 3.22 presents the described TPD classification steps.

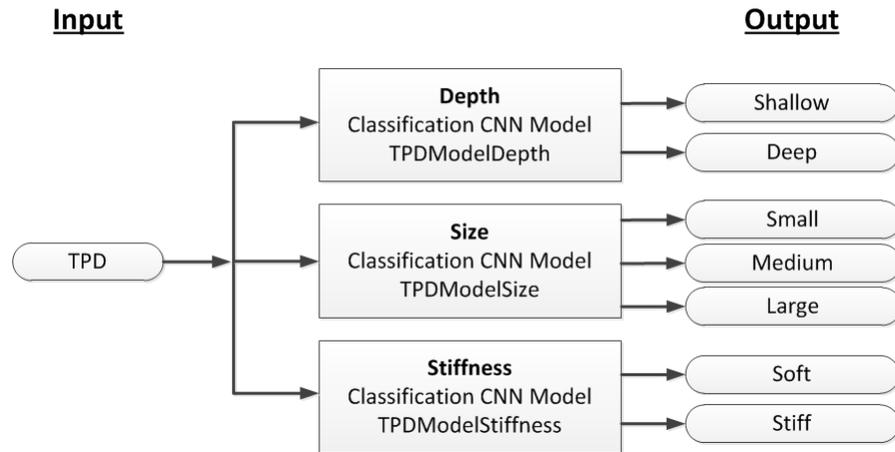


Figure 3.22: TPD classification steps

3.8.1 Depth Classification Model

We designed a CNN sequential model without a feedback loop for tumor depth evaluation, with three convolution layers, three *Max pooling* layers, and activation functions for the feature learning part. The model is designed as a binary classifier to distinguish shallow and deep tumors.

Fig. 3.23 shows the structure of the depth classification model, and details on the model are given in Fig. 3.24. The classification part of the model includes four dense layers with *ReLU* and *ELU* activation functions and *Dropout* layers of 0.3 and 0.2. The final fully connected layer has *Softmax* activation function and two nodes, corresponding to the shallow or deep classes. We employ *Adamax* optimizer, and *accuracy* as the metric for the model performance evaluation during training. The Python script for the model development is given in Appendix E.

The data for training and validation of the model is developed from tactile phantom imaging data combined into Tactile Profile Diagrams dataset for depth classification. The dataset development and model training are described in Chapter 4. With the sufficient training data and the modified final layer, the model can output multiple depth ranges (classes) if required by the application.

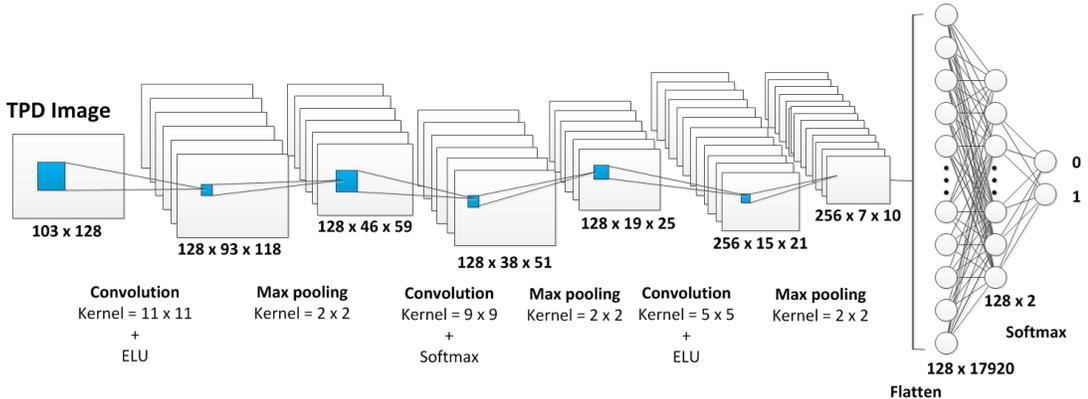


Figure 3.23: TPD depth classification model (TPDModelDepth)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 93, 118, 128)	15616
activation (Activation)	{None, 93, 118, 128}	0
max_pooling2d (MaxPooling2D)	(None, 46, 59, 128)	0
conv2d_1 (Conv2D)	(None, 38, 51, 128)	1327232
activation_1 (Activation)	{None, 38, 51, 128}	0
max_pooling2d_1 (MaxPooling2D)	(None, 19, 25, 128)	0
conv2d_2 (Conv2D)	(None, 15, 21, 256)	819456
activation_2 (Activation)	{None, 15, 21, 256}	0
max_pooling2d_2 (MaxPooling2D)	(None, 7, 10, 256)	0
flatten (Flatten)	(None, 17920)	0
dropout (Dropout)	(None, 17920)	0
dense (Dense)	(None, 128)	2293888
activation_3 (Activation)	{None, 128}	0
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
activation_4 (Activation)	{None, 128}	0
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
activation_5 (Activation)	{None, 128}	0
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 2)	258
activation_6 (Activation)	{None, 2}	0

Figure 3.24: Depth classification model training details

3.8.2 Size Classification Model

For tumor size evaluation, we developed another CNN sequential model. The model was designed to distinguish the sizes of tumors. It also has three convolution layers, three Max pooling layers, and activation functions for the feature learning part.

The structure of the model is shown in Fig. 3.25, and details on the model development are given in Fig. 3.26. The classification is implemented via four dense layers with *ReLU* and *ELU* activation functions and *Dropout* layers of 0.3 and 0.2. The final fully connected layer has *Softmax* activation function and three nodes, corresponding to the small, medium, or large classes. We used *Adamax* optimizer, and *Accuracy* as the metric for the model performance evaluation during training. The Python script for the model development is given in Appendix E.

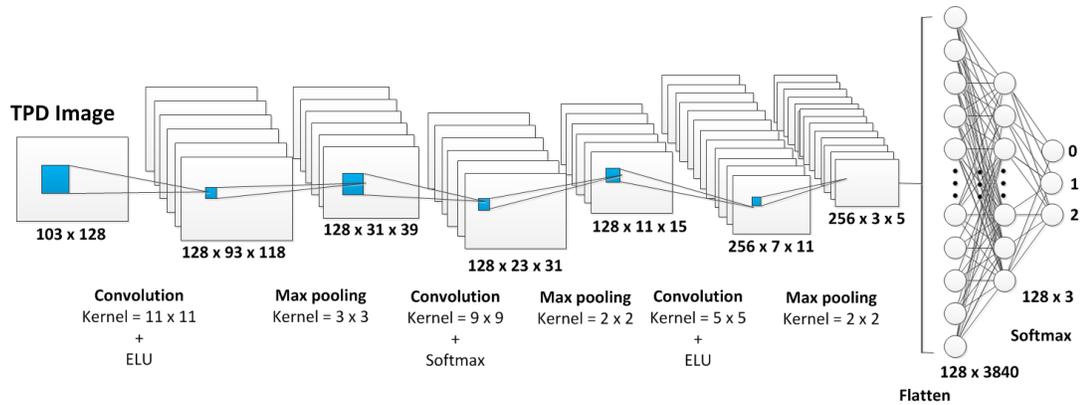


Figure 3.25: TPD size classification model (TPDModelSize)

The data for training and validation of the model is developed from the tactile phantom imaging data combined into the Tactile Profile Diagrams dataset for size classification. The dataset development and model training are described in Chapter 4.

3.8.3 Stiffness Classification Model

For tumor stiffness evaluation from Tactile Profile Diagrams, we designed and trained a third sequential CNN model named TPDModelStiffness. As with two previous models, it has three convolution layers, three *Max pooling* layers, and activation functions for the feature learning part.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 93, 118, 128)	15616
activation (Activation)	(None, 93, 118, 128)	0
max_pooling2d (MaxPooling2D)	(None, 31, 39, 128)	0
conv2d_1 (Conv2D)	(None, 23, 31, 128)	1327232
activation_1 (Activation)	(None, 23, 31, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 11, 15, 128)	0
conv2d_2 (Conv2D)	(None, 7, 11, 256)	819456
activation_2 (Activation)	(None, 7, 11, 256)	0
max_pooling2d_2 (MaxPooling2D)	(None, 3, 5, 256)	0
flatten (Flatten)	(None, 3840)	0
dropout (Dropout)	(None, 3840)	0
dense (Dense)	(None, 128)	491648
activation_3 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
activation_4 (Activation)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
activation_5 (Activation)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 3)	387
activation_6 (Activation)	(None, 3)	0

Figure 3.26: Size classification model training details

The structure of the model is presented in Fig. 3.27. The details on the model development are given in Fig. 3.28. The classification is done using four dense layers with *ReLU* and *ELU* activation functions and *Dropout* layers of 0.3 and 0.2. The final fully connected layer has *Softmax* activation function and two nodes, corresponding to the soft and stiff classes. Similar to the other two models, we used *Adamax* optimizer, and *Accuracy* as the metric for the model performance evaluation during training. The model was built to distinguish tumors with different stiffness. The Python script for the model development is given in Appendix E.

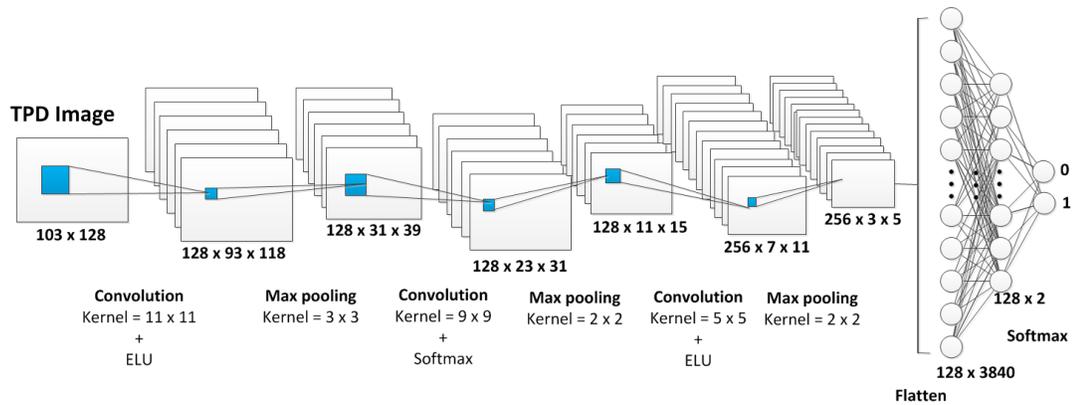


Figure 3.27: TPD stiffness classification model (TPDModelStiffness)

The data for training and validation of the model is developed from the tactile phantom imaging data combined into the Tactile Profile Diagrams dataset for stiffness classification. The dataset development and model training are described in Chapter 4.

3.9 Summary

In this chapter, we described the sensing principle and design of Tactile Imaging Probe. Then we outlined the capabilities of the developed TIP acquisition software. We presented the calculation algorithms for tumor size and stiffness estimation from sets of tactile images. We also proposed a method to extract the application specific information from the sets of TIP images and combine it into a pattern image or Tactile Profile Diagram for each corresponding tactile imaging set. Next, we described algorithms for tumor size and stiffness estimation from Tactile Profile Diagrams directly. Finally, we outlined the method to

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 93, 118, 128)	15616
activation (Activation)	(None, 93, 118, 128)	0
max_pooling2d (MaxPooling2D)	(None, 31, 39, 128)	0
conv2d_1 (Conv2D)	(None, 23, 31, 128)	1327232
activation_1 (Activation)	(None, 23, 31, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 11, 15, 128)	0
conv2d_2 (Conv2D)	(None, 7, 11, 256)	819456
activation_2 (Activation)	(None, 7, 11, 256)	0
max_pooling2d_2 (MaxPooling2D)	(None, 3, 5, 256)	0
flatten (Flatten)	(None, 3840)	0
dropout (Dropout)	(None, 3840)	0
dense (Dense)	(None, 128)	491648
activation_3 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
activation_4 (Activation)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
activation_5 (Activation)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 2)	258
activation_6 (Activation)	(None, 2)	0

Figure 3.28: Stiffness classification model training details

estimate depth, size, and stiffness of tumors by employing Convolutional Neural Network pattern recognition capabilities on Tactile Profile Diagrams.

CHAPTER 4

TACTILE IMAGING PROBE EXPERIMENT

In this chapter, we describe the Tactile Imaging Probe experiments. We outline the development of breast tissue and tumor phantoms. Next, we present the experimental setup and results for size and stiffness estimation from tactile images using phantom data and *in-vivo* human patients' data. Furthermore, we show examples of constructed Tactile Profile Diagram for varying tumors. We also show the performance of Tactile Profile Diagrams classification with Convolutional Neural Network to estimate depth, size, and stiffness of the tumors for phantom data and for *in-vivo* human data. Finally, we show the TPD classification results with CNN models improve the performance of the size calculation algorithm.

4.1 Phantom Development

Here we outline the construction of the breast tissue mimicking phantom. The phantom is necessary to test the capabilities of the Tactile Imaging Probe.

4.1.1 Review of Phantoms that Mimic Mechanical Properties of Tissues

The Tactile Imaging Probe experiments require custom breast tissue mimicking phantoms. Multiple studies have been done on developing tissue-like artificial breast phantoms [64, 65, 66, 67, 68, 69, 70, 71]. The phantoms were made from different types of materials, which are inorganic or organic by nature. The most commonly used materials for mechanical phantoms are gelatin, agar, fibrin, Polydymethyl siloxane, Polyvinyl chloride, and Polydymethyl alcohol (PVA) silicones.

Not many of the mechanical property tissue phantoms incorporate tumors within the synthetic tissue. Researchers who work in breast cancer research develop phantoms with tissue inclusions to test methods. Egorov and Sarvazyan fabricate a SEMICOSIL hydrogel tissue and tumor mimicking phantom [72]. The inclusions had varying sizes yet constant stiffness.

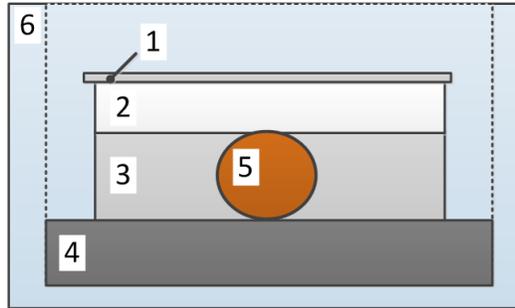
Polyvinyl chloride (PVC) and Polydimethyl siloxane (PDMS) phantoms are well suited for mechanical property estimation experiments [34, 36, 73]. In this work, we developed a phantom with tissue-like mechanical characteristics and with embedded inclusions using PDMS plastisol. The phantom is composed of multiple layers and is durable even with repeated mechanical compressions. It includes palpable PDMS inclusions of variable depth, stiffness, and size.

4.1.2 Proposed Phantom Design

The proposed mechanical property breast tissue phantom is composed of multiple layers. The base material of the tissue layers is Polydimethyl siloxane (PDMS) due to its safety and easiness to use, adjustable stiffness, and high tolerance to heat and mechanical compressions. Polydimethyl siloxane is commercially available silicone rubber. It is composed of two materials: Base agent (A) and Curing Agent (B). We used RTV 6136-D1 (Momentive Performance Materials, Waterford, NY), a low viscosity silicone dielectric gel, to prepare PDMS. Two components A and B were mixed in a different mixing ratio by weight.

We developed four types of tissue layers using PDMS: base, intermediate, depth, and skin layers. Figure 4.1 shows the schematics of the phantom and its layers. The description of each layer is also provided on the figure. The spherical tumors are manually cut out of cured PDMS cubes with a range of stiffness. The spherical tumors are manually cut out of cured PDMS cubes with a range of stiffness to mimic different tumor sizes and stiffness characteristics.

The base layer is a supportive layer, which mimics ribs and muscles under breast tissues. The PDMS ratio (A:B components) of the layer is 1:20 (629 kPa). Intermediate layers of different thicknesses (10 mm, 12 mm, 14 mm, 16 mm and 18 mm) are developed to embed the tumors of different diameters (10 mm, 12 mm, 14 mm, 16 mm and 18 mm). Examples of four manually cut tumor mimicking samples (12 mm, 14 mm, 16 mm, and 18 mm) are shown in Fig. 4.2. PDMS ratio of the intermediate layer is 1:2 (94 kPa). Depth layers are made of PDMS ratio 1:2 (94 kPa) to vary the depth of the embedded tumors. Depth layers' thicknesses are the following: 2 mm, 4 mm, 6 mm, 8 mm, 10 mm, 12 mm, and 14 mm. The skin layer is a thin (< 1 mm) and transparent PVC layer 78 kPa. It serves as a protective



1. Protective top layer	PVC	78 kPa
2. Depth layer	PDMS 1:2 Ratio	94 kPa
3. Intermediate layer	PDMS 1:2 Ratio	94 kPa
4. Base layer	PDMS 1:20 Ratio	629 kPa
5. Tumor Phantom	PDMS Ratios (from 1:2.5 to 1:20)	130 kPa ... 629 kPa
6. Pyrex glass container		

Figure 4.1: Breast phantom for TIP experiments

layer for compression experiments.



Figure 4.2: Phantom tumor samples

The PDMS ratios (A:B components) of the tumors are the following: 1:2.5, 1:3, 1:3.5, 1:4, 1:4.5, 1:5, 1:5.5, 1:6; 1:7, 1:8, 1:9, 1:10, 1:12.5, 1:15; 1:17.5, 1:20. These PDMS mixing ratios are covering Young's Modulus range from approximately 130 kPa to 629 kPa. All of Young's moduli of the PDMS samples were obtained using the compression technique with the Instron 5944 testing machine [74]. Each of the stiffness levels had four cube size samples for the test with Instron (Figure 4.3, Attachment D).



Figure 4.3: PDMS samples from Instron test

4.2 Phantom Experiment

In this section, we outline the experimental design for Tactile Imaging Probe to test its capabilities and algorithms. We estimate the size and stiffness of tumors directly from the sets of TIP images and later present the results [34].

4.2.1 3D Interpolation for Size Estimation Using Tactile Images

To estimate the size of embedded tumors TIP images, we developed three-dimensional interpolation surfaces for four different cases: small and shallow tumors, small and deep tumors, large and shallow tumors, and large and deep tumors. Tumors smaller than 12 mm are considered small, and tumors above 12 mm in diameter are considered large. Depths smaller than 10 mm are shallow in this study, and depths above 10 mm are deep. The 3D interpolation model is a calibration technique for TIP size estimation from sets of tactile images. However, the calibration may differ if the same PDMS sensing element is used for experiments for an extended amount of time. The PDMS becomes less translucent, and the calibration requires adjustment.

We developed interpolation models for both PDMS sensing element conditions. The first model was developed for the new condition of PDMS. We built the models from the TIP dataset, which contained images of small inclusions (8.05 mm, 9.86 mm, 11.70 mm) and large inclusions (13.63, 16.38 mm, 19.47 mm). The shallow samples had 0 mm depth, and deep samples had 10 mm depth. Each sample was repeated in 3 trials. The Young's

modulus of all samples was above 250 MPa.

The second 3D interpolation model was developed for the used condition of the sensing element. Small size samples included 9.81 mm and 11.94 mm inclusions. The sizes of the large samples were 16.08 mm and 22.48 mm. Deep samples were embedded in 10 mm depth, and shallow samples were embedded in 5 mm depth. Each sample was repeated in 3 trials as well. The Young's modulus of all samples was above 250 MPa.

We used MATLAB to fit the data from the tests and to build four models for each PDMS condition (new PDMS probe and used PDMS probe). The models' shape is given by Equation 3.1, which can be rewritten as follows.

$$D(f, I) = P_{00} + P_{10}f + P_{01}I + P_{20}f^2 + P_{11}fI + P_{02}I^2 + P_{30}f^3 + P_{21}f^2I + P_{12}fI^2 + P_{03}I^3, \quad (4.1)$$

where $D(f, I)$ correspond to the function of tumor's size, which depends on TIP applied force, f , in N, and sum of intensities, I , from the corresponding tactile image (see Section 4.2.1 for details).

The calculated polynomial interpolating parameters for the models presented in Table 4.1. As an example, Fig. 4.4 shows the 3D size interpolating models for the used PDMS condition. The four plots correspond to small and shallow, small and deep, large and shallow, and large and deep tumors size estimation models. Larger samples have fewer variations among the three trials, and as a result, the sizes of larger tumors will be estimated with higher accuracy than of smaller tumors.

4.2.2 Size and Stiffness Estimation Using Tactile Images

The test objective of this experiment was to determine the depth and size of tumors that TIP can detect. We used different tumor depths (0 mm, 3 mm, 5 mm, 10 mm, 12 mm and 15 mm), and tumor sizes (3 mm, 10 mm, 15 mm and 18 mm) for this experiment. We also used two types of tumor stiffness: softer tumor inclusion (1:3 ratio A:B or 134 kPa) and stiffer tumor inclusion (1:20 ratio A:B or 466 kPa). The inclusions were tested in 2017 using Instron testing machine to determine the true elastic modulus values for the tumor samples.

Table 4.1: TIP size estimation interpolation parameters

Parameters	PDMS Probe Condition			
	New	Used	New	Used
	Size: small		Size: large	
	Depth: shallow			
P_{00}	-724.70000	-37.17000	-104.80000	-10.48000
P_{10}	-11.77000	-1.69400	0.36490	0.20650
P_{01}	1.341e-05	3.956e-07	1.821e-06	2.115e-07
P_{20}	-0.02005	0.00303	-	0.003634
P_{11}	1.33e-07	6.948e-09	-2.243e-09	-1.192e-09
P_{02}	-8.124e-14	-	-9.583e-15	-
P_{30}	-	-	-	-
P_{21}	1.017e-10	-	-	-
P_{12}	-3.736e-16	-	8.705e-19	-
P_{03}	1.631e-22	-	1.731e-23	-
R-square	0.8845	0.2245	0.9952	0.9838
	Depth: deep			
P_{00}	634.80000	-20.73000	-9.11400	-16.71000
P_{10}	-13.56000	0.32120	0.59430	0.347300
P_{01}	-8.161e-06	2.6e-07	1.463e-07	2.717e-07
P_{20}	-0.17210	0.002390	-0.02784	0.005321
P_{11}	2.126e-07	8.178e-12	-4.067e-09	-9.882e-10
P_{02}	2.635e-14	-	-	-
P_{30}	-0.0002036	-	4.47e-05	-
P_{21}	1.066e-09	-	1.38e-10	-
P_{12}	-7.875e-16	-	-	-
P_{03}	-	-	-	-
R-square	0.7327	0.7913	0.9682	0.9788

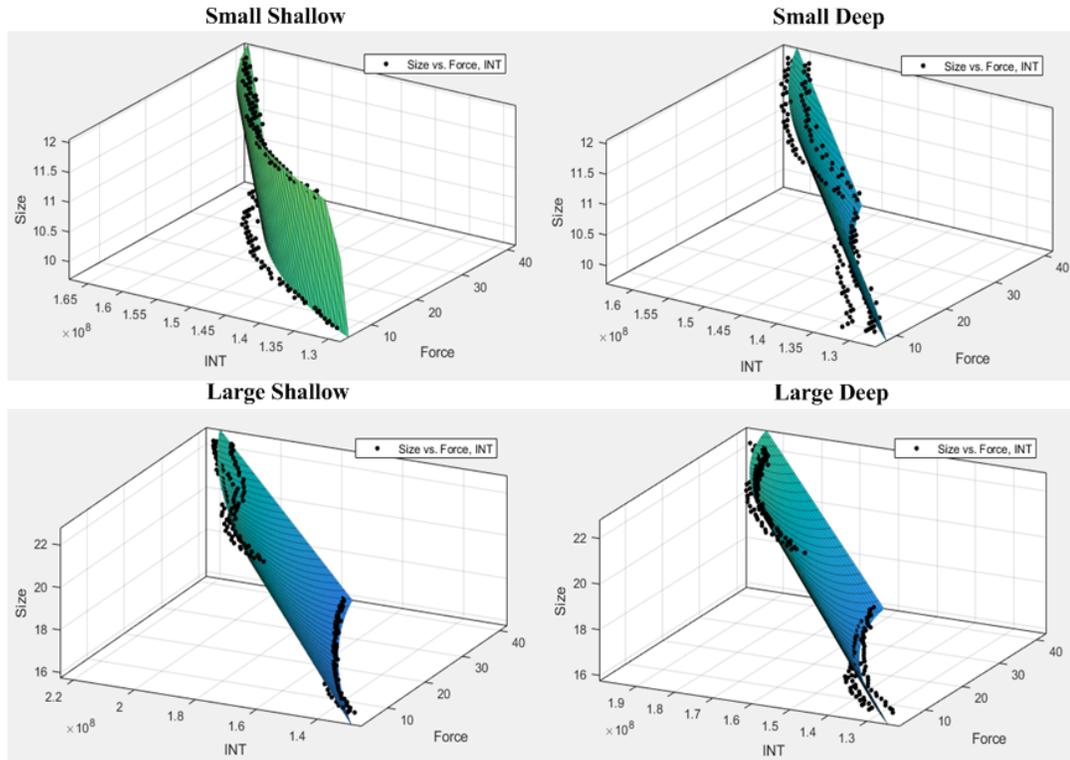


Figure 4.4: 3D Interpolating models for the used PDMS

Polyvinyl chloride (PVC) was used to prepare multiple layers of the breast phantom. There are four layers; base, intermediate, depth, and protective layer, as shown in Fig 4.5. The PVC materials we used were "regular liquid plastic," "super-soft liquid plastic," and "softener" (M - F Manufacturing Company Inc., TX, USA). The base layer was made from "regular liquid plastic." The intermediate and depth layers were softer and were made of "regular liquid plastic" plus "softener" with the 1:1 ratio and addition of skin color dye (Flesh Tone Silicone Pigment, Smooth-On Inc.). The thin protective layer was made from the "super soft liquid plastic" material.

During the tests, we varied the depth by adding different layers from 0 mm to 23 mm to estimate the maximum depth of the tumor that the TIP can detect [34]. The results of size and stiffness estimation experiments are presented in Table 4.2 and Table 4.3.

Soft Tumor

At superficial depth (0 mm), the tumor phantom sizes 10 mm, 15 mm, and 18 mm are detected better with a less size error percentage of 1.87%, 9.61%, and 18.12%, respectively.

Table 4.2: Size and Deformation Index estimation at shallow tumor depths

Tumor inclusion	True Size (mm)	Estimated Size (Mean)	Size Error, (%)	DI (10^4)
Softer	3	8.24	174.78++	3.32
	10	10.77	1.87	5.06
	15	13.56	9.61	12.13
	18	14.74	18.12	25.94
Stiffer	3	8.55	185.51++	3.59
	10	9.50	5.01	8.68
	15	13.92	7.20	15.81
	18	14.95	16.94	29.56
(a) Depth=0 mm				
Softer	3	8.26	175.40++	2.71*
	10	9.59	13.54	3.57
	15	13.58	9.47	3.25
	18	13.95	22.51	2.53
Stiffer	3	6.07	102.22++	2.07*
	10	8.66	13.35	3.88
	15	12.20	18.70	5.97
	18	14.07	21.83	9.95
(b) Depth=3 mm				
Softer	3	8.34	178.13++	2.17
	10	9.24	15.97	3.38
	15	13.20	12.03	4.02
	18	13.63	24.27	9.41
Stiffer	3	8.58	186.14++	3.61
	10	8.68	13.15	4.53
	15	11.69	22.10	6.18
	18	13.97	22.41	17.47
(c) Depth=5 mm				

Table 4.3: Size and Deformation Index estimation at deep tumor depths

Tumor inclusion	True Size (mm)	Estimated Size (Mean)	Size Error, (%)	DI (10^4)
Softer	3	8.37	179.14++	3.38
	10	8.65	24.67	2.26
	15	13.30	11.36	3.09
	18	14.13	21.49	5.97
Stiffer	3	7.93	164.36++	3.80
	10	9.03	9.68	3.50
	15	13.65	8.98	4.36
	18	13.93	22.61	10.60
(d) Depth=10 mm				
Softer	3	n/a	n/a	2.19
	10	8.95	23.40	3.04
	15	12.99	13.38	2.66
	18	14.78	17.92	3.53
Stiffer	3	n/a	n/a	2.98
	10	9.22	7.77	3.15
	15	14.13	5.78	2.70
	18	14.40	20.03	4.212
(e) Depth=12 mm				
Softer	3	n/a	n/a	1.88*
	10	8.58	23.97	2.94*
	15	13.77	8.23	2.04
	18	14.73	18.19	3.69
Stiffer	3	n/a	n/a	1.47*
	10	8.84	11.64	2.37*
	15	14.10	5.98	2.98
	18	14.69	18.38	4.60
(f) Depth=15 mm				

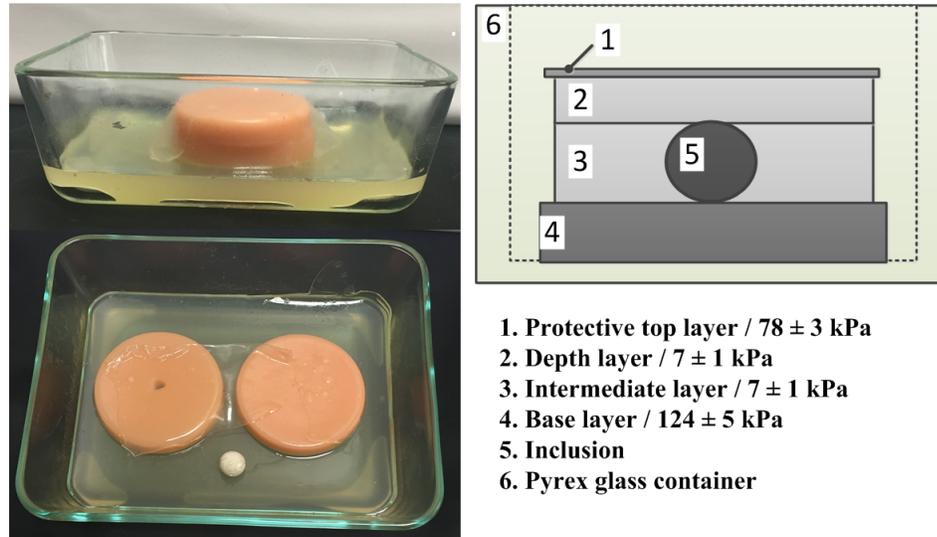


Figure 4.5: Breast phantom for tactile experiments

As the depth increased from 0 mm to 15 mm, the size estimation error only increased slightly, which shows the sensitivity of TIP and the accuracy of measuring the tumor size in varying depth. The 3 mm size estimation shows a large error, which shows that size detection performance of TIP decreases when the tumor sizes are less than 3 mm.

Stiffer Tumor

The 1:20 samples are stiffer than 1:3; hence theoretically, the 1:20 samples represent malignant tumors, which are considered to be stiffer than benign tumors. The result shows that TIP can determine the size of 1:20 samples more accurately than that of 1:3, and the error is smaller. This shows that TIP is able to detect malignant tumors or stiffer growths over benign or non-stiff tumors. The phantom testing showed that the TIP is able to determine the size of tumors above 3 mm in size up to 18 mm and to a depth of up to 15 mm.

Deformation Index (DI)

The deformation index is calculated as described in Chapter 3, and the results are presented in Table 4.2 and Table 4.3. Comparing the same size tumors at the same depth between the softer and stiffer tumors, we note that the values follow a similar pattern as explained above. The stiffer samples show a higher deformation index than the softer samples. At a depth of 15 mm, a small sized-tumor showed a large error due to the difficulties

in detecting deeply embedded tumors.

4.3 *In-vivo* Human Experiment Using Tactile Images

The TIP was tested in a pilot study of twenty-one human patients (IRB# 22050 Temple University) to determine the accuracy of the device in a real healthcare environment [34]. The patients were scheduled for a biopsy, where the histopathology reports from the biopsy were used as the truth values in the classification output of TIP. Patients were randomly selected by the radiologists.

4.3.1 *Size and Stiffness Estimation*

For the human patient experiments, the physicians initially approximate the location, size, and depth of the masses. The doctors were instructed to use a 12 mm threshold for size and a 10 mm threshold for depth. However, if the mass was approximated as more than 30 mm in diameter, the 3D interpolation for large and deep inclusions was used for size calculation. The results from the human patients' experiments *in-vivo* are shown in Table 4.4. The patients used for the studies had tumors in varying sizes from 11 mm to 60.8 mm, which was determined from the ultrasound images (true tumor size in Table 4.4).

From the results, the serial numbers from 1 to 15 have a size estimation error of less than 20%. In these cases, the estimated size values are close to the real values. For example, in serial number 10, a patient with a small tumor of size 11 mm in a depth of 7 mm is showing a very good estimated size and is found to be malignant through the histopathology report. This shows the ability of the TIP to detect small tumors at deeper depth if it is stiffer or malignant. In serial numbers 16 to 21, the error percentage increased from 20% to 70%. This is mainly due to the non-palpable nature of the tumor and the human error in estimating the depth and size.

The deformation index is also given in Table 4.4. We observe that the malignant tumors (denoted with "+" symbol) show a higher deformation index than the benign cases. The malignant and benign cases are confirmed using histopathology reports. The malignant tumors have deformation index values above 50,000, whereas benign tumors have smaller

Table 4.4: The *in-vivo* test results for estimating size and deformation index using TIP

Serial number	True tumor size (mm)	Calculated tumor size (mm)	Error (%)	DI (10^4)
1+	54.20	57.19	5.52	6.79
2+	56.40	52.95	6.12	4.19
3+	43.30	46.33	7.00	4.91
4	12.20	13.15	7.79	2.56
5	60.40	54.37	9.98	1.60
6	17.70	15.77	10.90	1.39
7	13.90	15.5	11.51	3.88
8+	53.80	60.82	13.05	4.31
9+	60.80	52.49	13.67	3.58
10+	11.00	9.39	14.64	9.34
11	35.00	40.31	15.17	1.90
12+	53.80	45.04	16.28	3.30
13	17.50	14.58	16.69	2.74
14	14.00	16.56	18.29	3.62
15+	24.29	19.65	19.10	5.04
16+	12.00	14.7	22.50	5.47
17	20.10	14.04	30.15	9.40
18+	35.90	50.23	39.92	5.58
19	20.06	11.86	40.88	2.90
20	11.02	15.96	44.83	10.75
21	30.61	52.23	70.63	4.58

+ Represents the malignant cases: information obtained by clinical pathology reports

deformation index values. This pattern is due to the fact that malignant tumors are stiffer than benign tumors.

4.3.2 Breast Tumor Risk Score

After the size and deformation index of tumors are estimated using TIP, we calculate the Risk Score to classify tumors as benign or malignant.

Risk Score is a unitless numerical value, which can be used as a scale to classify the tumor as malignant and benign. Based on the calculated size of the tumor and measured deformation index, the breast tumors are classified as benign and malignant using the scoring method. The Risk Score ranges from 0 to 5, where zero represents the benign, and five represents the malignant tumor. A marginal threshold value was set, where any risk score below the threshold is considered benign. The calculated Risk Score is based on the below equation,

$$RiskScore = \left[\frac{W_1 \times S}{S_{max}} + \frac{W_2 \times DI}{DI_{max}} \right] R, \quad (4.2)$$

where W_1 and W_2 are the two weights used for size and deformation index, respectively, S represents the estimated size value, S_{max} is the maximum estimated size value, DI is the calculated deformation index, DI_{max} is the maximum calculated deformation index. $R = 5$ is the highest value of Risk Score used.

To classify tumors, we choose the pair of weights (W_1 and W_2), which gives the best classification sensitivity and specificity for a training subset (20 patients). We computed the Receiver Operating Characteristic (ROC) curve with different weights for each training subset, where W_1 and W_2 varied from 0 to 1 in steps of 0.1. The sum of the weights has to be one by using any of the eleven combinations (e.g., 0.1 and 0.9, 0.2 and 0.8, etc.). The optimal weights and corresponding threshold values were determined. For each pair of weights, we computed the ROC curve varying the threshold values from the smallest calculated score to the largest calculated score in a subset. Then we looked for minimum distance from the (0, 1) point in the ROC graph to the curve, where (0, 1) point is a perfect classifier. In Fig. 4.6, a sample ROC curve is shown with the optimal point at (0.1818, 1), which corresponds to 100% sensitivity and 82% specificity. The corresponding threshold

value for the risk score came out to be 1.99, and the optimal weight came out to be $W_1 = 0.3$ and $W_2 = 0.7$. Then the test subset (1 patient) was classified as malignant or benign with the found weights and threshold. The classification results are compared with original clinical pathological reports, as shown in Table 4.5. From the table, we note that except for two false positive cases, the rest of the cases are accurately classified.

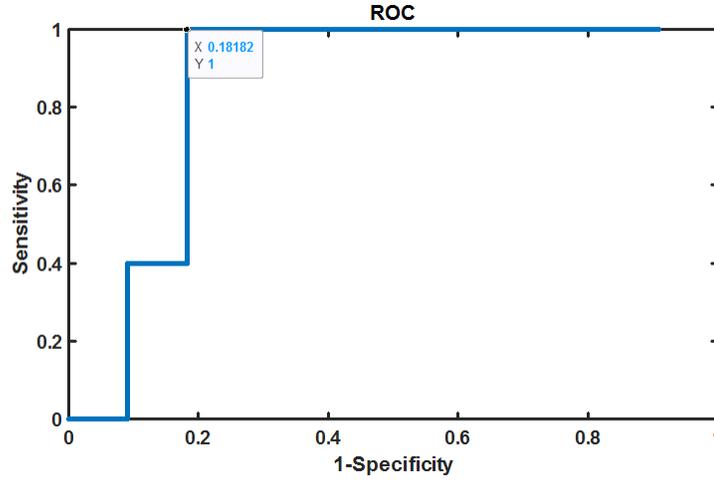


Figure 4.6: ROC curve to determine the threshold

We employed the Leave-One-Out-Cross-Validation (LOOCV) technique to validate the human test results to determine the performance of the TIP during the pilot study. The sensitivity, specificity, and accuracy of the system were measured to assess the reliability of the TIP. These performance metrics were calculated based on the Risk Score using the below equations,

$$Sensitivity_{TIP} = \frac{TP}{TP + FN}(\%), \quad (4.3)$$

$$Specificity_{TIP} = \frac{TN}{TN + FP}(\%), \quad (4.4)$$

$$Accuracy_{TIP} = \frac{TN + TP}{TN + FP + TP + FN}(\%), \quad (4.5)$$

were TP , TN , FP , and FN represent true positive, true negative, false positive, and false negative cases, respectively. False positive is considered to be a case where benign masses

Table 4.5: Risk Score based classification of tumors using TIP output

Serial Number	Pathology Results	Calculated Risk Score	TIP Classification
1	Malignant	3.62	Malignant
2	Malignant	2.67	Malignant
3	Malignant	2.74	Malignant
4	Benign	1.16	Benign
5	Benign	1.86	Benign
6	Benign	0.84	Benign
7	Benign	1.65	Benign
8	Malignant	3.00	Malignant
9	Malignant	2.46	Malignant
10	Malignant	3.27	Malignant
11	Benign	1.61	Benign
12	Malignant	2.40	Malignant
13	Benign	1.25	Benign
14	Benign	1.59	Benign
15	Malignant	2.13	Malignant
16	Malignant	2.15	Malignant
17	Benign	0.65	Benign
18	Malignant	3.05	Malignant
19	Benign	1.24	Benign
20	Benign	4.42	Malignant++
21	Benign	2.78	Malignant++

++ Denotes false positive cases

are classified as malignant, whereas false negative are cases where malignant masses are classified as benign. True positives are correctly classified malignant cases. True negatives are correctly classified as benign cases.

For our data set, $Sensitivity_{TIP}$, $Specificity_{TIP}$ and $Accuracy_{TIP}$ are calculated to be 100%, 82%, and 90.5%, respectively. We note that the optimal weights for all of the subsets came out to be $W_1 = 0.3$, $W_2 = 0.7$. The optimal threshold came out to be 1.99. Those optimal weights may not be the global optimal values. However, weighing stiffness more than the size seems to agree with the literature and the experiences of doctors.

4.4 Tactile Profile Diagrams Using Phantom Data

Tactile Profile Diagram (TPD) is a representative pattern image of a TIP image set and a pictorial representation of relative stiffness and size of the tumor. Therefore we hypothesize that it is possible to estimate the size and stiffness of tumors from their corresponding TPDs.

4.4.1 Construction of Tactile Profile Diagrams

We constructed Tactile Profile Diagrams using the method outlined in Section 3.6. Representative samples of the TPDs, their corresponding volumetric reconstructions, composite gradient images, and segmented magnitude gradient images are presented in Fig. 4.7. Column a) of the table describes a stiff and small tumor. Column b) of the table shows a stiff and large tumor. The following c) column represents a soft and small inclusion. Finally, d) column shows a soft and large inclusion. One can see the differences in the area of high-intensity regions for smaller and larger inclusion. The sharpness and height of the 3D reconstruction peaks, and the average magnitude of the G_{magS} images, contain the information about the stiffness of inclusions. Also, the "fuzziness" of the edges in composed gradient images seemed to correlates with inclusion's stiffness.

4.4.2 Size and Stiffness Estimation from Tactile Profile Diagrams

To validated the proposed TPD method for the size and stiffness estimation, we performed the experiments with custom tissue and tumors phantom. We incorporated in the phantom

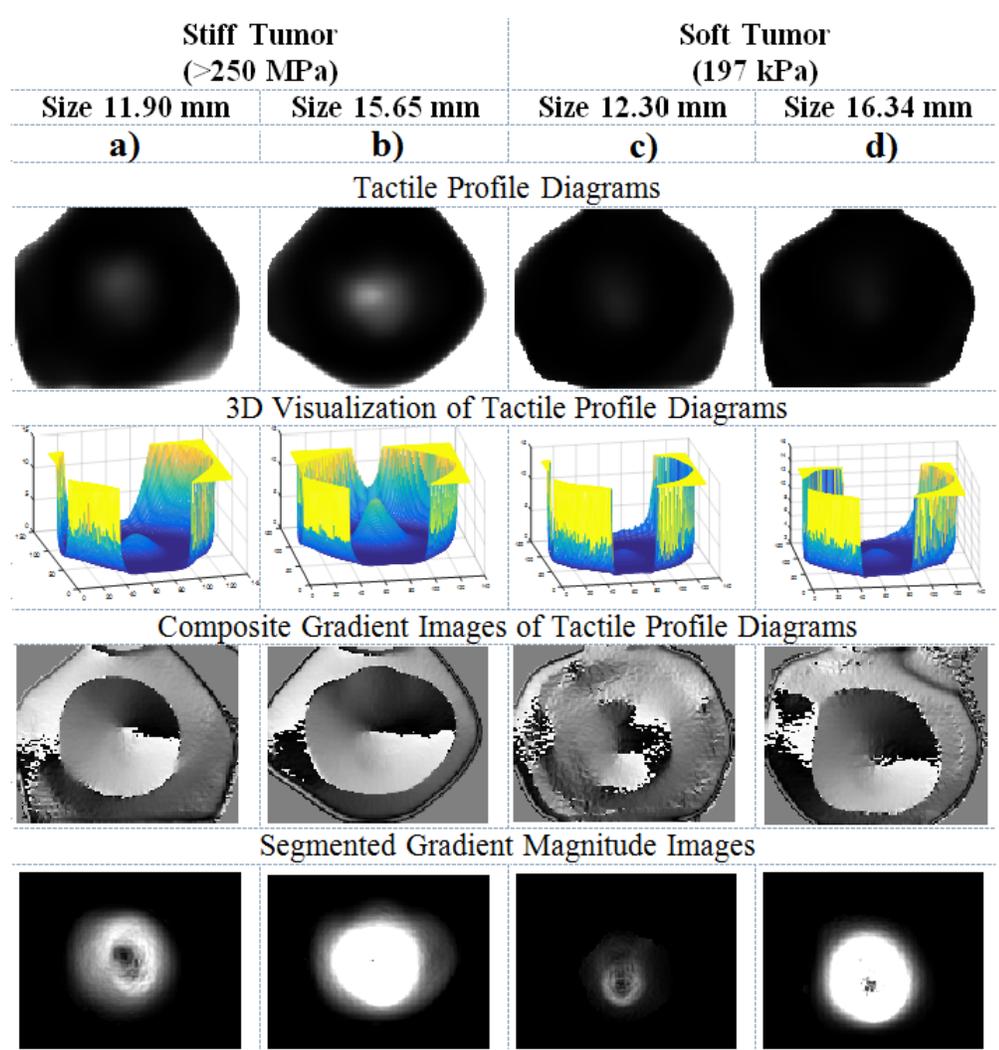


Figure 4.7: Tactile Profile Diagram results

Table 4.6: Size interpolation development samples

Sample	Size, mm	Num. of Pixel (Mean \pm STD)
1	9.73	2329 \pm 149
2	11.98	3548 \pm 832
3	16.17	5058 \pm 306
4	16.08	4323 \pm 278
5	20.16	8450 \pm 529

five different tumor sizes (10 mm, 12 mm, 14 mm, 16 mm, and 18 mm), eight tumor depths (0 mm, 2 mm, 4 mm, 6 mm, 8 mm, 10 mm, 12 mm, and 14 mm), and 17 PDMS ratios of component B to vary the tumor stiffness (2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 7, 8, 9, 10, 12.5, 15, 15.5, 20, plastic ball). We completed the imaging for each combination of depth, size, and stiffness in 13 to 20 trials. TIP was applied in a normal direction to the tissue. The compression forces started at 0 N and were gradually increased to 45 N during the experiments. Each imaging trial corresponds to an independent compression with TIP. The phantom dataset, obtained by TIP imaging using the phantom, contains thousands of entries. This dataset was used for Tactile Profile Diagrams development and classification.

We developed a linear interpolation model to calculate the size of the tumors from TPDs directly. All the samples for the model training test were stiff samples with Young's modulus above 250 MPa. The sizes of the samples are described in Table 4.6. The depth for each tumor was 4 mm. The results of the linear interpolation fit to the training data are shown in Fig. 4.8. The model allows estimating the size of the imaged tumor from the sum of pixels within the deformed region on the corresponding TPD.

Table 4.7 describes the phantom used for the size and stiffness calculation directly from TPDs. Fig. 4.9 shows the phantom component layers, and Fig. 4.10 shows the photo of the test samples used for the test. The size and stiffness estimation results from the TPD method are presented in Table 4.8. The shaded rows correspond to stiff tumors. The calculated values are the average values of the three trials with stiff acrylic inclusions. The 1 mm to 3 mm depth will show a similar error as the 4 mm depth. But with 8 mm depth, the error will increase. The means and standard deviations of the relative size estimation errors are from 1.1% to 33.26% for shallow and from 16.24% to 45.46% for deep tumors.

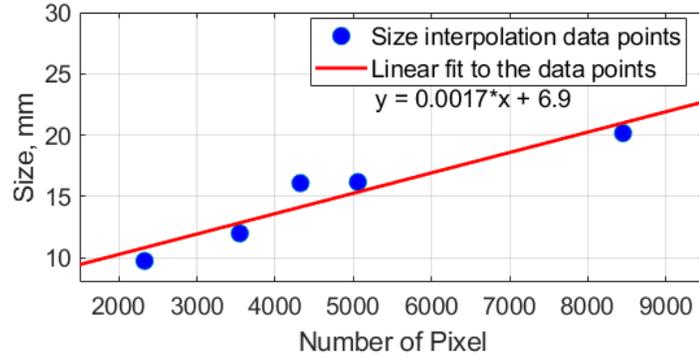


Figure 4.8: Linear size interpolation model results

Table 4.7: Properties of the phantom components

Description	Material	Thickness/Size, mm	Elastic Modulus, kPa
Tissue Layers			
Base	PDMS	15	629
Intermediate	PVC	12, 14, 16	7
Depth	PVC	4	7
Spherical Inclusions			
Stiff	Acrylic	11.90, 15.65, 17.20	250000
Soft	PDMS	12.30, 14.20, 16.34	197

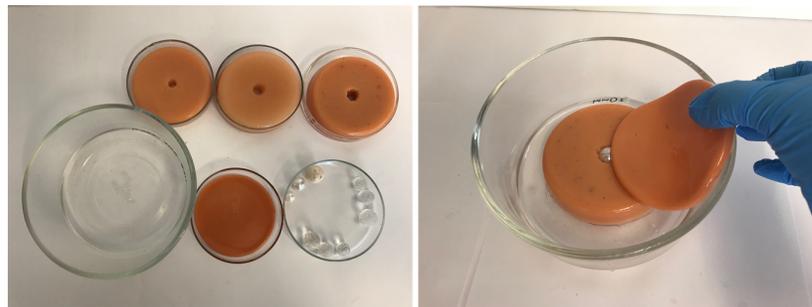


Figure 4.9: Test phantom components and their placement during the TIP experiments

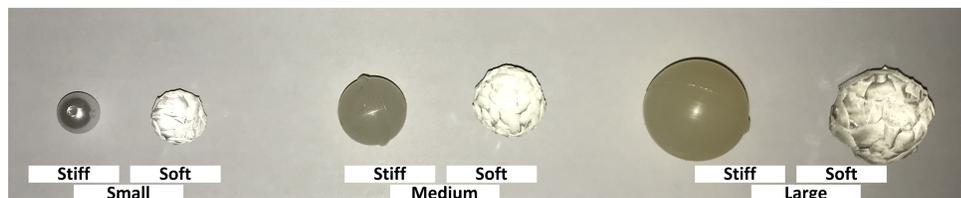


Figure 4.10: Samples for size and stiffness calculation from TPDs

Table 4.8: Size and Stiffness Index calculation for the phantom dataset

Test	Mean Calc. Size, mm	True Size, mm	Calc. Error, %	Mean Calc. SI	True YM, kPa
Shallow Tumors					
1	11.79±1.58	11.92	1.11	0.28±0.10	250000
2	11.75±1.00	12.15	3.25	0.26±0.05	169.00
3	14.19±1.37	15.99	11.28	0.56±0.28	250000
4	10.87±0.21	16.29	33.26	0.05±0.03	169.00
5	22.49±0.98	23.02	2.31	1.56±0.04	250000
6	14.58±0.70	20.92	30.32	0.41±0.21	169.00
Deep Tumors					
7	9.31±1.13	11.92	21.89	0.07±0.01	250000
8	10.08±0.20	12.15	17.05	0.05±0.02	169.00
9	12.07±0.41	15.99	24.53	0.29±0.16	250000
10	11.12±0.16	16.29	31.72	0.02±0.01	169.00
11	19.28±0.73	23.02	16.24	2.16±0.34	250000
12	11.41±1.24	20.92	45.46	0.03±0.00	169.00

Shaded rows correspond to stiff tumors

Stiffness index (SI) results which represent the tumor stiffness are visualized via plots for shallow and deep inclusions separately in Fig. 4.11. Lower SI values (in red) are obtained for the softer inclusions, and larger SI values (in blue) are obtained for the stiffer inclusions considering similar size and depth.

4.4.3 Classification of Tactile Profile Diagrams

We developed three models to classify TPDs on classes with respect to the underlying tumor's depth, size, and stiffness. Here we describe the results for TPD classification using Convolutional Neural Network (CNN).

To built CNN models, we used the following hardware configuration: Intel (R) Xeon(R) Gold 5122 CPU @ 3.60 GHz (4 cores), 16.0GB, 64bit Windows 10. We built the models using Spider 3, Python 3.7.3, TensorFlow 1.14.0, and Keras 2.2.4.

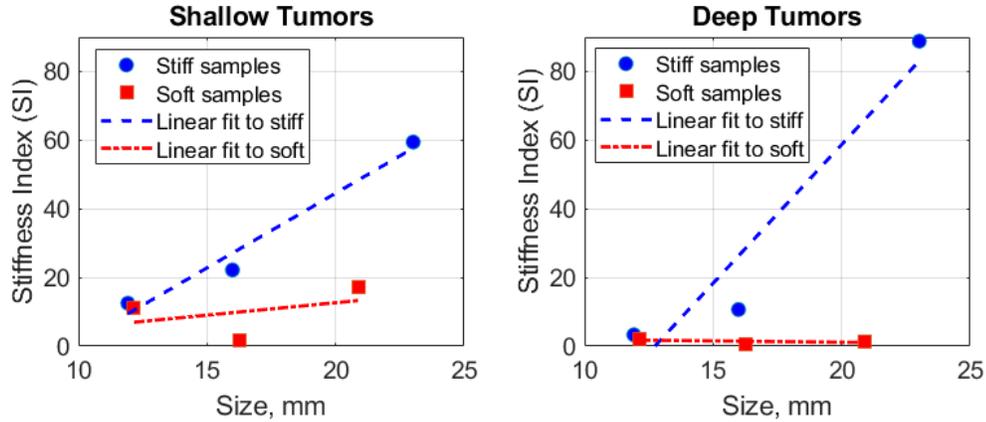


Figure 4.11: Stiffness Index for shallow and deep inclusions

During data pre-processing step, the number of images in each class for training should be the same, or the trained model will be biased towards the larger class. One of the simplest methods to resolve this issue is to randomly upsample the smaller class. The balance is achieved by adding random copies of the existed images in the underrepresented class [75]. For all models we used the random upsampling method to balance the classes.

We trained the models for 50 epochs with a batch size of 200 TPDs. We employed *Adamax* optimizer during training and *Accuracy* as the metric for models performance evaluation during models training. The accuracy and loss results for the three models and the examples of TPD classification are shown on figures later in this chapter. The models were further tested on a set aside subset of phantom data, and the results are presented in Section 4.4.3.

Depth Classification Model

The first CNN model is responsible for the tumor depth estimation. The description of the model is given in Section 3.8.1. The model was trained and tested on the data set obtained from the PDMS phantoms described in Section 4.4.2.

To train the *TPDModelDepth* model, we used samples of varying sizes (10 mm, 12 mm, 14 mm, 16 mm, 18 mm, 23 mm) and stiffness (from 130 kPa to >250 MPa). We included a shallow subset: depths 0 mm, 2 mm, and 4 mm, and a deep tumors subset: depths of 6 mm, 8 mm, 10 mm. During the model development, the data division was 80% for training

and 20% for validation. The model was trained on 6768 TPDs, validated on 1692 TPDs.

Figure 4.12 shows the accuracy and loss plot for the training and test subsets. The final validation accuracy of the classification is 0.97.

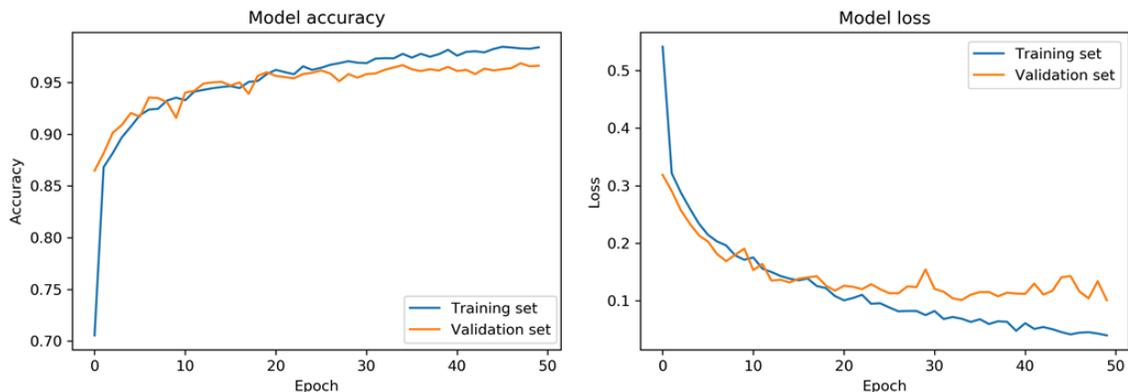


Figure 4.12: Accuracy and loss during training of TPDMoDelDepth over 50 epochs

The depth classification model Python implementation is given in Attachment E. Figure 4.13 shows six random TPD from the validation subset classified. There are examples of Shallow class samples (Class 0) and Deep class samples (Class 1). The figure shows classification predictions for each of the samples as a bar diagram. Text notations under TPD sample images show the model classification results with the winning class probability and the true class of the sample in parenthesis. Blue color bars correspond to the correctly classified TPDs. The red color bar signifies an incorrectly classified TPD.

Size Classification Model

The second TPD CNN model is responsible for classifying tumor sizes as Small, Medium, and Large. The description of the model is given in Section 3.8.2. The model was trained and tested on the dataset obtained from the phantoms described earlier.

During training the TPDMoDelSize model, we included varying depths (0 mm, 2 mm, 4 mm, 6 mm, 8 mm, and 10 mm) and stiffness (from 130 kPa to >250 MPa). The sizes were 10 mm, 12 mm, 14 mm, 16 mm, 18 mm, and 23 mm. We divided the dataset on three classes depending on the tumor's size: small (10 mm and 12 mm), medium (14 mm and 16 mm), and large (18 mm and 23 mm). Inclusions larger than 23 mm and smaller than 10

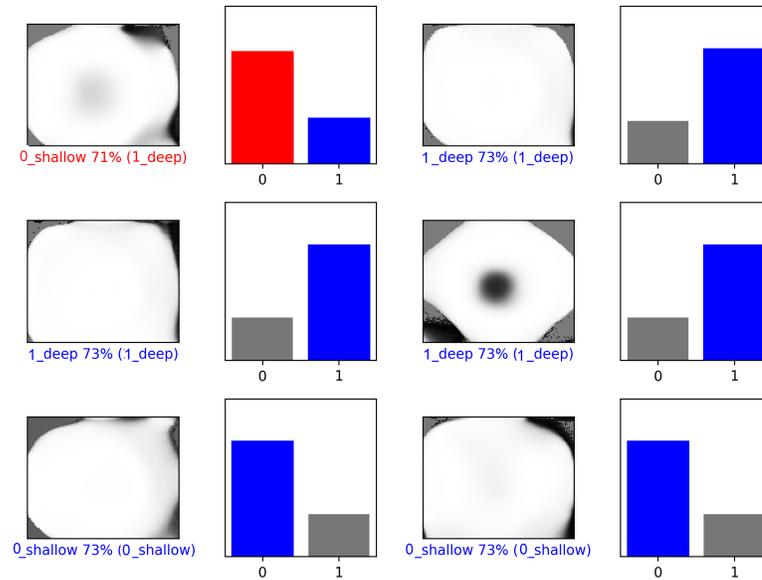


Figure 4.13: Examples of tumors depth classification using TPDMoDelDepth

mm were not used due to the TIP operational range limitations. The model was trained on 6789 TPDs and validated on 1696 TPDs.

Figure 4.14 shows the accuracy and loss plot along 50 epochs for training and validation subsets. The final validation accuracy was 0.96.

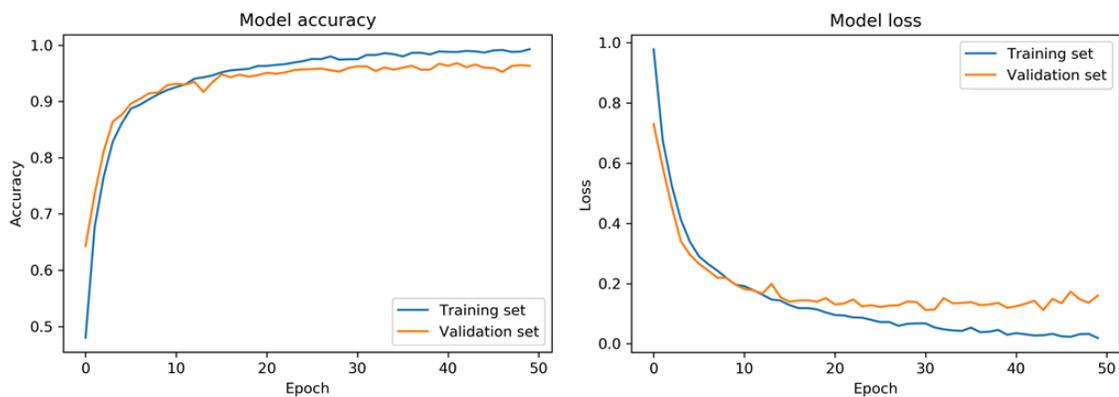


Figure 4.14: Accuracy and loss during training of TPDMoDelSize over 50 epochs

The size classification model implementation is given in Attachment E. Figure 4.15 shows six random TPD from the test subset classified. There are Small (Class 0) samples, Medium (Class 1) samples, and Large (Class 2) samples. The figure shows the classifica-

tion predictions for each of the samples as a bar diagram. The blue color corresponds to the correctly classified TPDs.

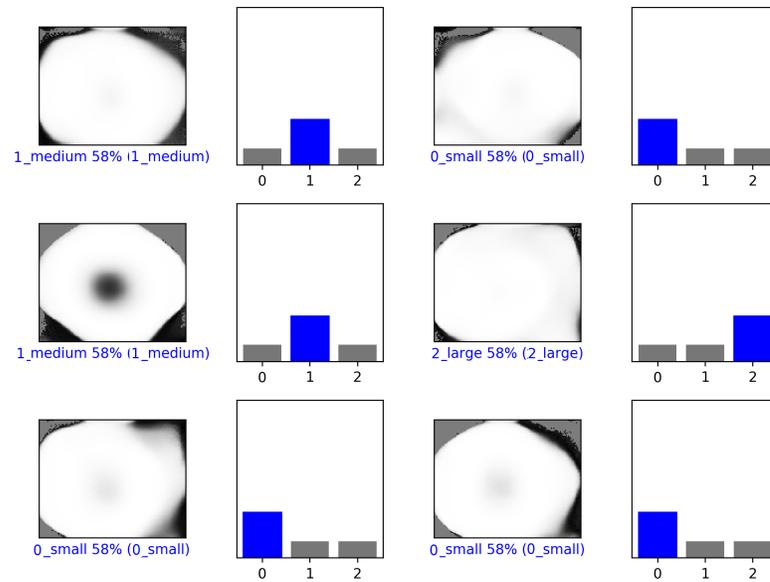


Figure 4.15: Examples for tumors size classification using TPDMoelSize

Stiffness Classification Model

The Stiffness model was trained on sizes 10 mm, 12 mm, 14 mm, 16 mm, 18 mm, 23 mm. We included 0 mm, 2 mm, 4 mm, 6 mm, 8 mm, and 10 mm depths into the dataset. The stiffness of embedded tumors ranged from 130 kPa to more than 250 MPa. The soft tumors subset included 130 kPa to 316 kPa samples. The stiff tumor subset included tumors from 376 kPa to more than 250 MPa elastic modulus values. These ranges correspond to the stiffness of benign *vs.* malignant tumors available in the literature [30]. The model was trained on 6788 TPDs and validated on 1697 TPDs. The description of the model is given in Section 3.8.3.

Figure 4.16 shows the accuracy and loss plot for the training and validation subsets. The final validation accuracy of the classification is 0.90.

The stiffness classification model implementation is given in Attachment E. Figure 4.17 shows six random TPD from the validation subset classified. There are Soft (Class 0) samples and Stiff (Class 1) samples. The figure shows the classification predictions for

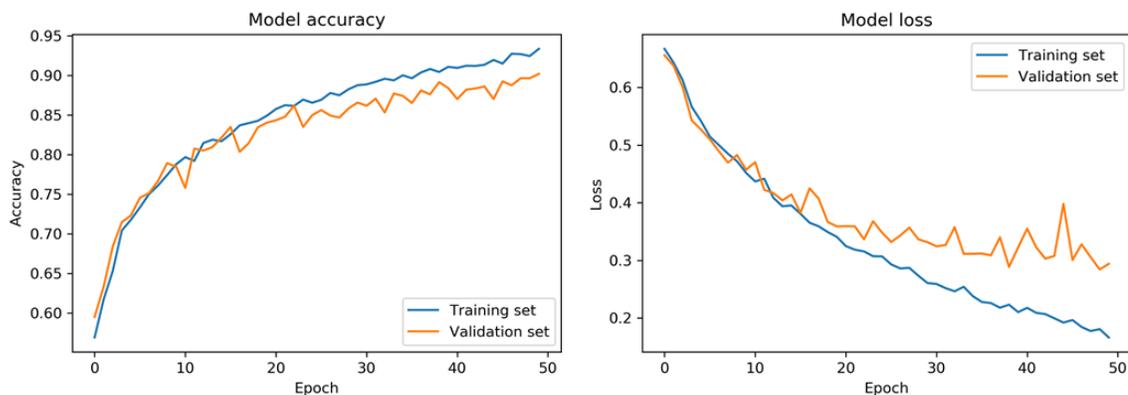


Figure 4.16: Accuracy and loss during training of TPDMoelStiffness over 50 epochs

each of the samples as a bar diagram. The blue color bars correspond to the correctly classified TPDs. The red color bar signifies an incorrectly classified TPD.

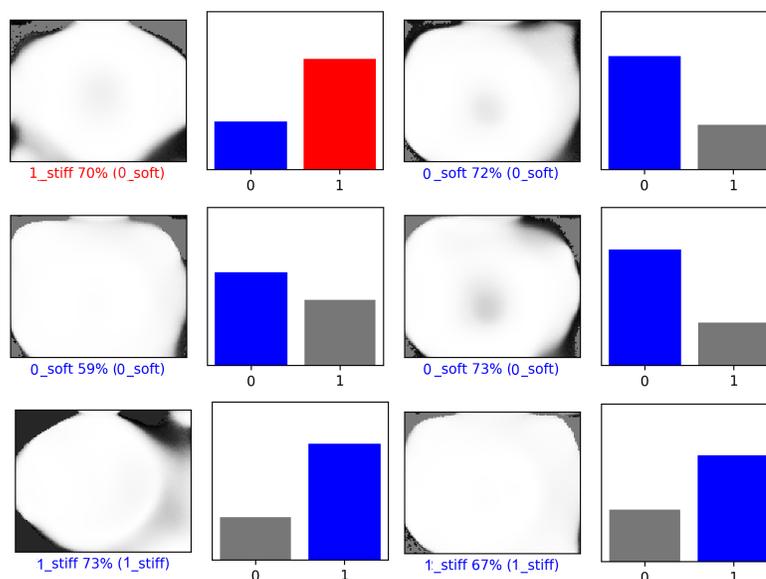


Figure 4.17: Examples of tumors stiffness classification using TPDMoelStiffness

Test Classification Results

To further analyze the developed models, we completed classification of depth, size, and stiffness for the set of phantom data that was used for size and stiffness estimation in Table 4.8. Results from TPD classification are presented in Table 4.9. In the table, the green

shaded rows indicate deep tumors, the light brown rows indicate stiff tumors, and light and darker blue rows correspond to medium and large size inclusions, respectively. The classification accuracy for depth, size, and stiffness for the test set are 92%, 67%, and 83%, respectively. The misclassified cases are shown in red color. The low size accuracy can be explained that the samples for the TPDModelSize were taken some time ago, and the PDMS sensing element slightly changed its properties, as discussed in Section 4.2.1.

Table 4.9: CNN classification results for the phantom data

Sample #	Depth		Stiffness		Size	
	CNN Class	CNN Class (Prob±STD)	CNN Class	CNN Class (Prob±STD)	CNN Class	CNN Class (Prob±STD)
1	Shallow	0.71±0.02	Stiff	0.55±0.23	Med	0.49±0.10
2	Shallow	0.73±0.00	Soft	0.72±0.01	Small	0.55±0.03
3	Shallow	0.68±0.05	Stiff	0.59±0.25	Med	0.58±0.00
4	Deep	0.71±0.01	Stiff	0.58±0.09	Med	0.41±0.01
5	Shallow	0.73±0.00	Stiff	0.73±0.00	Large	0.58±0.00
6	Shallow	0.52±0.10	Soft	0.64±0.09	Large	0.43±0.17
1	Deep	0.71±0.02	Stiff	0.51±0.20	Small	0.42±0.14
2	Deep	0.66±0.06	Stiff	0.50±0.15	Med	0.39±0.15
3	Deep	0.58±0.25	Stiff	0.72±0.01	Med	0.57±0.00
4	Deep	0.69±0.07	Soft	0.71±0.03	Small	0.40±0.18
5	Deep	0.63±0.07	Stiff	0.73±0.00	Large	0.58±0.00
6	Deep	0.73±0.00	Soft	0.51±0.22	Small	0.45±0.17
Accuracy	0.92		0.83		0.67	

Note: Green color indicates true deep inclusions, light brown color corresponds to true stiff tumors, light blue and dark blue colors indicate true Medium and Large inclusions, respectively. The red color font shows incorrectly classified samples.

4.5 Tactile Profile Diagrams Using *In-vivo* Human Data

To test Tactile Profile Diagrams method on *in-vivo* data, we used a pilot human dataset consisted of 13 patients: 5 malignant and 8 benign cases (IRB# 22050 Temple Univer-

sity). The patients were scheduled for a biopsy after TIP acquisition, and we collected the histopathology reports for these patients later from the clinicians.

4.5.1 *Size and Stiffness Estimation from Tactile Profile Diagrams*

We calculated the size and stiffness of the tumors from the corresponding TPDs, and the results are shown in Table 4.10. The patient numbers colored in gray correspond to the malignant cases, and the rest are the eight benign cases. The light blue rows and the dark blue rows correspond to the medium size and the large size tumors, respectively. The light brown rows indicate the stiff tumors among soft tumors. The stiffness of tumors was estimated by clinicians after manual palpation.

Samples 1, 2, 11, 12, and 13 are out of the TIP accuracy range; therefore, they show large estimation errors. The size error for the in-range samples is $17.07 \pm 11.63\%$. The stiffness estimation results generally follow the convention that stiffer samples have larger SI and softer have lower SI within similar in size and depth tumors.

4.5.2 *Classification of Tactile Profile Diagrams*

We used a small human dataset to test the developed methods. CNN classification accuracy for depth, stiffness, and size were 92%, 92%, and 69%, respectfully. Results are presented in Table 4.11. The low size estimation error can be again attributed to the differences in PDMS probe condition between the imaging experiments for the model development and the time of human data acquisition.

4.6 TPD Classification as Preliminary Step for Size Calculation

In this section, we show that TPD classification with CNN models can be integrated as an initial estimate to improve the size and deformation index calculation algorithms' performance.

We selected five representative samples to show the integration method. First, we built a TPD for each sample. Next, we classified the TPD using `TPDModelDepth`, `TPDModelSize`, and `TPDModelStiffness` models. Finally, we used the depth and size classes to select

the 3D interpolation model for the tuned size calculation. The results are presented in Table 4.12 and in Table 4.13. The size calculation result based on the selected with CNN classification 3D interpolation model is shown in bold for each sample. For the sample set, all of the selected models gave the smallest size estimation error, except Sample #5, shown in red. Sample #5 was correctly classified by CNN as a soft sample. We applied an empirical adjustment to the softness of Sample #5 by adding 8 mm to each model's calculated size. After the adjustment, the large shallow model gave the lowest error out of four. Therefore, we suggest that by estimating tumor depth, size, and stiffness parameters with CNN models, we can better calculate the size of the tumor.

4.7 Summary

To verify the performance of the system and the algorithms, first, we developed a breast tissue mimicking phantom with the mechanical properties of breast tissues and tumors. Then, we completed phantom experiments with TIP to create the application specific dataset. We calculated the size and stiffness of tumors from the sets of tactile images using the phantom data and data from human experiments. Size calculation errors were 1.87%, 9.61%, and 18.12% for 10 mm, 15 mm and 18 mm shallow tumors sizes, respectively, for phantom experiments. The tumors' depth increase brought a slight increase in the size calculation error. Additionally, stiffer tumors showed a higher measure of stiffness (DI) vs. the softer tumors of similar depths and sizes. In the case of a human dataset, besides calculating the size and stiffness of the tumors, we also calculated Malignancy Risk Score for patients to classify malignant and benign cases. The classification sensitivity, specificity, and accuracy of the 21 patients' pilot dataset were 100%, 82%, and 91%, respectively.

Next, we proposed the Tactile Profile Diagrams method to encode dynamic tactile information from TIP as pattern images. We showed the calculation method and results for size and stiffness estimation directly from TPDs using phantom data and a human patients' *in-vivo* dataset. Later, we developed CNN classification models to classify the depth, size, and stiffness of tumors from TPD images and demonstrated their capabilities. The validation accuracy for the depth, size, and stiffness classification were 97%, 96%, and 90%,

respectfully. Tumors size calculation from phantom data TPDs directly showed error range from 1.10% to 33.26% for shallow tumors, and from 16.24% to 45.46% for deep tumors. Softer inclusions had smaller SI values for the samples of similar size and depth, while stiffer samples showed higher values.

Finally, we proposed the method to improve tumors size calculation by using the estimation information from TPD CNN classification. The method was verified on a small phantom dataset to eliminate the need for subjective size and depth estimation inputs from a TIP operator and to improve tumors size estimation from tactile images.

Table 4.10: Results for human data size and stiffness estimation

Patient #	US Size, mm	Calc. Size, mm	Calc. Error, %	Doctor Est. Stiffness	SI Calc.
1	7.00	11.57	65.35	soft	0.19
2	7.50	9.38	25.01	stiff	0.47
3	12.20	15.19	24.48	stiff	0.14
4	13.90	15.52	11.69	soft	0.12
5	14.00	17.08	22.01	stiff	0.12
6	15.00	13.97	6.89	soft	0.07
7	17.70	18.78	6.12	soft	0.24
8	18.00	17.50	2.78	soft	0.11
9	30.61	20.51	32.99	soft	0.24
10	32.70	23.02	29.59	soft	0.15
11	53.80	26.64	50.49	stiff	0.16
12	53.80	25.18	53.20	stiff	0.13
13	54.20	23.86	55.98	stiff	0.17

Gray color indicates malignant tumors. The red font shows tumor sizes outside of TIP range.

Table 4.11: CNN classification results for the human dataset

Patient	Depth		Stiffness		Size	
	US Est. Class	CNN Class	Doctor Est. Class	CNN Class	US Est. Class	CNN Class
1	deep	shallow	soft	soft	small	small
2	shallow	shallow	stiff	stiff	small	small
3	shallow	shallow	stiff	stiff	small	large
4	shallow	shallow	soft	soft	medium	medium
5	shallow	shallow	stiff	stiff	medium	medium
6	shallow	shallow	soft	soft	medium	small
7	shallow	shallow	soft	soft	large	large
8	deep	deep	soft	soft	large	large
9	shallow	shallow	soft	soft	large	medium
10	deep	deep	soft	stiff	large	large
11	shallow	shallow	stiff	stiff	large	large
12	shallow	shallow	stiff	stiff	large	small
13	shallow	shallow	stiff	stiff	large	large
Accuracy	0.92		0.92		0.69	

Table 4.12: CNN classification results for the selected samples

Sample #	True Values			CNN Classified		
	Depth, mm	Stiffness, kPa	Size, mm	Depth	Stiffness	Size
1	5	250000	15.99	Shallow	Stiff	Med
2	10	250000	11.92	Deep	Stiff	Small
3	10	250000	15.99	Deep	Stiff	Med
4	10	250000	23.02	Deep	Stiff	Large
5	5	169	20.92	Shallow	Soft	Large

Note: Green color indicates deep tumors.

Table 4.13: Size calculation results for the selected samples

Sample #	Calc. Size, mm (Mean±STD)	Size Error, % (Mean±STD)	Calc. Size, mm (Mean±STD)	Size Error, % (Mean±STD)
	Shallow		Deep	
Small Size Models Results				
1	12.99±0.49	18.76±3.08	13.94±0.24	12.82±1.53
2	6.13±0.18	48.60±1.52	10.75±0.10	9.82±0.88
3	7.42±0.39	53.58±2.42	11.36±0.16	28.98±1.00
4	15.88±0.53	31.00±2.29	15.24±0.27	33.78±1.15
5	5.68±0.24	72.85±1.16	10.59±0.13	49.36±0.62
5adj	13.68±0.24	34.61±1.16	18.59±0.13	11.12±0.62
Large Size Models Results				
1	15.60±0.17	2.46±1.07	17.29±0.23	8.13±1.43
2	13.41±0.05	12.50±0.44	14.29±0.09	19.88±0.73
3	13.82±0.16	13.55±0.98	14.86±0.18	7.05±1.10
4	16.39±0.15	28.82±0.65	18.44±0.16	19.91±0.71
5	13.27±0.08	36.58±0.36	14.12±0.11	32.49±0.51
5adj	21.27±0.08	1.66±0.36	22.12±0.11	5.75±0.51

Note: Bold font reveals the selected 3D interpolation model by CNN classification for size calculation. The red font shows the case of selection the sub-optimal 3D interpolation model.

CHAPTER 5

MULTISPECTRAL IMAGING PROBE

In this chapter, we describe the design and development of the Multispectral Imaging Probe to characterize the superficial properties of the potentially cancerous tissues. We developed a method to extract differential tissue properties from the sets of Multispectral Imaging Probe images by introducing pattern images or Multispectral Profile Diagrams.

5.1 Overview of the Modality

In Chapter 2, we gave an overview of breast imaging technologies, including near-infrared (NIR) and thermal imaging. The two methods use near-infrared or infrared (IR) light spectrum to image breast tissues. The advantages of these spectral methods are their non-ionizing nature and the ability to characterize the breast tissue metabolic properties.

Hyperspectral imaging is commonly used to image samples with high resolutions in the spectral and spatial domains. It requires significant acquisition time, special setup and conditions, as well as sophisticated and complex image analysis [76]. Because the tissue optical properties change when the tissue becomes malignant due to angiogenesis and hypermetabolic activity, hyperspectral imaging has great potential in cancer diagnostic applications. Tissue absorption in the near infrared region is the lowest, so the light can penetrate deepest into the tissue. For this reason, NIR imaging is the primary type of hyperspectral imaging for biological tissues [2, 77].

In our previous work, we used hyperspectral imaging for mammary tumor characterization [73]. The method evaluated levels of deoxyhemoglobin (*Hb*), oxy-hemoglobin (*HbO₂*), water (*H₂O*), and lipid chromophores within the tissue regions.

The typical absorption band of water is 970 nm from the second vibrational overtone of the O-H bond [78]. The typical absorption NIR band for lipids is 930 nm, which corresponds to the second overtone of C-H stretching [78].

Pulse oximetry, optical tomography, and optical mammography methods measure amounts

of deoxy- and oxy-hemoglobin chromophores [79, 80, 81]. Researchers use a range of NIR wavelengths to target each of the blood chromophores. The typical wavelengths to measure deoxy-hemoglobin are 660 nm, 670 nm, 760 nm or 780 nm. Oxy-hemoglobin is usually measured at 808 nm, 840 nm, or 940 nm wavelengths [79, 80, 81].

For the current work, we build on our expertise in hyperspectral imaging of mammary tumors and use a multispectral imaging modality for breast tumor tissue characterization. Instead of imaging the whole NIR spectrum, we target the mentioned chromophores by tuning the imaging system to the chromophore-revealing wavelengths. We verify the target wavelengths proposed in the literature and select the wavelengths for MIP acquisitions. The test results are discussed later in this document.

5.2 Hardware Design and Imaging Principles

Hyperspectral imaging collects and processes information for a scene from across the electromagnetic spectrum. Fig. 5.1 illustrates the difference between an RGB image and a hyperspectral image. The goal of hyperspectral imaging is to obtain the spectrum for each pixel in the image of the scene, to finding objects, identify materials, or detect metabolic changes.

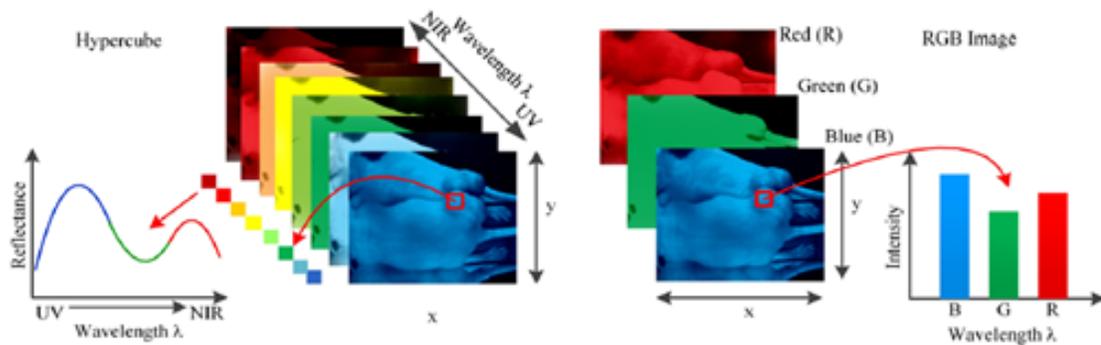


Figure 5.1: Hyperspectral image vs. bitmap image [2]

Biological tissues are composed of multiple molecular types, which cause their non-homogeneous optical properties. Light scatters and gets absorbed while traveling through the tissue (Fig. 5.2). The difference in refractive indices causes the light to be scattered within the tissue.

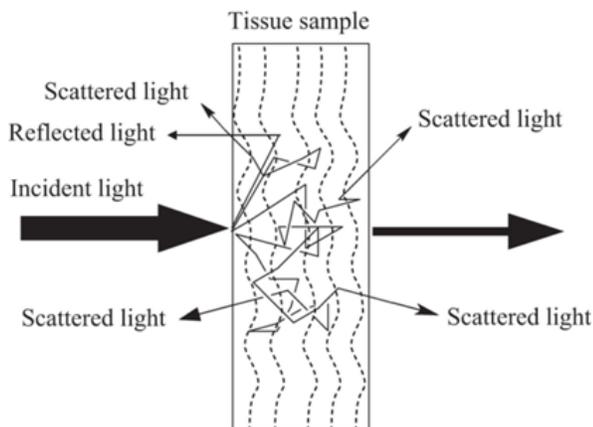


Figure 5.2: Light reflectance, scattering, and transmission within a tissue sample

The depth of light penetration is an important characteristic of optical measurements. The imaging depth depends on the tissue absorption properties [82]. Light absorption is at its minimum at the NIR range from 600 nm to 1300 nm for most living tissues (Fig. 5.3). In that range, the scattering is greater than absorption, so the light diffuses into the tissue the most and can reach the depth of up to 15 mm [2].

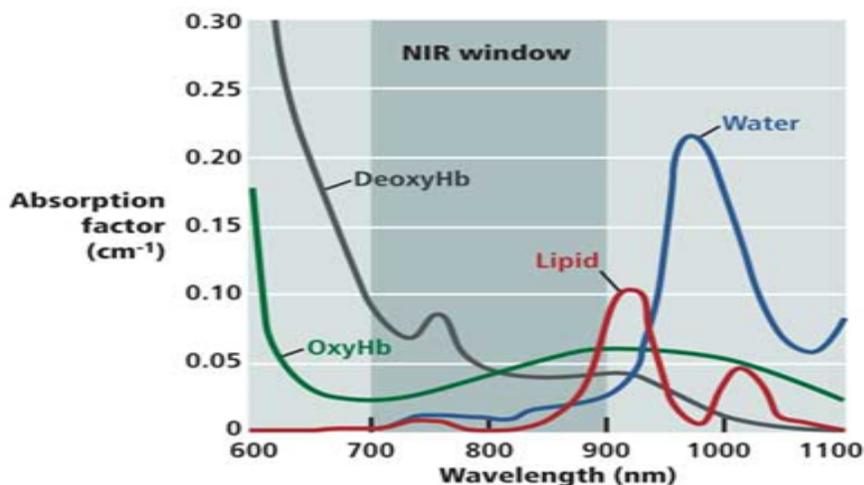


Figure 5.3: Absorption spectra of different tissue chromophores [3]

Chromophores are the tissue components that absorb light, and their absorbance depends on the wavelength as shown in Fig. 5.3. Light travels in a random path within tissue while scattering, which is the principle of diffused reflectance. The gradual histopathology changes of the unhealthy tissue will be captured by the reflectance spectra shape of that

region.

In this dissertation work, we design the acquisition hardware to make it less expensive and bulky than common hyperspectral imaging systems. Our design is adjusted to the breast imaging application. We shift from hyperspectral imaging (tenths of wavelengths) to the multispectral imaging idea to reduce the acquisition time and data storage requirements. We target the most significant deo-xyhemoglobin, oxy-hemoglobin, water, and lipids chromophores [73], instead of taking images for the entire NIR spectrum. The number of images to acquire, store, and analyze will decrease to only four per sample. We complete wavelengths selection experiments to identify the four wavelengths of interest from the suggested range of wavelengths presented in the literature [79, 80, 81] for this application. The results from the wavelengths selection imaging experiments are presented later in this document.

Next, we selected hardware components for the multispectral imaging system. For hyperspectral imaging of mammary tumors in our previous work [73], we used Retiga EXi F-M-12-C hyperspectral camera (QImaging, Surrey, BC, Canada) and a VeriSpec NIR liquid crystal tunable filter (LCTF) (Cambridge Research, and Instrumentation Inc., Woburn, MA, USA). The camera with the lens had an LCTF block, pictured in Fig. 5.4. Because the TIP video camera (IDS UI-3240CP-NIR) has NIR sensitivity, and the Xenoplan 1.4/23mm, 5-megapixel lens (Schneider Optics Inc., Hauppauge, NY) is designed to work with NIR spectra, we compared two cameras to assess the possibility of using TIP's camera and lens (Fig. 5.5) for multispectral imaging.

We also make sure that the two cameras have comparable quantum efficiency of their imaging sensors. Figure 5.6 shows the comparison of the quantum efficiency characteristics between the two sensors. The data about both sensors was re-plotted from the specification documentation for each camera [37, 83]. Table 5.1 shows a broader comparison of the two cameras' parameters, also taken from the specification documents. We can see from Table 5.1 that the key properties for the two cameras are very similar, and the imaging performance of the TIP's camera (IDS UI-3240CP-NIR) is comparable to the hyperspectral camera (Retiga EXi F-M-12-C) performance.

To further verify the idea of using IDS UI-3240CP-NIR camera for NIR imaging in



Figure 5.4: Hyperspectral camera acquisition system (Retiga EXi F-M-12-C)



Figure 5.5: TIP camera (IDS UI-3240CP-NIR) and lens

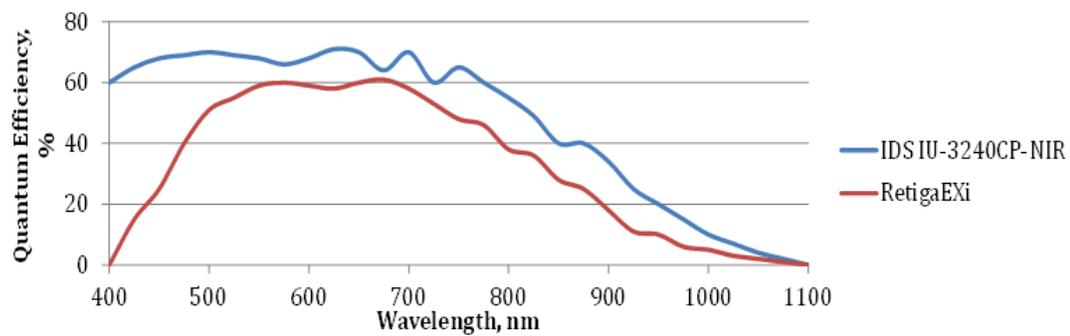


Figure 5.6: Comparative characteristic of the quantum efficiency curve for IDS UI-3240CP-NIR vs. Retiga EXi F-M-12-C

Table 5.1: Comparative characteristics of the two cameras (IDS UI-3240CP-NIR vs. Retiga EXi F-M-12-C)

Parameter	IDS UI-3240CP-NIR	Retiga EXi F-M-12-C
Sensor Type	CMOS	CCD
Resolution, μm	5.3×5.3	6.45×6.45
Max Image Size, <i>pixel</i>	$1,280 \times 1,024$	$1,390 \times 1,040$
Pixel Depth, <i>bit</i>	8	12
Acquisition Speed, <i>f/s</i>	60	10
Dimension, <i>mm</i>	29L \times 29W \times 29H	150L \times 76W \times 64H

our work, we complete an empirical comparison of the two cameras. Both cameras were focused on the same image sample. The illumination and distance settings were exactly the same for both setups. The setup of the experiment is shown in Fig. 5.7.

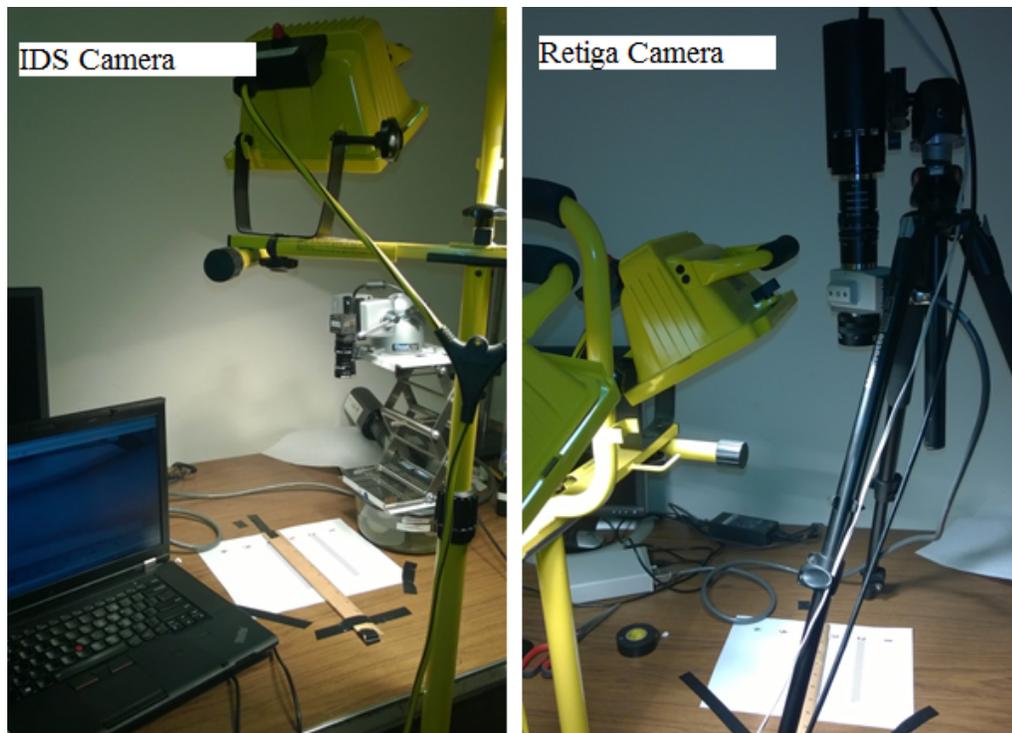


Figure 5.7: Two imaging systems setups

From the comparative research on the topic and the completed imaging experiment, we found the more compact and less expensive IDS UI-3240CP-NIR camera had very similar imaging performance to the performance of Retiga EXi F-M-12-C camera for breast tissue

imaging application. The spectral response and the pixel resolution are alike. The hyperspectral camera has the larger pixel depth and the larger image area size. The optics is better in the Retiga camera. Nevertheless, the resulting images from the experiments showed very good quality for both cameras at different tested wavelengths. The temperature conditions influenced the performance of both cameras and have to be taken into account during acquisitions by allowing appropriate time for the cameras to cool off during imaging experiments. It is reasonable to assume that the performance of IDS UI-3240CP-NIR camera will match the needs of the application.

The final Multispectral Imaging Probe design includes the TIP camera and the lens with four bandpass filters attached to it using a filter wheel (Thorlabs Inc., Newton, NJ). The filters are the hard coated OD 4.0 10 nm bandpass filters (Edmund Optics Inc., Barrington, NJ). They will limit the imaged reflected light to the target wavelengths. Broad spectrum two quartz tungsten halogen lamps (500 W each) are used as light sources during experiments. Fig. 5.8 shows Multispectral Imaging Probe (MIP) components and setup.

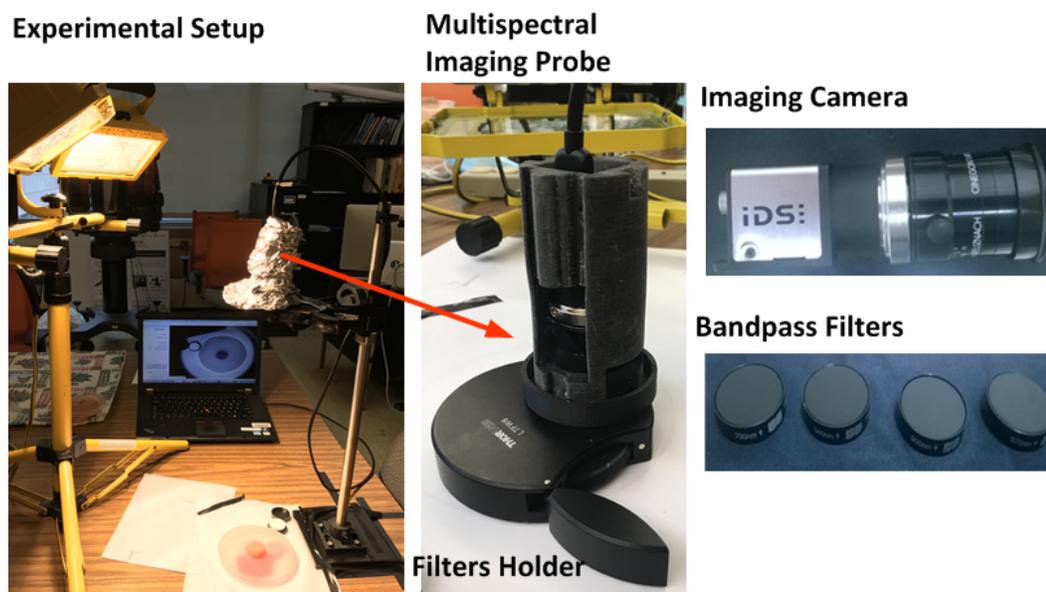


Figure 5.8: Multispectral Imaging Probe components and experimental setup

5.3 Acquisition Software

We create a MIP acquisition graphical user interface (GUI) by modifying the TIP GUI software in Qt (The Qt Company, Espoo, Finland) described in Chapter 3 to meet the specific needs of multispectral imaging experiments. Using the modified GUI software (tactile-qt-gui_v5_Temple_Experiment_Multispectral), we optimize the device operation procedures and acquisition time. Screenshots for the Setup tab and Capture tab views of the GUI are shown in Fig. 5.9.

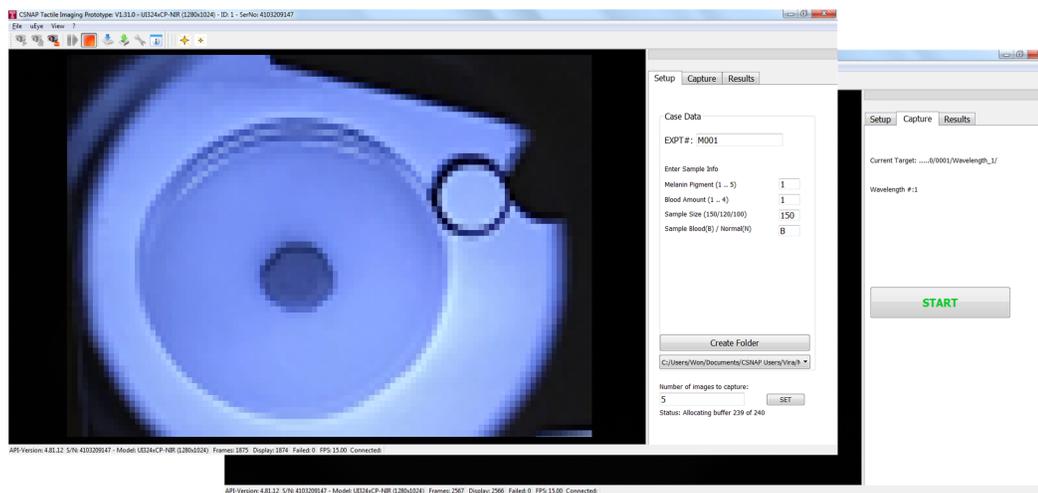


Figure 5.9: Multispectral Imaging Probe graphical user interface

On the Setup tab, MIP operator can specify the test sample information within GUI, the directory where to save the imaging data, and the number of images to acquire in one press of the acquisition button. This specified number of images (five on the screen shot) is equivalent to the number of experimental trials.

The Capture tab shows the imaging wavelength, target directory information and contains the Start button to initialize the acquisition for each wavelength. One imaging session corresponds to one of the selected wavelengths. The software default is four imaging sessions for each of the four selected wavelengths. The session name changes automatically from 1 to 4 after each acquisition of a trial set to reflect the imaging wavelength. Each wavelength set of trial images with the corresponding acquisition information are automatically saved in a directory with four sub-folders corresponding to each wavelength.

5.4 Imaging Wavelengths Selection

We complete experiments to select the four imaging wavelengths for Multispectral Imaging Probe. We empirically confirm the suggested in the literature bands of peak absorption for deo-xyhemoglobin (Hb), oxy-hemoglobin (HbO_2), water (H_2O), and lipid chromophores [78, 79, 80, 81, 84]. Based on these experiments, we select four optical bandpass filters to use with the MIP system for acquisitions.

Fig. 5.10 shows the absorption vs. wavelength spectra of the four key chromophores (Hb , HbO_2 , H_2O , and lipid) available in the literature [80]. To identify the Hb and HbO_2

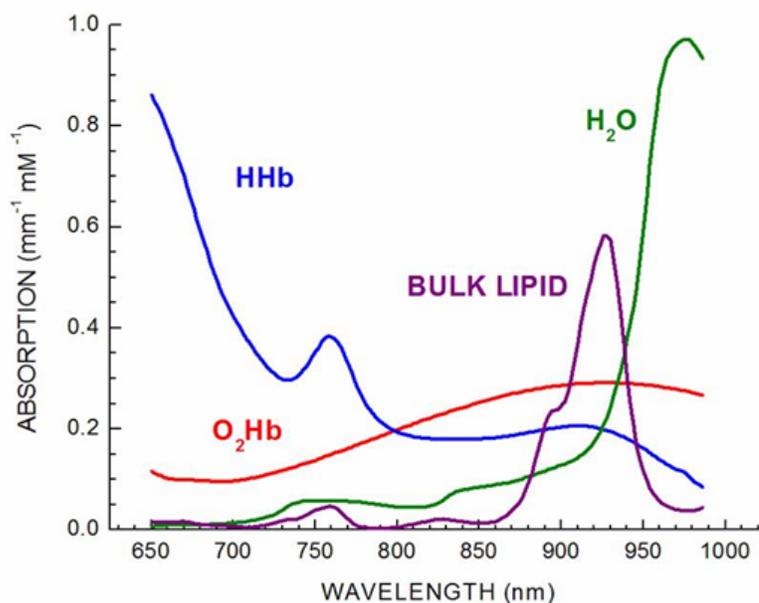


Figure 5.10: Spectra of the key tissue absorbers in NIR window

absorption peaks, we used two samples of blood: an *ex-vivo* venous bovine blood sample, and an *in-vivo* venous human blood sample. The bovine blood was used to fabricate our multispectral phantoms for experimental imaging with MIP. Therefore a sample of it was tested in this experiment to identify hemoglobin peaks. Hand veins of a human volunteer were imaged to obtain human blood hemoglobin peaks *in-vivo* from the selected vein regions to compare results with bovine hemoglobin peaks, to identify the oxy-hemoglobin peak, which is very difficult to implement in a phantom and to facilitate future MIP clini-

cal trials. We also tested the water and porcine lard samples to identify the corresponding absorption peaks of H_2O and lipid. Additionally, we tested a PDMS sample to characterize the absorption spectrum of the phantom material.

For the wavelengths selection experiments, we use Retiga EXi F-M-12-C hyperspectral imaging system with VeriSpec NIR liquid crystal tunable filter to image the samples and to characterize their absorption responses. We take hyperspectral images for each 10 nm band from 650 nm to 1100 nm. Two quartz tungsten halogen lamps (500 W each) are used for tissue samples illumination. Visualization of images and analysis of the spectral data is performed using ENVI v4.5 (Exelis Visual Information Solutions, Boulder, CO, USA) and MATLAB (The MathWorks, Inc., Natick, MA, US). Results from these experiments are presented in Chapter 6.

5.5 Optical Properties Estimation with Multispectral Imaging Probe

We describe the method to use Multispectral Imaging Probe (MIP) for breast tissue superficial optical property estimation.

5.5.1 Image Pre-processing

Image pre-processing in this work involves data normalization and image registration. The normalized multispectral images are aligned to improve the optical properties characterization performance of MIP.

Image Normalization

Reflectance normalization is done to normalize the raw acquisition image data to the reflectance or absorption values. Normalization decreases the effect of the inhomogeneous illumination of the imaged region and the hardware-related noise, such as the dark current of the imaging sensor. The dark current level increases with temperature and integration time.

In our test setup, we acquire one additional image for normalization purposes. The image is taken with a closed lens, where no external light should reach the sensor. This image will characterize the dark current level in the imaging sensor. We also placed a

white diffuse reflectance target (National Institute of Standards and Technology certified 99% Spectralon white diffuse reflectance target) near each imaging sample during experiments. The intensity information from the Spectralon standard area is used to convert raw multispectral images, I_{raw} , to normalized reflectance images, I_{norm} .

All four multispectral images from one test sample set have to be pre-processed and normalized. The conversion from raw images to normalized images is done pixel by pixel using the following equation [2, 73].

$$I_{norm} = \frac{I_{raw} - I_{dark}}{I_{white} - I_{dark}}, \quad (5.1)$$

where I_{raw} is the reflectance intensity from the raw multispectral images obtained during the experiments, I_{white} represents pixel intensity of the white reflectance standard, I_{dark} is the black current pixel intensity.

Image Registration

Image registration searches for the geometric transformation of multiple images of the same scene to align them. In our case, we align four images that captured reflectances in the four selected wavelength bands. Image registration is a necessary step for multispectral image alignment and analysis [85, 86, 87]. We are mostly interested in translational distortions due to the nature of MIP tests. We use Matlab image registration tools equipped with *Sobel* filters to carry out the proper alignment of four multispectral images in one set.

5.5.2 Multispectral Profile Diagram

To characterize the superficial optical properties of the breast tissue, we will implement the same idea that was developed for the sets of TIP images. In this section, we describe how we can construct Multispectral Profile Diagrams (MPD) from the sets of pre-processed (normalized and aligned) multispectral images. MPDs carry unique information about the optical properties of breast tissue from four imaging bands fused in one pattern image. The wavelength bands are selected to target the chromophores of interest in breast cancer application, such as deoxy-hemoglobin, oxy-hemoglobin, lipids, and water.

Hyperspectral and multispectral image fusion is an active area of research. Hyperspectral images are high dimensional and characterized by information redundancy and

sparsity. Therefore, one of the major issues that have to be solved to analyze hyperspectral and multispectral images is feature reduction or extraction [88]. Multiple methods for feature extraction are developed by scientists for various applications, such as remote sensing, biometrics, and medical diagnostic [88, 89, 90, 91, 92]. The feature extraction and classification methods are divided on non-supervised, supervised, and semi-supervised techniques [88]. Deep learning methods for classification and feature extraction are especially under consideration for the task [93]. Some methods incorporate knowledge about human visual perception to extract features for better recognition performance [89]. However, all of these methods do not account for the application context information in the data. To the best of our knowledge, there is no method available in hyperspectral image analysis to incorporate the tissue optical index calculation into the dimensional reduction or multimodal image fusion.

From our previous research developments, we know that a higher total hemoglobin content indicates higher tissue blood volume and malignancy. Higher water content suggests malignancy as well. Decreased lipid content indicates that the parenchymal adipose tissue has been displaced, which is a warning sign of malignant tumors [73]. The reflectance changes of these chromophores are combined into Tissue Optical Index (TOI) formulation expressed via Equation (5.2). The lower TOI values suggest high metabolic activity and the increased probability of malignancy [73].

$$TOI = \frac{R_{H_2O}(R_{Hb} + R_{HbO_2})}{R_{lipid}}, \quad (5.2)$$

where R_{Hb} is the total reflectance of deoxy-hemoglobin chromophores, R_{HbO_2} denotes the total reflectance of oxy-hemoglobin chromophores, R_{H_2O} is the reflectance of water, and R_{lipid} expresses the reflection of lipids within the imaged tissue.

The tissue optical index was calculated for the manually segmented multipixel patch within the imaged tissue region. In this work, we construct a Multispectral Profile Diagram from the four multispectral images and calculate TOI for each pixel in the aligned image set. It helps to extract the high resolution information on the chromophores' concentration automatically without manual region selection for each hyperspectral imaging set in ENVI software.

After MIP multispectral images are pre-processed and aligned, we create a Multispectral Profile Diagram for the imaged tissue sample. Each pixel of the Multispectral Profile Diagram is calculated using Equation (5.3). When calculations are done for each pixel location, the Multispectral image is made and shown in Fig. 5.11.

$$MPD_{M \times N} = \begin{bmatrix} \frac{c_{1,1}(a_{1,1}+b_{1,1})}{d_{1,1}}, & \frac{c_{1,2}(a_{1,2}+b_{1,2})}{d_{1,2}}, & \dots & \frac{c_{1,N}(a_{1,N}+b_{1,N})}{d_{1,N}} \\ \frac{c_{2,1}(a_{2,1}+b_{2,1})}{d_{2,1}}, & \frac{c_{2,2}(a_{2,2}+b_{2,2})}{d_{2,2}}, & \dots & \frac{c_{2,N}(a_{2,N}+b_{2,N})}{d_{2,N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{c_{M,1}(a_{M,1}+b_{M,1})}{d_{M,1}}, & \frac{c_{M,2}(a_{M,2}+b_{M,2})}{d_{M,2}}, & \dots & \frac{c_{M,N}(a_{M,N}+b_{M,N})}{d_{M,N}} \end{bmatrix}, \quad (5.3)$$

where $a_{m,n}$ corresponds to the multispectral image taken to capture Hb concentration, $b_{m,n}$ is the multispectral image taken to capture HbO_2 concentration, $c_{m,n}$ corresponds to the multispectral image taken to capture $Water$ concentration, and $d_{m,n}$ is the multispectral image taken to capture $Lipid$ concentration. $m = 1, 2, \dots, M$ is the image row number, $n = 1, 2, \dots, N$ is the image column. The size of a reconstructed MPD ($M \times N$) is the same size as individually acquired multispectral images ($1280 \text{ pixels} \times 1024 \text{ pixels}$).

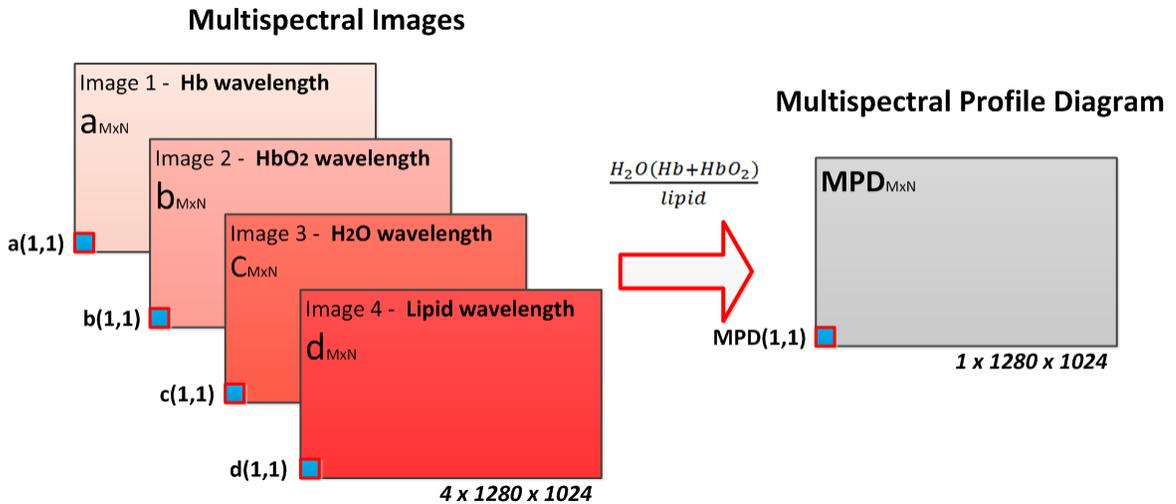


Figure 5.11: Construction of a Multispectral Profile Diagram

We present an example of a Multispectral Profile Diagram in Fig. 5.12. The MPD image was reconstructed from a set of four multispectral images using Equation (5.3). The figure also shows a matching photograph of the sample and the corresponding MPD in false color.

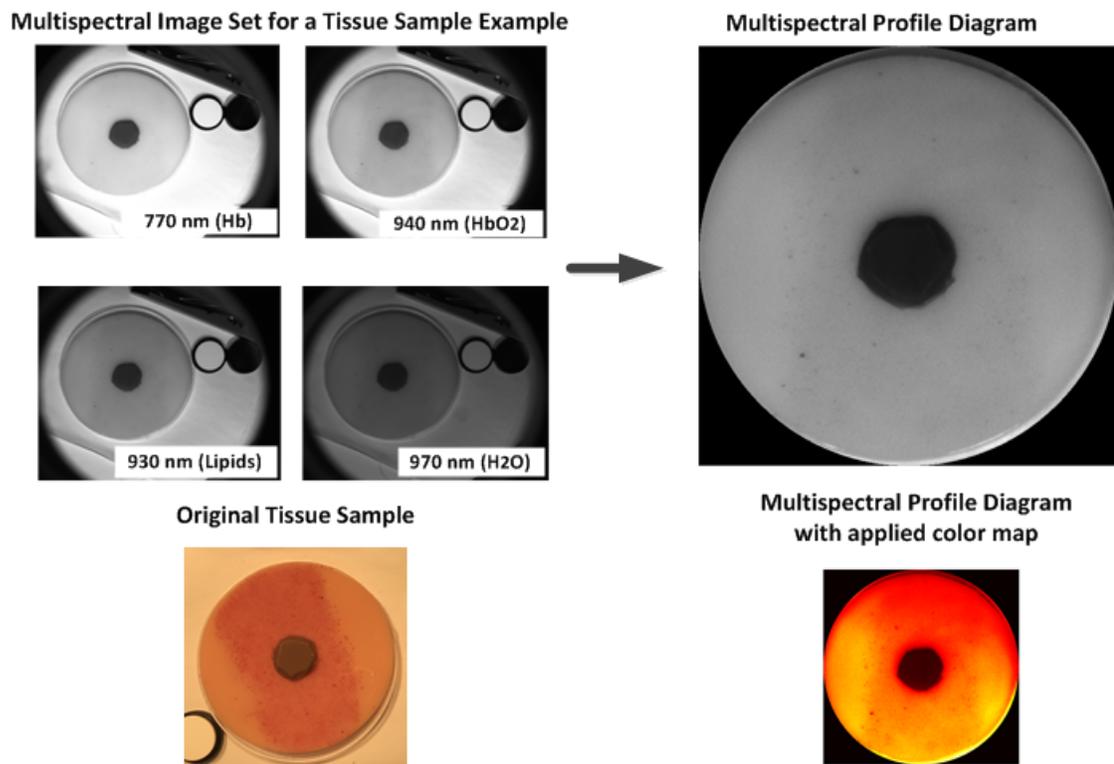


Figure 5.12: An example of Multispectral Profile Diagram

Each pixel on MPD pattern image carries unique information about the tissue properties. The mimicked malignant region (in red) within the original sample, translates to the area with lower values of MPD pixels (Fig. 5.13). This matches with the assumptions of lower TOI values for malignant tissue regions in our previous work [73].

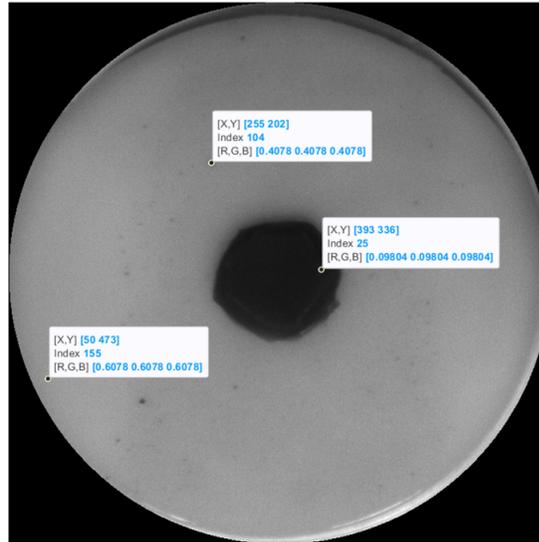


Figure 5.13: Multispectral Profile Diagram data points

Optical tissue property can also be used for other applications, for example, automatic tissue segmentation, morphology classification, or segmentation. Material optical property can be similarly incorporated into material characterization/classification, fault detection, and remote sensing by tuning the imaging bands for the specific application.

5.5.3 *Multispectral Profile Diagrams for Differential Tissue Properties Estimation*

To characterize the superficial optical properties of breast tissue, we encode the broad imaging information via Multispectral Profile Diagrams, which are application-specific pattern images. Next, we develop a method to find the following parameters: differential asymmetry, differential texture, and differential inflammation from a set of two Multispectral Profile Diagrams. These parameters showed potential in classifying the tumors as malignant or benign, which allows doctors to decide the stage of the disease [10, 18, 19, 20, 94].

The differential parameters reflect the difference between the normal breast and the diseased breast tissue in terms of lateral asymmetry, skin texture, and local inflammation.

The differential MPD idea is similar to the background normalization idea. Because all of the patients have distinct anatomy of normal breast tissues, the abnormal tissues will have differing manifestations. Therefore, tissue normalization is crucial for abnormality detection in clinical applications, especially IBC.

To measure the differential parameters, we propose the Differential MPD method. A Differential MPD image, MPD_{diff} , is constructed by taking a weighted absolute difference between the MPD of the affected breast, $MPD_{Abnormal}$, and the smoothed MPD image of the normal (or reference) breast of the same subject, $MPD_{SNormal}$, using Equation (5.4). We present an example of differential MPD image and its component MPD images in Fig. 5.14.

$$MPD_{diff} = |K \times MPD_{Abnormal} - MPD_{SNormal}|, \quad (5.4)$$

where K is an empirical constant coefficient.

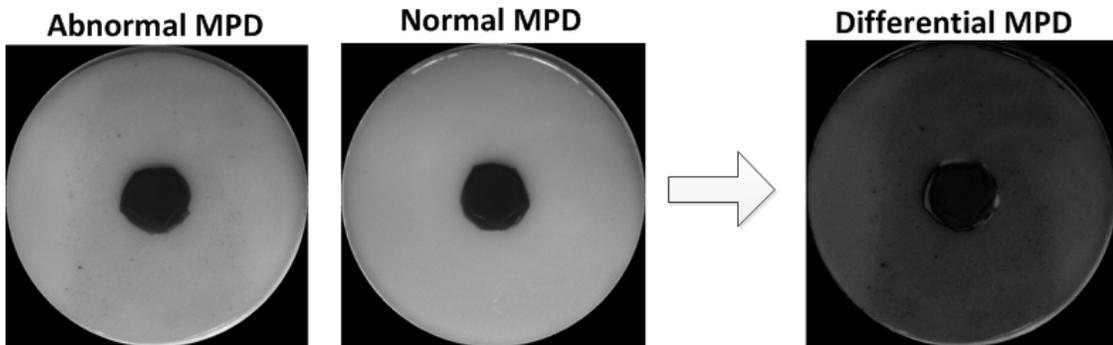


Figure 5.14: Formation of a Differential Multispectral Profile Diagram

To emphasize the difference texture features on Differential MPD images, we applied a custom edge enhancement filter described as Equation (5.5) to the Differential MPDs.

$$h = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -8 \end{bmatrix}. \quad (5.5)$$

A filter, h , or the 5×5 neighborhood of point A on an image $MPD_{diff}(x,y)$ in a spatial domain, is moved through the image to compute an output texture-enhanced image [86]. The visualization of the image filtering process is shown in Fig. 5.15. The filter is moved

along the traced path (gray line in the figure), and the texture-enhanced $MPD_{diff}(x,y)$ image is computed.

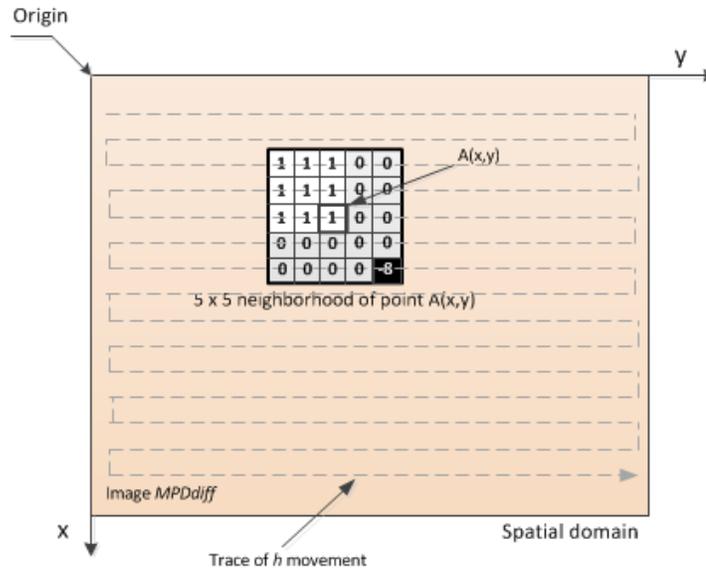


Figure 5.15: Image filtering in the spatial domain

Two examples of texture-enhanced Differential MPD images are shown in Fig. 5.16. Using this technique, we emphasize the tissue surface texture (in our case, pitted vs. not pitted skin texture) and preserve the image intensity information for inflammation detection at the same time. This step is crucial for the texture classification task.

5.5.4 Segmentation of Differential Multispectral Profile Diagrams

For IBC assessment, it is important to measure the percentage of the inflammation-affected area from a Differential Multispectral Profile Diagram. The affected region shows as darker pixels than the pixels of the surrounding tissue; therefore, it is easy to use MATLAB-assisted segmentation tool to select the region and to compare it with the total tissue area. The manual segmentation example is shown in Fig. 5.17.

5.6 Classification of Multispectral Profile Diagrams

In this section, we outline the method to classify and quantify the differential parameters from Multispectral Profile Diagrams (MPD) using CNN classification models. Fig-

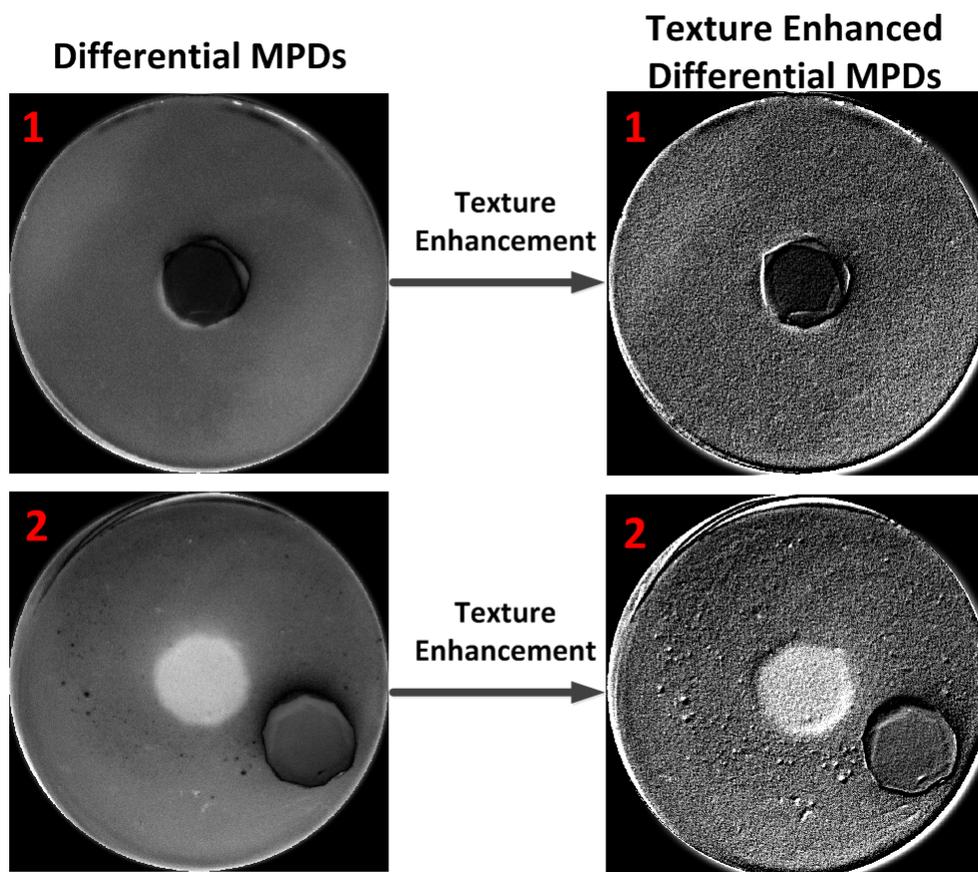


Figure 5.16: Two examples of Differential Multispectral Profile Diagram texture-enhancement

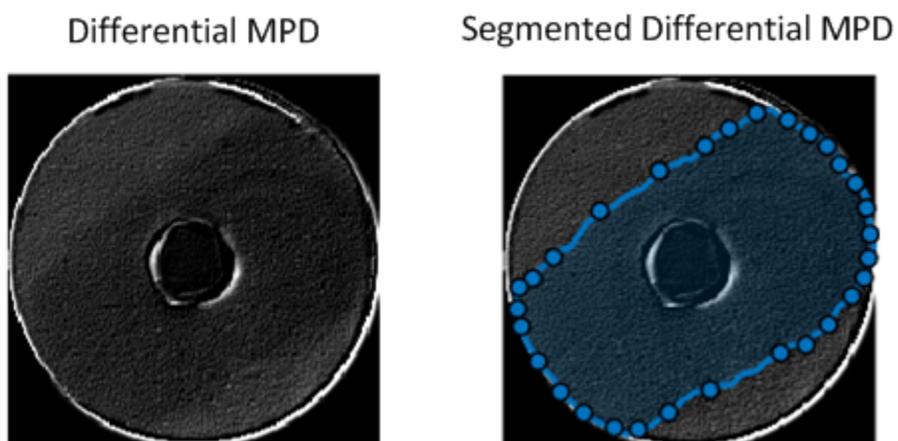


Figure 5.17: Inflammation segmentation on a Differential Multispectral Profile Diagram

Figure 5.18 shows the steps we perform for classification of Differential Multispectral Profile Diagrams.

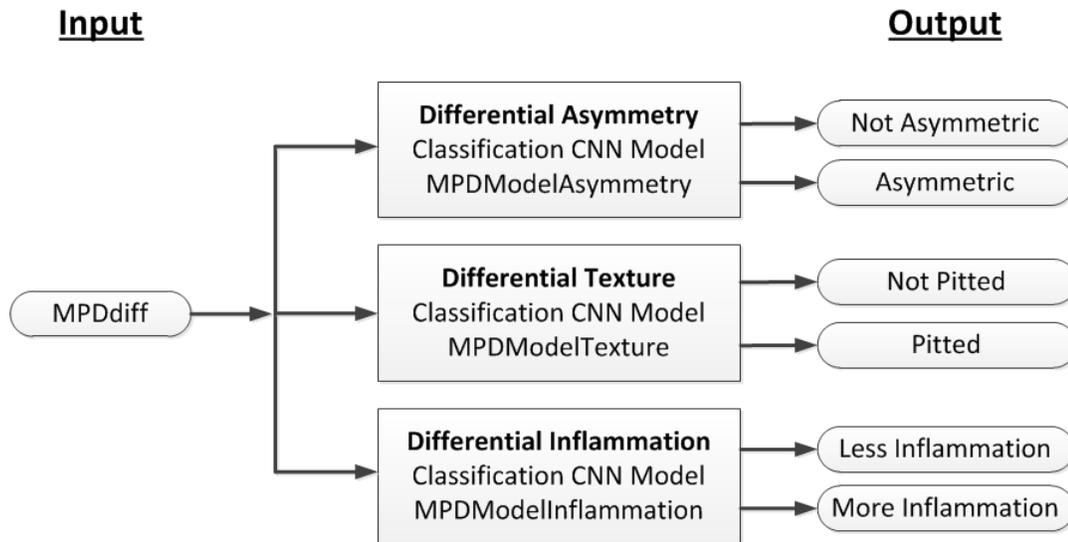


Figure 5.18: Classification of Differential Multispectral Profile Diagrams

We develop a CNN classification model for each of the three differential parameters: asymmetry, texture, and inflammation. We built the CNN models by utilizing structure and development suggestions from [63, 95]. We trained MPDModelAsymmetry, MPDModelTexture, and MPDModelInflammation on asymmetry, texture, and inflammation texture-enhanced Differential MPD datasets, respectively.

Next, we describe each classification model separately. We will use class probabilities for asymmetry, texture, and inflammation from the classification to construct MIP Index.

5.6.1 Asymmetry Classification Model

The description of the asymmetry classification model is presented in Fig. 5.19 and Fig. 5.20. The model contains three convolutional layers with *ELU* activations, and *Max pooling* layers for feature extraction. The four fully connected layers are added for the classification task. We utilize *Dropout* layers to improve the classification accuracy of the model. We employ *Adamax* optimizer, and *Accuracy* as the metric for the model performance evaluation during training. The final fully connected layer has *Softmax* activation function and

two nodes, corresponding to the Symmetric and Assymmetric classes. The Python script for the model development is given in Appendix F.

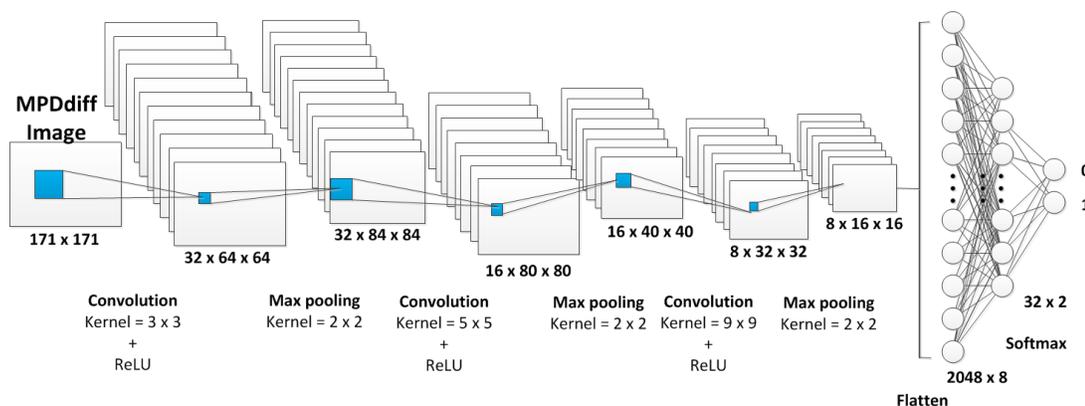


Figure 5.19: Asymmetry classification model structure (MPDModelAsymmetry)

The data for training and validation of the model is developed from multispectral phantom imaging data combined into Differential Multispectral Profile Diagrams dataset for asymmetry classification. The dataset development and model training are described in Chapter 6.

5.6.2 Texture Classification Model

Fig. 5.21 and Fig. 5.22 outline the texture classification model structure. The model consists of three convolutional layers, including one complex layer, *Max pooling* layers, and *ELU* activations for the feature extraction. The four fully connected layers are responsible for the classification. We use normalization and *Dropout* layers to improve the performance of the model. We employ *Adamax* optimizer, and *Accuracy* as the metric for the model performance evaluation during training. The final fully connected layer has *Softmax* activation function and two nodes, corresponding to the Not Pitted and Pitted classes. The Python script for the model development is given in Appendix F.

The multispectral phantom imaging data was used for training and validation of the model. Multispectral images were combined into Differential Multispectral Profile Diagrams dataset for texture classification. The dataset development and model training are described in Chapter 6.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 169, 169, 32)	320
activation (Activation)	(None, 169, 169, 32)	0
max_pooling2d (MaxPooling2D)	(None, 84, 84, 32)	0
conv2d_1 (Conv2D)	(None, 80, 80, 16)	12816
activation_1 (Activation)	(None, 80, 80, 16)	0
max_pooling2d_1 (MaxPooling2D)	(None, 40, 40, 16)	0
conv2d_2 (Conv2D)	(None, 32, 32, 8)	10376
activation_2 (Activation)	(None, 32, 32, 8)	0
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 8)	0
flatten (Flatten)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 8)	16392
activation_3 (Activation)	(None, 8)	0
dropout_1 (Dropout)	(None, 8)	0
dense_1 (Dense)	(None, 16)	144
activation_4 (Activation)	(None, 16)	0
dropout_2 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 32)	544
activation_5 (Activation)	(None, 32)	0
dropout_3 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 2)	66
activation_6 (Activation)	(None, 2)	0

Total params: 40,658
 Trainable params: 40,658
 Non-trainable params: 0

Figure 5.20: Asymmetry classification model details

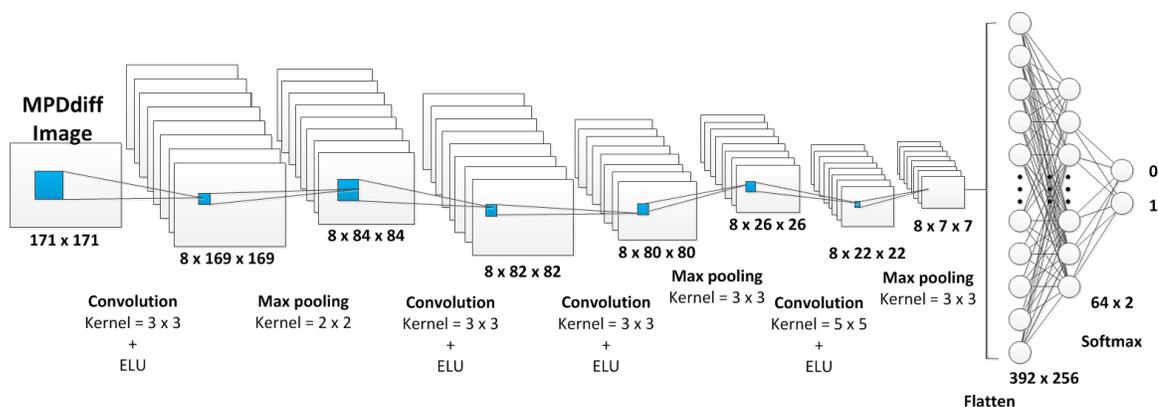


Figure 5.21: Texture classification model structure (MPDModelTexture)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 169, 169, 8)	80
activation (Activation)	(None, 169, 169, 8)	0
batch_normalization (Batch Normalization)	(None, 169, 169, 8)	32
max_pooling2d (MaxPooling2D)	(None, 84, 84, 8)	0
conv2d_1 (Conv2D)	(None, 82, 82, 8)	584
activation_1 (Activation)	(None, 82, 82, 8)	0
conv2d_2 (Conv2D)	(None, 80, 80, 8)	584
activation_2 (Activation)	(None, 80, 80, 8)	0
batch_normalization_1 (Batch Normalization)	(None, 80, 80, 8)	32
max_pooling2d_1 (MaxPooling2D)	(None, 26, 26, 8)	0
conv2d_3 (Conv2D)	(None, 22, 22, 8)	1608
activation_3 (Activation)	(None, 22, 22, 8)	0
batch_normalization_2 (Batch Normalization)	(None, 22, 22, 8)	32
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 8)	0
flatten (Flatten)	(None, 392)	0
dropout (Dropout)	(None, 392)	0
dense (Dense)	(None, 256)	100608
activation_4 (Activation)	(None, 256)	0
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
activation_5 (Activation)	(None, 128)	0
batch_normalization_4 (Batch Normalization)	(None, 128)	512
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
activation_6 (Activation)	(None, 64)	0
batch_normalization_5 (Batch Normalization)	(None, 64)	256
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 2)	130
activation_7 (Activation)	(None, 2)	0
Total params: 146,634		
Trainable params: 145,690		
Non-trainable params: 944		

Figure 5.22: Texture classification model details

5.6.3 Inflammation Classification Model

Fig. 5.23 and Fig. 5.24 present the proposed model description for inflammation classification. The model includes four convolutional layers, including three complex layer structures, *ELU* activations, and *Max pooling* layers for feature extraction. The three fully connected layers are added for the classification task. We employ *Adamax* optimizer, and *Accuracy* as the metric for the model performance evaluation during training. The final fully connected layer has *Softmax* activation function and two nodes, corresponding to the Less inflammation and More inflammation classes. The Python script for the model development is given in Appendix F.

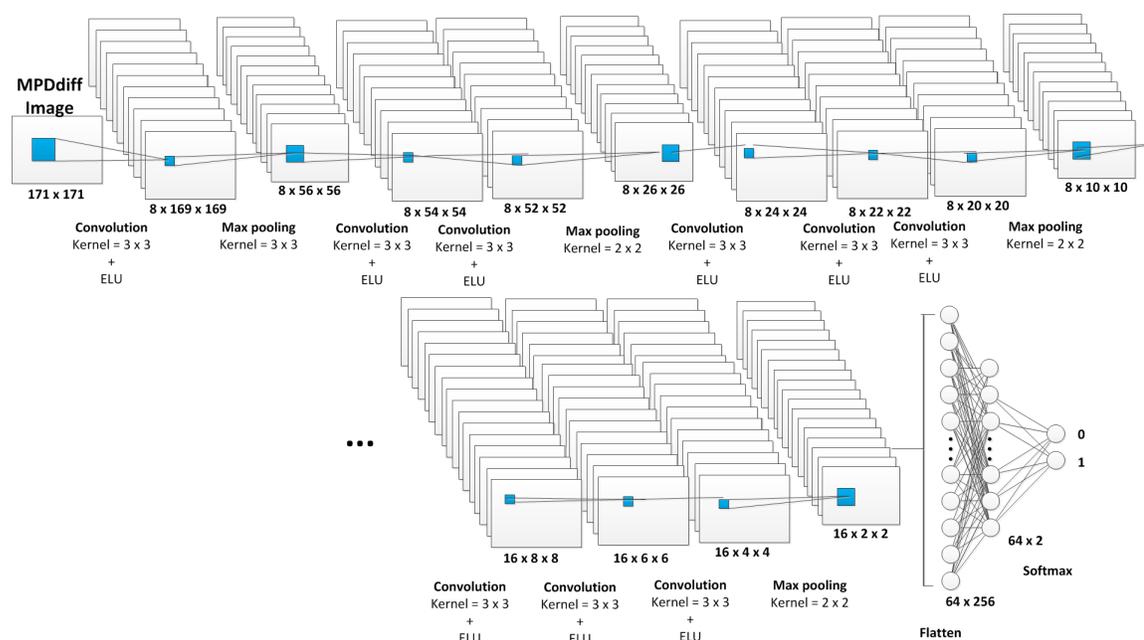


Figure 5.23: Inflammation classification model structure (MPDModelInflammation)

We used the multispectral phantom imaging data to create Differential Multispectral Profile Diagrams dataset for inflammation classification model development. The dataset was split independently on training and validation subsets for training and validation of the model, respectively. The dataset development and model training are described in Chapter 6.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 169, 169, 8)	80
activation (Activation)	(None, 169, 169, 8)	0
batch_normalization (Batch Normalization)	(None, 169, 169, 8)	32
max_pooling2d (MaxPooling2D)	(None, 56, 56, 8)	0
conv2d_1 (Conv2D)	(None, 54, 54, 8)	584
activation_1 (Activation)	(None, 54, 54, 8)	0
conv2d_2 (Conv2D)	(None, 52, 52, 8)	584
activation_2 (Activation)	(None, 52, 52, 8)	0
batch_normalization_1 (Batch Normalization)	(None, 52, 52, 8)	32
max_pooling2d_1 (MaxPooling2D)	(None, 26, 26, 8)	0
conv2d_3 (Conv2D)	(None, 24, 24, 8)	584
activation_3 (Activation)	(None, 24, 24, 8)	0
conv2d_4 (Conv2D)	(None, 22, 22, 8)	584
activation_4 (Activation)	(None, 22, 22, 8)	0
conv2d_5 (Conv2D)	(None, 20, 20, 8)	584
activation_5 (Activation)	(None, 20, 20, 8)	0
batch_normalization_2 (Batch Normalization)	(None, 20, 20, 8)	32
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 8)	0
conv2d_6 (Conv2D)	(None, 8, 8, 16)	1168
activation_6 (Activation)	(None, 8, 8, 16)	0
conv2d_7 (Conv2D)	(None, 6, 6, 16)	2320
activation_7 (Activation)	(None, 6, 6, 16)	0
conv2d_8 (Conv2D)	(None, 4, 4, 16)	2320
activation_8 (Activation)	(None, 4, 4, 16)	0
batch_normalization_3 (Batch Normalization)	(None, 4, 4, 16)	64
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 16)	0
flatten (Flatten)	(None, 64)	0
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 256)	16640
activation_9 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 64)	16448
activation_10 (Activation)	(None, 64)	0
batch_normalization_4 (Batch Normalization)	(None, 64)	256
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130
activation_11 (Activation)	(None, 2)	0

Figure 5.24: Inflammation classification model details

5.7 Summary

In this chapter, we described the Multispectral Imaging Probe development. We outlined its hardware design, the principle of operation, and the imaging data analysis strategies for MIP imaging modality.

We presented a method to fuse optical tissue-characterizing information from four multispectral images into one application-meaningful pattern image, a Multispectral Profile Diagram. We also described the reasoning and development of Differential Multispectral Profile Diagrams to characterize breast tissues with attention to individual anatomical and physiological differences in patients and to aid IBC pre-screening.

Finally, we propose to classify Multispectral Profile Diagrams using Convolutional Neural Network models to extract valuable information about breast tissue asymmetry, texture, and inflammation automatically. The classification results from these three models are used to calculate Spectral Index, and further, Multimodal Index for breast cancer risk assessment.

CHAPTER 6

MULTISPECTRAL IMAGING PROBE EXPERIMENTS

In this chapter, we describe the multispectral imaging experiments and present our results. We also introduce a custom tissue phantom, which mimics superficial optical properties of breast tissues. These phantoms help us train and test the classification models to characterize differential asymmetry, texture, and inflammation parameters. This phantom can be combined with the mechanical properties phantom for bimodal imaging experiments.

6.1 Optical Property Phantom

Here, we complete a short survey on the available tissue phantoms, which mimic the optical properties of human tissues. We describe the design of the optical phantom and describe how the phantoms are fabricated for our application.

6.1.1 Review of Optical Property Phantoms

Fluid-based phantoms are very convenient and commonly used to mimic the optical property of living tissues in lab settings. They are relatively easy to fabricate and to make changes to their property. The majority of the liquid phantoms are water-based, with the addition of scattering and absorption agents [96]. However, liquid phantoms need to be in a secure container, and they are difficult to transport. In addition, the liquid phantom mixture can be unstable, with the larger particles concentrated at the bottom of the container [69].

To overcome these difficulties, researchers are developing solid phantoms from a variety of materials, such as Delrin plastic, PDMS, wax, polyester resin, epoxy resin, agar gel, gelatin, and fibrin. The absorption and scattering agents that are used to fabricate the solid phantoms are intralipid, hemoglobin, dyes, titanium dioxide, carbon black, ferric chloride, and silica spheres [64, 65, 66, 67, 69, 70, 97].

Unfortunately, hydrogels (agar and gelatin) are not durable, they cannot withstand mechanical compressions, and they melt at room temperature. The difficulty with silicone

phantoms such as PDMS is in the challenge of adding organic additives. Fibrin can be used for optical property phantoms because it is a naturally occurring protein in humans, and it can be easily mixed with different types of scattering and absorption agents [32]. Also, Polyvinyl alcohol (PVA) is commonly used for many types of optical imaging phantoms [32].

Some of the phantoms are able to mimic several types of tissue property - multimodal imaging phantoms. Lamouche *et al.* [64] provide a review of the possible materials for optical coherence tomography phantom, which has to possess the optical and mechanical properties of tissues. The study found that no one material is perfect for the application, yet several of them can be implemented. They state the advantages and disadvantages of the phantom materials. They suggest using silicone if a volumetric phantom will be made, yet prefer fibrin when adding organic agents is necessary. They also mentioned that PVA phantoms could provide the ability to tune the mechanical property of the phantom, yet the adding of optical characteristics to such phantom is difficult. However, in another study, researchers incorporate optical property in the PVA phantom to calibrate an optical tomograph [69].

Researchers also mimicked the optical and acoustic property of tissues with Polyvinyl chloride (PVC), titanium dioxide, and a black plastic coloring agent [98].

Price *et al.* [32] were able to combine mechanical and x-ray attenuation properties in phantom. They developed a new phantom material, which was a solution of Polyvinyl alcohol in ethanol and water. The material underwent freeze-thaw cycles to obtain the desirable elastic property. Depending on the firmness of the material, it had varying x-ray attenuation properties.

One of the latest multimodal phantoms with tunable mechanical, optical, and acoustic properties was developed from multiple compounds (gelatin, agar, PDMS silicone) and additives (blood mimicking fluid, bovine blood serum, intralipids, microbeads, India Ink and dyes) [99]. In the study, the authors proposed a multilayered design with the ability to change the property as desired by adding or removing phantom layers.

6.1.2 Phantom Development

Multispectral Imaging Probe and its algorithms are developed to characterize the superficial optical property of tissues, which can help distinguish cancer from benign tissue. The targeted features, such as breast tissue differential asymmetry, differential texture, and differential inflammation size, skin pigmentation level, and depth of the inflammation, are implemented in the phantom.

After reviewing the available literature on optical property phantoms [32, 64, 69, 99], and based on our tactile phantom fabrication experience from previous work [34, 100], we decided to create a multicomponent phantom from PDMS with mixed in organic and inorganic components. We complete a preliminary study of mixing a small amount of organic absorbers (bovine blood and porcine lard) with PDMS material prior to curing. The study shows that the mixture cured well, and we can use a small amount of the organic components with PDMS material.

For custom MIP phantom, we made multiple thin layers of silicone (Polydimethyl siloxane) with added artificial coloring agents (Smooth-On Inc., Macungie, PA), water, lipids, and bovine blood (Lampire Biological Laboratories, Pipersville, PA) to mimic normal breast tissue, and breast tissue with asymmetry, texture, and inflammation changes. Different amounts of coloring agents were added to the PDMS silicone layers to mimic different skin color tones in the phantom. Breast tissue inflammation depth was also implemented in our phantom. The reflectance variations due to PDMS concentration are experimentally validated in Section 6.2 and show the negligible effect on imaging reflectance spectra.

The design of the MIP phantom is shown in Fig. 6.1. We made multiple normal breast tissue samples and samples with the asymmetry, texture, and inflammations changes. Asymmetric changes in the breast phantom are introduced by placing the mimicked nipple in random locations on the phantom during imaging. To implement breast tissue texture changes, we pinch with small tweezers pit-like regions on the top of cured PDMS samples to mimic the skin texture changes described in the literature. Inflammation changes are implemented by adding different amounts of blood in varying patterns to the skin-colored

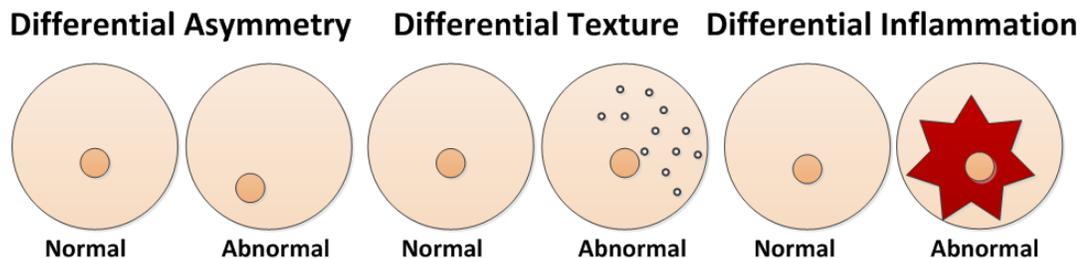


Figure 6.1: MIP phantom design

PDMS samples (abnormal samples), in contrast to the samples of skin color only (normal samples).

We made 240 abnormal inflammation pattern samples and 10 normal samples for classification training purposes. This amount of the test samples allows us to complete MIP experiments and collect a meaningful dataset for the application. Fig. 6.2 presents the description and amount of the phantom samples. We also fabricated a small representative set of four additional abnormal inflammation pattern samples and two normal samples using the third pigmentation level to test the developed algorithms.

Normal Tissue Samples

Skin Pigmentation

2 samples				
-----------	-----------	-----------	-----------	-----------

Abnormal Inflammation Pattern Tissue Samples

Skin Pigmentation

40 samples				
------------	------------	------------	------------	------------

Blood Amount

.15 ml 50 samples	0.25 ml 50 samples	0.5 ml 50 samples	0.75 ml 50 samples
----------------------	-----------------------	----------------------	-----------------------

10 samples 10 samples 10 samples 10 samples

Figure 6.2: Description and quantity of MIP samples

All phantom samples are 150 mm in diameter. Each of the 5 pigmentation levels has

40 abnormal inflammation pattern samples and 2 normal samples. Normal samples have no blood mixed in them. The abnormal inflammation samples have 4 levels of blood (0.15 ml, 0.25 ml, 0.5 ml, 0.75 ml) with 50 samples of each pigmentation level. Fig. 6.3 shows examples of the fabricated phantoms.

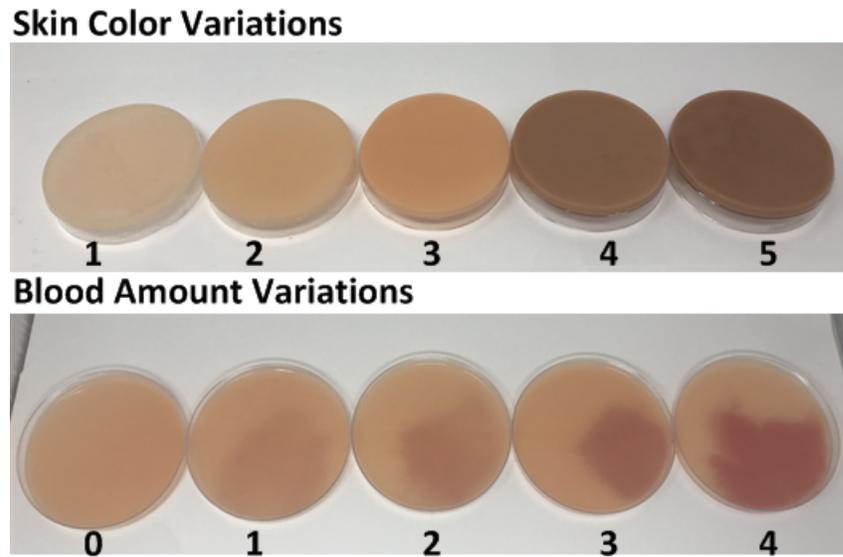


Figure 6.3: Skin tone and blood amount variations in MIP phantoms

To image differential parameters with MIP, such as asymmetry, texture, and inflammation changes, we consider sets of phantom samples. We build abnormal samples, which represent the diseased breasts. These samples include one or more abnormal features. Then we also build normal samples, which represent the normal breasts and no abnormal breast tissue changes.

In our experiments, normal and abnormal phantom samples are imaged separately with MIP in five imaging sessions (trials). We imaged phantoms with different combinations and extend of asymmetry, texture, and inflammation parameters implemented in them. There are three cases: no features implemented, or one feature implemented on one sample, or several features implemented on one phantom sample. Each normal and abnormal sample is imaged separately and then combined in a set for further analysis.

To mimic breast tissue inflammation changes, we create random patterns, while adding blood into the silicone phantoms (Fig. 6.4). We also implement the depth of inflammation parameter into our phantoms. During phantom fabrication, we notice that the mixed-in

blood tends to keep the pattern yet blood molecules gravitate toward the bottom of the sample before PDMS media is cured. That allows us to have 2 variations of depth (shallow: 0 mm and deep: 5 mm) for the same inflammation pattern phantom sample by imaging it from both sides.

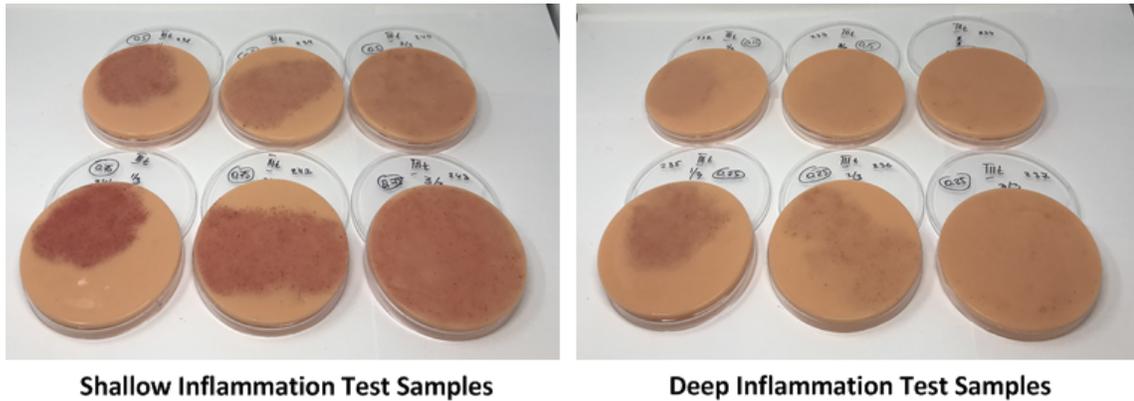


Figure 6.4: MIP phantom test patterns and depths

To mimic texture and asymmetry tissue changes, we do not produce additional silicone phantoms, yet reuse the existent ones with or without inflammation changes explained previously. The differential texture parameter was implemented by pinching with small tweezers on the silicone surface at multiple random locations. The differential asymmetry parameter was done by shifting one of the phantom nipples to a different location during MIP acquisition (Fig. 6.5).

6.2 Wavelengths Selection Results

The first step of MIP experiments was to verify the target wavelengths for multispectral image acquisitions. We employed the Retiga EXi hyperspectral camera to image the samples in the NIR window from 650 nm to 1100 nm in the step of 10 nm, to observe the spectra, and to verify our assumptions about the absorption properties of the MIP phantom components.

The imaged samples of PDMS, water, lipids (as porcine lard), deoxy-hemoglobin and oxy-hemoglobin (*ex-vivo* bovine blood sample and *in-vivo* over veins of a human volunteer) shown in Fig. 6.6. The amount of liquid samples (water, lipids, and bovine blood) was 3

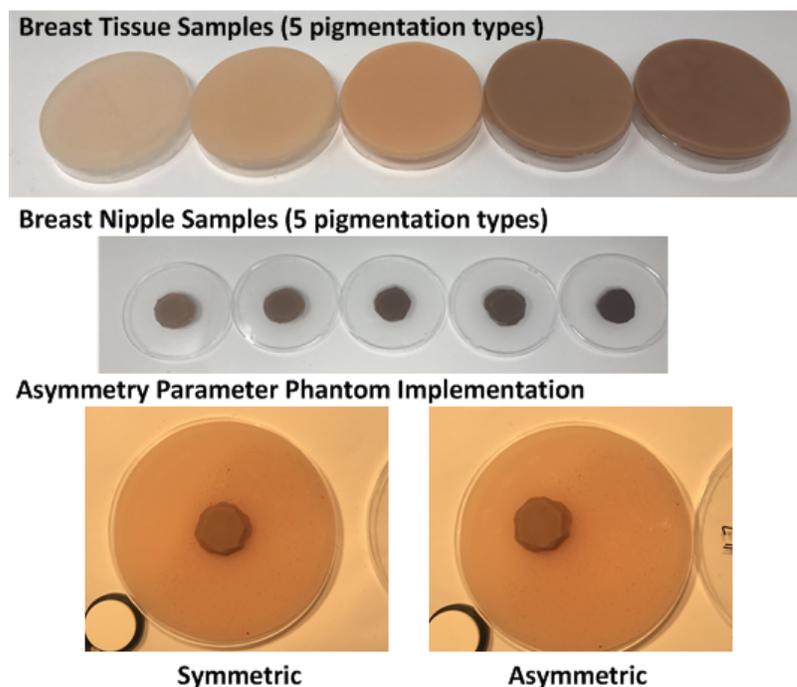


Figure 6.5: MIP asymmetry parameter implementation

ml each. Bovine blood was used in the phantom fabrication. We compare its spectra with the venous human blood to see the differences that we can expect from the MPD method in humans. Each sample was imaged in 3 trials, and the results were averaged.

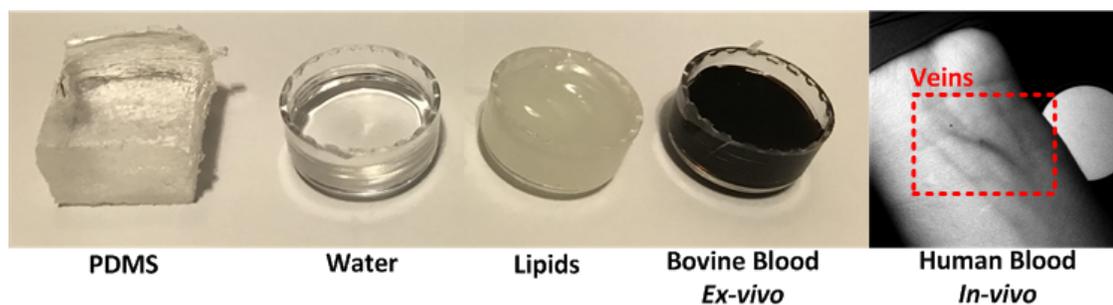


Figure 6.6: Wavelengths selection experiment samples

After collecting the images for each sample in 3 trials, we used ENVI v4.5 (Exelis Visual Information Solutions, Boulder, CO, USA) to select the regions of interest and to analyze the reflectance and absorption/diffusion spectra. Reflectance spectra for all five samples from Fig. 6.6 are presented in Fig. 6.7 (a).

It is not easy to analyze the reflectance spectra and find the corresponding wavelengths

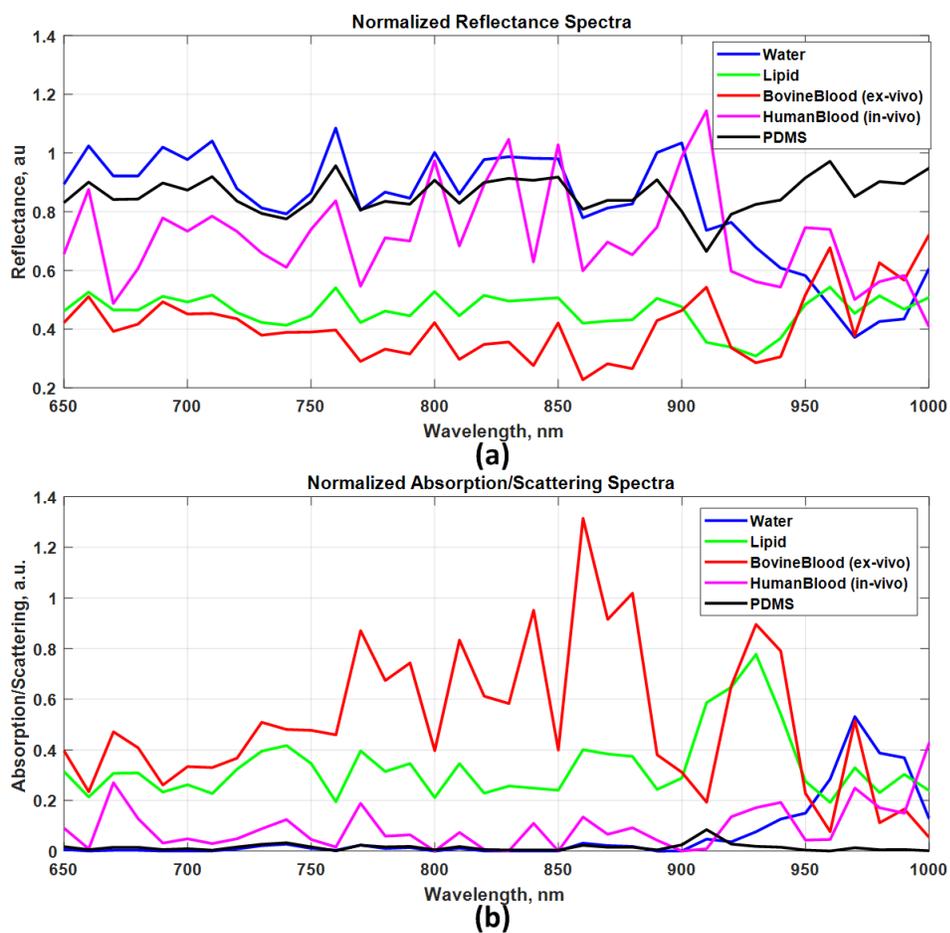


Figure 6.7: Reflectance and absorption/scattering spectra for the experimental samples

for the key chromophores because most of the literature discusses absorption spectra in different tissues and materials, and not the reflectance spectra. Therefore, we used Kubelka-Munk method [101, 102, 103] to translate our reflectance results into absorbance/scattering data for the wavelengths selection task. The absorbance/scattering spectra for the five samples presented in Fig. 6.7 (b).

To better visualize and analyze the absorbance/scattering peaks, we plotted spectra of water, lipids, and PDMS separately from blood samples spectra in Fig. 6.8. We can see from this plot, that PDMS medium contributes very little to the optical absorption in the NIR window, which is the ideal condition for further experiments. We also can confirm that the literature suggested wavelengths to characterize chromophores of water and lipids (discussed in Chapter 5). Absorption of Lipids is the most prominent around 930 nm, whereas the water's absorption spectrum shows a peak around 970 nm.

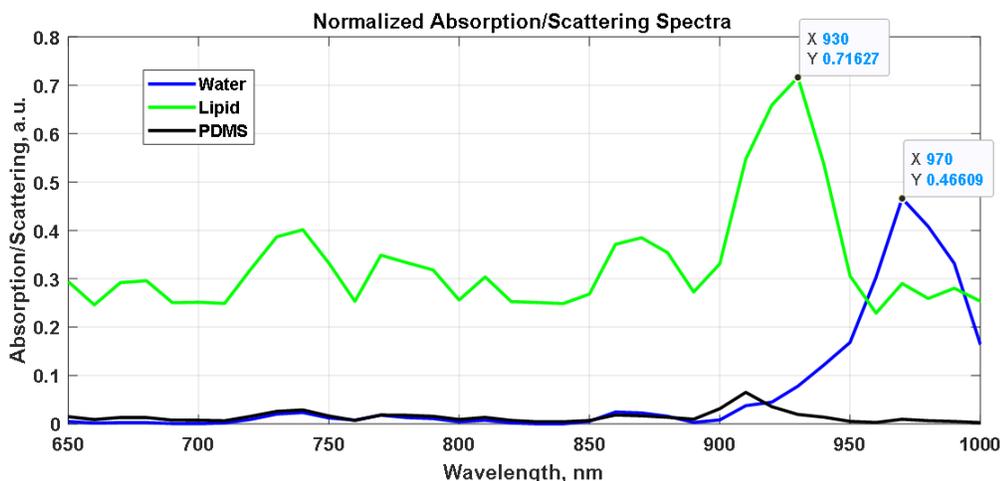


Figure 6.8: Absorbance/scattering spectra for water, lipids, and PDMS samples

To analyze the results for bovine and human blood samples, we plotted the corresponding samples' absorbance/scattering spectra in Fig. 6.9. The human blood values were offset to the level of the bovine blood sample spectra. We added the difference in values between 650 nm values from both samples to the human blood values for all wavelengths. The blood absorption/diffusion spectra display small peaks around 670 nm, 770 nm, and 940 nm. Based on the literature discussed in Chapter 5, first two values correspond to deoxyhemoglobin and the last one to oxy-hemoglobin chromophores. The signal from the

human blood *in-vivo* has a smaller magnitude than the bovine blood sample due to the nature of the experiment. Imaging of veins *in-vivo* solved the issue of being able to image HbO_2 chromophore; however, the sample cannot be called fully homogeneous, which is required for the Kubelka-Munk method. It suggests that the signal was not purely from blood but also from the skin with pigmentation and vein walls. In addition, the amount of blood in veins was less than 3 ml, as each of the liquid test samples had. But the relative peaks are aligned in both cases. The 670 nm peak was not pronounced in the bovine blood sample, so for the dissertation work, we selected the distinctive 770 nm wavelength for the Hb . We confirmed the 94 nm wavelength for HbO_2 characterization from the human blood distinct peak.

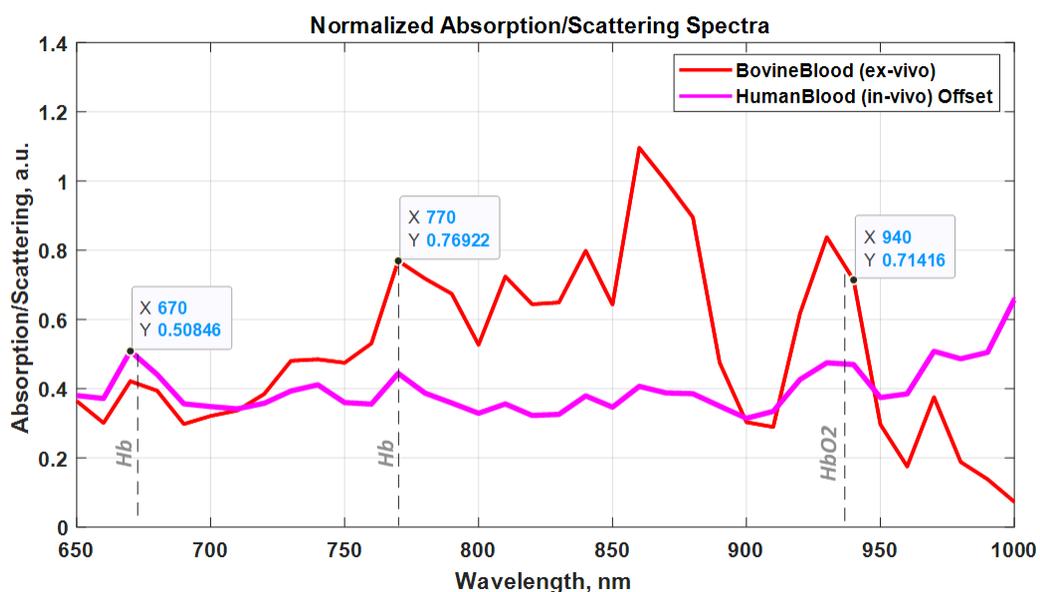


Figure 6.9: Absorbance/scattering spectra for the blood samples

Our final results from the wavelengths selection experiments presented in Fig. 6.10 show the selected wavelengths from our experiments on the tissue chromophores absorbance spectra [80]. Our results matched the literature (Section 5.1).

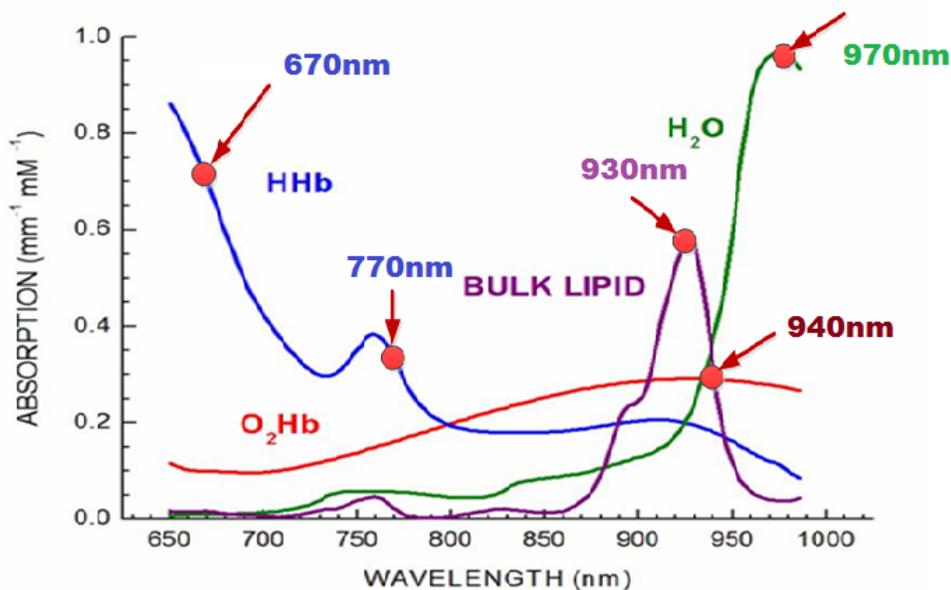


Figure 6.10: Selected wavelengths for MIP

6.3 Differential Multispectral Profile Diagrams Using Phantom Data

We created a large dataset of Differential Multispectral Profile Diagrams by combining MPDs of the individual MIP phantom samples of mimicked normal and diseased breast tissues. Further, we arbitrarily selected representative data from the set for training, validation, and independent test of the classification models. Examples of composed Differential Multispectral Profile Diagrams are shown in Fig. 6.11.

Differential MPD examples 1 - 4 illustrate asymmetry phantoms, where 1 - 2 are symmetric, and 3 - 4 are asymmetric examples with varying inflammation patterns. Examples 5 - 8 show texture changes in Differential MPD samples. Examples 5 - 6 are not pitted and do not have changes in texture, and examples 7 - 8 have implemented texture change as pitted regions. Distinct inflammation levels are shown with examples 9 - 12. Examples 9 - 10 have about 1/3 of the phantom area showcase inflammation change, whereas examples 11 - 12 have the most of the phantom with inflammation.

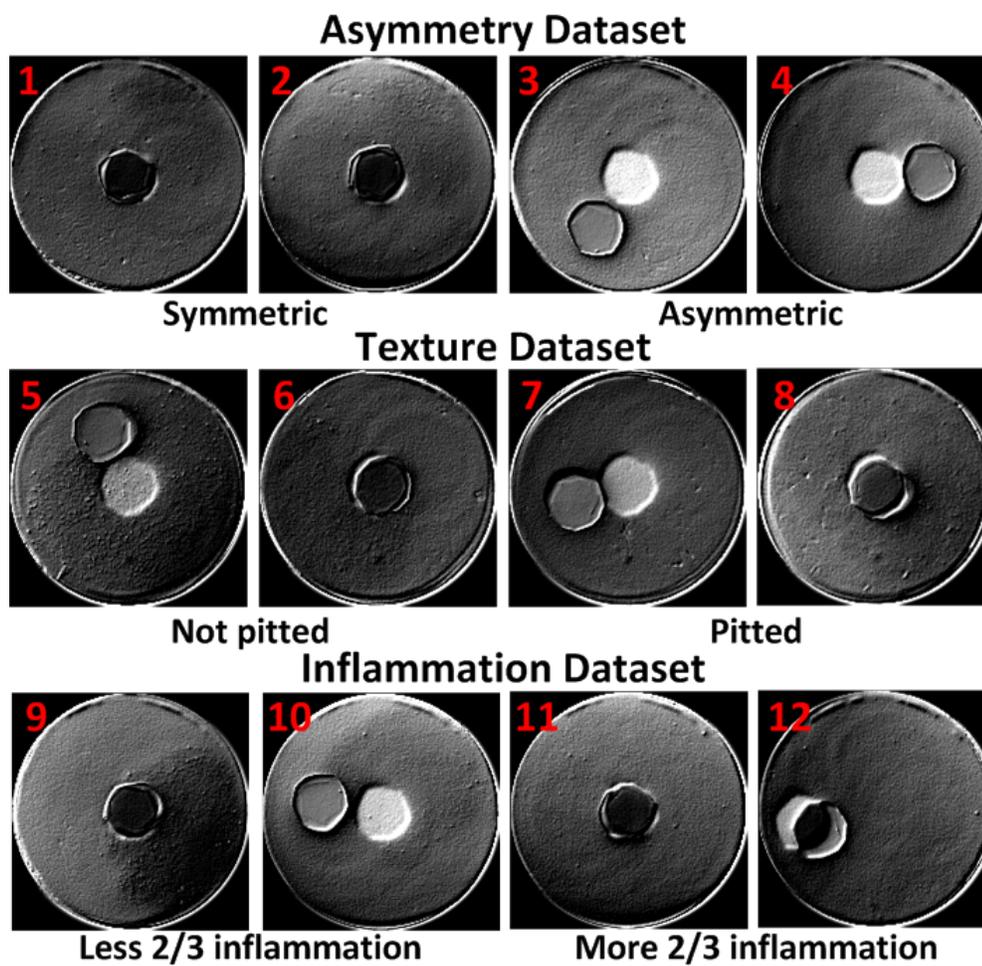
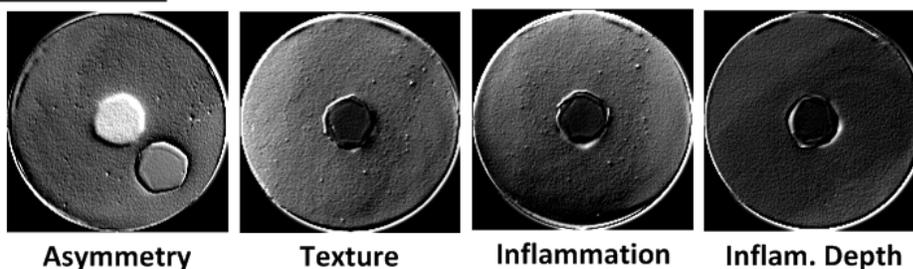


Figure 6.11: Examples of Differential Multispectral Profile Diagrams

6.3.1 Construction of Differential Multispectral Profile Diagrams

Two examples of Differential Multispectral Profile Diagrams with asymmetry, texture, inflammation, and inflammation depth changes are shown in Fig. 6.12. In both phantom samples, asymmetry change shows as two misaligned nipples. After close inspection of the texture change samples, one can note the pitted and the inflammation regions. Both Inflammation samples contain mimicked inflammation regions, yet they cover different area in size ($2/3$ and $1/3$ of the area). The inflammation depth samples illustrate the effect of inflammation depth on the corresponding Differential MPDs. The inflammation region in the deep samples is less pronounced than in superficial samples. Consequently, not all blood amounts and patterns will be distinguishable in deep samples subset during data analysis. The samples of mimicked deep inflammation (5 mm depth) with higher blood concentration (0.5 ml and 0.75 ml) have more chances to be distinguished and classified correctly.

Test Phantom 1



Test Phantom 2

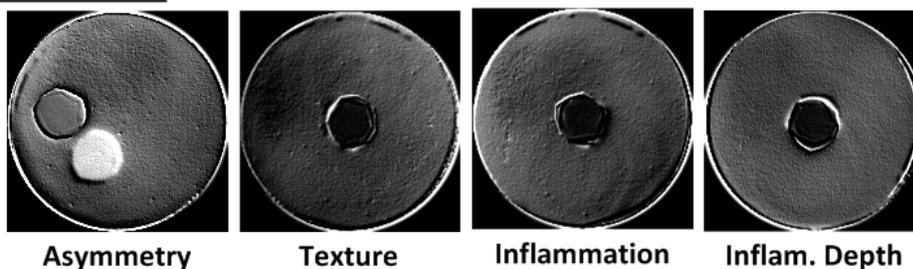


Figure 6.12: Examples of Differential MPDs from two phantom samples

6.3.2 *Classification of Differential Multispectral Profile Diagrams*

We developed three models to classify Differential Multispectral Profile Diagrams with respect to asymmetry, texture, and inflammation. The results from models training and classification are presented in the following sections.

To built these models, we used the same hardware and software as we used for TPD classification models development (Section 4.4.3). To balance data, we used the random upsampling method (Section 4.4.3). We employed Adamax optimizer during training, and Accuracy for models performance evaluation.

The accuracy and loss results for the three models and the examples of Differential MPDs classification are shown on figures later in this chapter. The models were further tested on a set-aside subset of phantom data, and the results are presented in Section 6.3.3.

Asymmetry Classification Model

Our asymmetry classification model was trained on 30720 samples of two balanced via random upsampling classes. The samples were obtained from pairwise combinations of 240 affected and 6 normal MPDs of MIP samples. Validation of the model was completed using 7656 samples. The independent test was done on 1344 set aside samples. The manually selected subsets for the model's training, validation, and test was approximately 75%, 20%, 5%, respectively. The manual method for dataset division allowed to represent each skin pigmentation level and inflammation size equally within the classes.

The asymmetry model description is presented in Section 5.6.1, and the model implementation is given in Attachment F. The results are shown in Fig. 6.13. One can see from the plot that MPDModelAsymmetry was able to quickly learn from the data and approach high classification accuracy for training and validation subsets. The training accuracy for 5 epochs and 100 MPDs in a batch, validation accuracy, and test accuracy are 99%, 100%, and 98%, respectively.

In Fig. 6.14 we show six asymmetry classification examples. The two examples on the top are the phantom samples with asymmetry. The four samples on the bottom are the samples without asymmetry. The bar plots next to each sample show the output class prob-

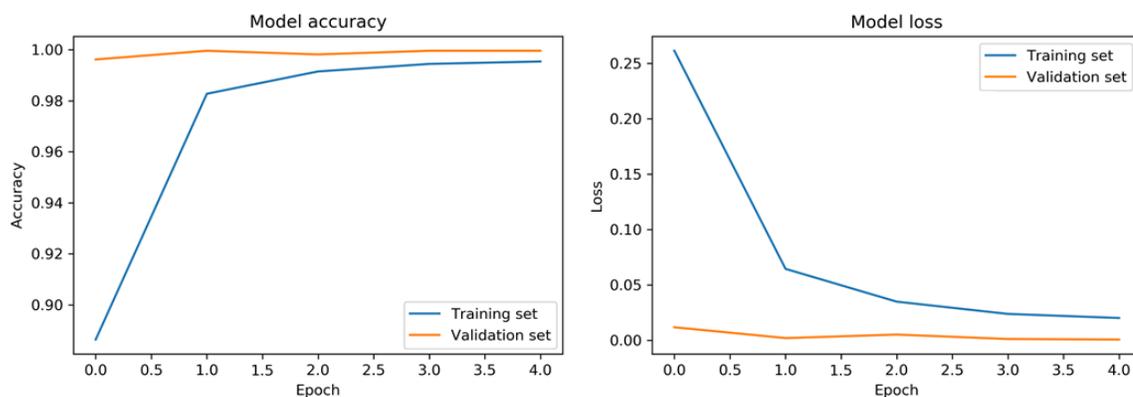


Figure 6.13: Asymmetry model training accuracy and loss

abilities of classification with MPDModelAsymmetry, where class 0 is Symmetric class, and class 1 is Asymmetric class. Text notations under MPD sample images show model classification results with the winning class probability and the true class of the sample in parenthesis. The blue color on the bar plots indicates correct classification.

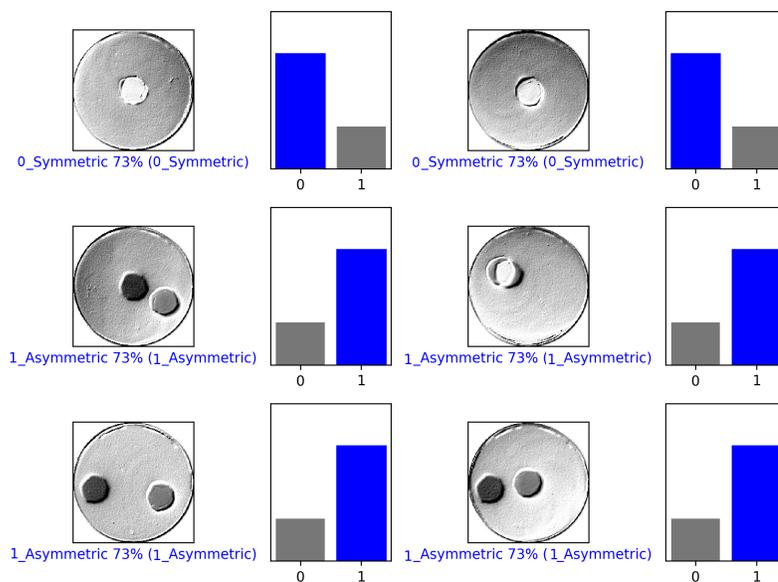


Figure 6.14: Asymmetry classification examples using MPDModelAsymmetry

Texture Classification Model

The texture classification model was trained on 26880 samples of two classes. The classes was initially balanced due to the experimental setup. The validation of the model was completed using 7656 samples. The independent test was done on 3840 samples. The manually selected subsets for the model's training, validation, and test were approximately 70%, 20%, 10%, respectively. The manual method for dataset division allowed to represent each skin pigmentation level and inflammation size equally within the classes.

The texture model description is presented in Section 5.6.2, and the model implementation is given in Attachment F. The model development results are shown in Fig. 6.15. We can see from the plots that the model overfits beginning from the 5th epoch, yet still approaches good validation accuracy. The model would benefit from larger and more representative dataset. The training accuracy for 25 epochs and 100 MPDs in a batch, validation accuracy, and test accuracy are 95%, 81%, and 83%, respectively.

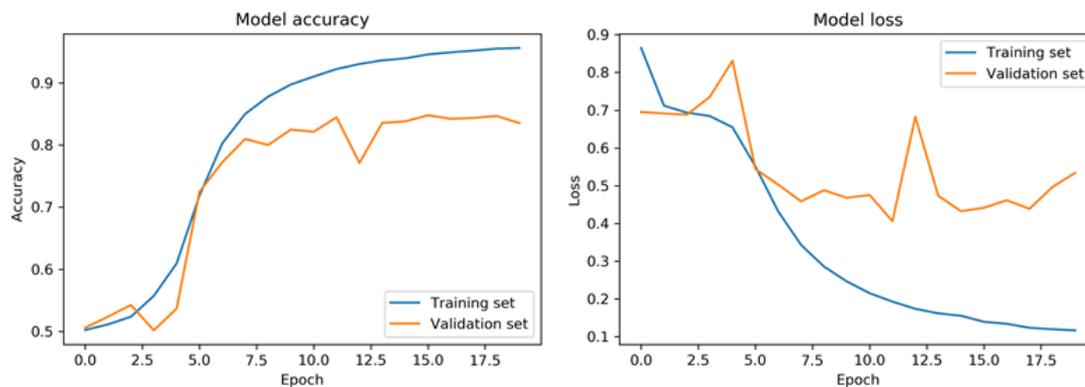


Figure 6.15: Texture model training accuracy and loss

Six examples of Texture model classification results are shown in Fig. 6.16. There are Not Pitted and Pitted samples. The bar plots next to each sample show the output class probabilities of classification with MPDModelTexture, where class 0 is Not Pitted class, and class 1 is Pitted class. The text notation under the MPD sample image shows the model classification results with the winning class probability and the true class of the sample in parenthesis. The blue color on the bar plots indicates correct classification, and the red color shows the incorrectly classified example.

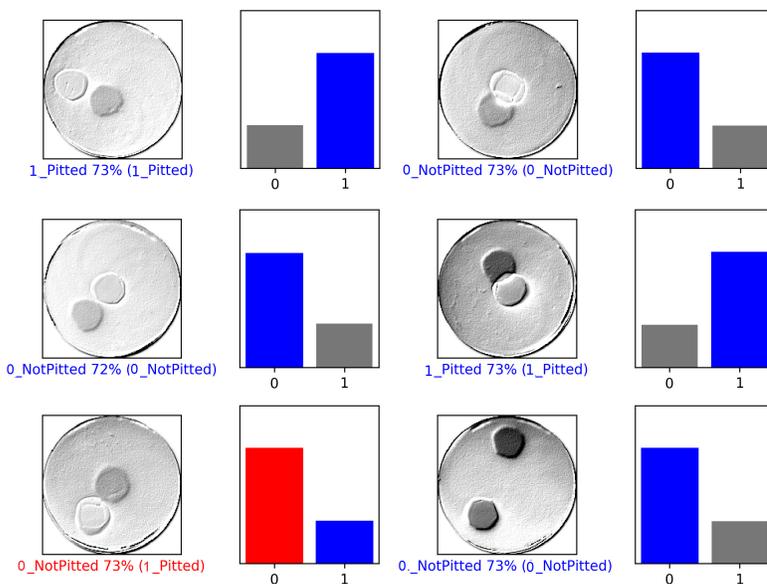


Figure 6.16: Texture classification examples using MPDModelTexture

Inflammation Classification Model

The inflammation classification model was trained on 12251 samples of two balanced via upsampling classes. Validation of the model was completed using 3056 samples. The independent test was done on 372 samples. The manually selected subsets for the model's training, validation, and test were approximately 77.5%, 20%, 2.5%, respectively. The manual method for dataset division allowed to represent each skin pigmentation level and inflammation size equally within the classes.

The inflammation model description is presented in Section 5.6.3, and the model implementation is given in Attachment F. The model development results are shown in Fig. 6.17. Training accuracy for 20 epochs and 50 MPDs in a batch, validation accuracy, and test accuracy are 0.85, 0.61, and 0.77, respectively. The model significantly overfits yet shows the potential for better classification on a larger and more representative dataset. We investigate the additional reasons for the poor model performance during the analysis of the independent test results in the next section.

Six examples of MPDModelInflammation classification results are shown in Fig. 6.18. There are random samples with Small Inflammation (Class 0) and Large Inflammation

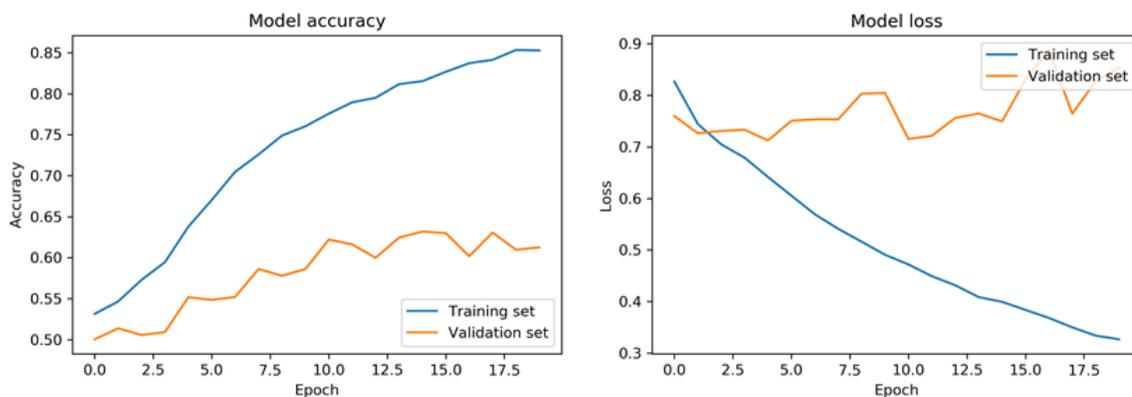


Figure 6.17: Inflammation model training accuracy and loss

(Class 1). One of the samples indicated with red color was miscalssified.

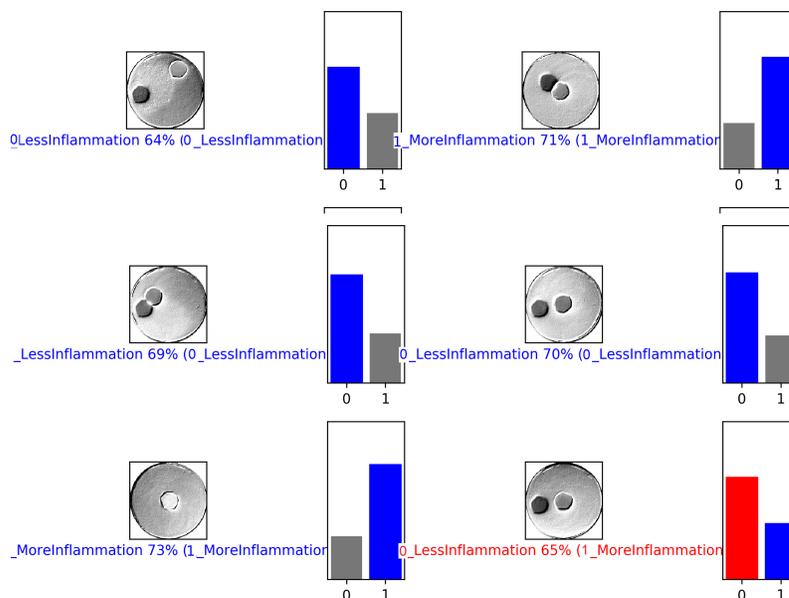


Figure 6.18: Inflammation classification examples using MPDModelInflammation

6.3.3 Test Classification Results

We completed an additional test to further characterize and visualize CNN classification capabilities of differential MPDs. The test samples, which have different inflammation patterns and blood amount, are shown in Fig 6.19. These samples were not included in the classification models' training or validation datasets. The samples' description is presented

in Table 6.1.

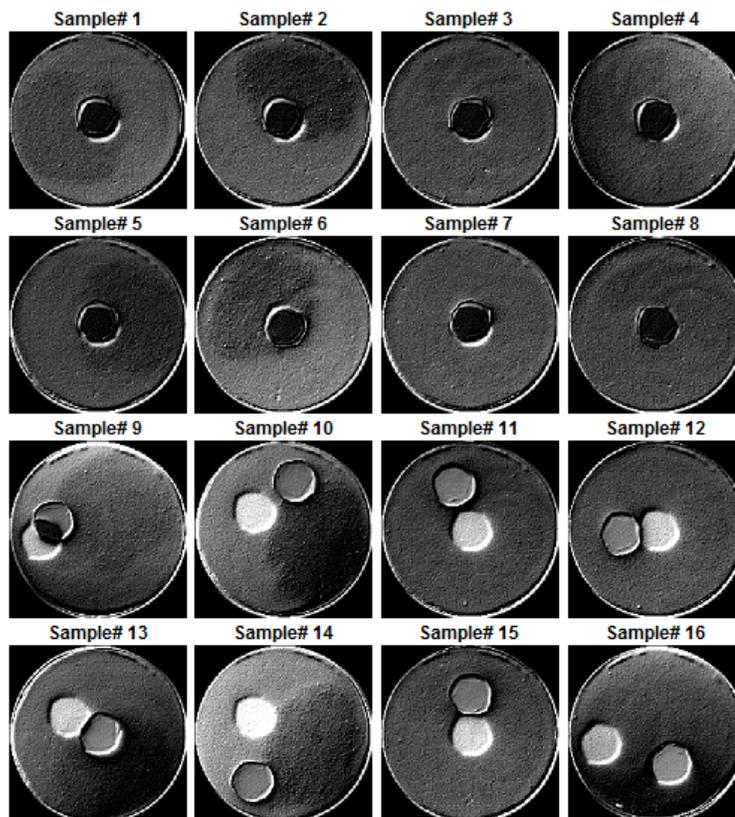


Figure 6.19: Independent classification test samples

The results from the Differential MPD classification of the independent test dataset are presented in Table 6.2. In the table, the rows shaded in gray indicate asymmetric samples, the rows shaded in green point to pitted samples, and the rows shaded in light brown visualize the large size inflammation samples. The misclassified samples are colored in red. The classification accuracy for asymmetry, texture, and inflammation size parameters in the independent test were 100%, 94%, and 88%, respectively.

The inflammation model performs better on the independent test than on the validation set because we excluded samples with 0.15 ml and 0.25 ml blood concentrations. The reason for the exclusion was to keep the more visible inflammation patterns for testing and results evaluation. This exclusion of some samples was done to empirically understand the causes of the inflammation model poor classification performance.

Table 6.1: Independent classification test samples description

Sample	Blood Amount, ml	Inflammation Area, %	Added Features
1	0.50	40	-
2	0.75	40	-
3	0.50	95	-
4	0.75	95	-
5	0.50	40	Pitted
6	0.75	40	Pitted
7	0.50	95	Pitted
8	0.75	95	Pitted
9	0.50	40	Asymmetric
10	0.75	40	Asymmetric
11	0.50	95	Asymmetric
12	0.75	95	Asymmetric
13	0.50	40	Asym Pitted
14	0.75	40	Asym Pitted
15	0.50	95	Asym Pitted
16	0.75	95	Asym Pitted

Table 6.2: CNN classification results for the independent test dataset

Sample	Asymmetry		Texture		Inflammation	
	CNN Class	Prob.	CNN Class	Prob.	CNN Class Prob.	
	Sym.	Asym.	NotPit.	Pitted	Small Infl.	Large Infl.
1	1.00	0.00	1.00	0.00	0.96	0.04
2	1.00	0.00	1.00	0.00	0.99	0.01
3	1.00	0.00	1.00	0.00	0.96	0.04
4	1.00	0.00	1.00	0.00	0.02	0.98
5	1.00	0.00	0.01	0.99	0.99	0.01
6	1.00	0.00	0.00	1.00	1.00	0.00
7	1.00	0.00	0.12	0.88	0.23	0.77
8	1.00	0.00	0.00	1.00	0.36	0.64
9	0.00	1.00	1.00	0.00	0.76	0.24
10	0.00	1.00	1.00	0.00	1.00	0.00
11	0.00	1.00	1.00	0.00	0.01	0.99
12	0.00	1.00	0.99	0.01	0.03	0.97
13	0.00	1.00	0.01	0.99	0.97	0.03
14	0.00	1.00	0.05	0.95	1.00	0.00
15	0.00	1.00	0.14	0.86	0.00	1.00
16	0.00	1.00	0.99	0.01	0.69	0.31
Accuracy		1.00		0.94		0.88

Note: The gray color indicates true Asymmetry class samples, the green color shows the true Pitted class samples, and the pink color highlights the true Large Inflammation class samples. Misclassified samples are indicated with red font color.

6.4 Summary

In this chapter, we developed the test phantom and Multispectral Imaging Probe experiments. We presented the literature overview on the silicone optical property phantom. Further, we demonstrated the results from our target wavelengths selection experiment. We verified the literature specified wavelengths for deo-xyhemoglobin, oxy-hemoglobin, water, and lipids chromophores as 770 nm, 940 nm, 970 nm, and 930 nm, respectively. Next, we showed the construction of Differential Multispectral Profile Diagrams and presented examples of how asymmetry, texture, and inflammation features are implemented using multispectral phantoms and Differential MPDs. Then, we described training and validation details of three CNN classification models for asymmetry, texture, and inflammation size evaluation. The probabilities from the classification via these tree models will be used for Spectral Index calculation in the next chapter. Finally, we completed an independent test to classify asymmetry, texture, and inflammation features using a representative subset with higher blood concentration levels. The classification accuracy results on the subset were 100%, 94%, and 88% for asymmetry, texture, and inflammation size parameters, respectively.

CHAPTER 7

MULTIMODAL BREAST CANCER RISK ASSESSMENT

In this chapter, we present a method to evaluate the risk of cancer for the imaged breast tissue and tumor region based on the results from two imaging modalities (Tactile Imaging Probe and the Multispectral Imaging Probe), and the personalized risk probability from the Breast Cancer Risk Assessment Tool (BCRAT) developed by National Cancer Institute [4]. We describe the method to combine Tactile Index and Spectral Index with the BCRAT calculator output to find Multimodal Index for the breast cancer risk assessment.

7.1 Review of Cancer Risk Calculations

Machine learning shows great potential in supporting doctors during clinical diagnostic procedures [104]. Machine learning methods are more robust, capable, and better individualized, so they outperform the well established, manually developed empirical malignancy risk scoring models. It gives additional information for the individualized cancer risk assessment [105].

Accurate predictions on who will develop breast cancer are not available; however, there are several well-known risk assessment methods [104]. These methods help patients and doctors to develop a custom breast diagnostic plan, or even schedule a preventive surgery.

The Gail model [106] is the most popular tool for the five and ten-year cancer development risk assessment based on a short questionnaire that women complete with their doctors. The Claus model [107] gives age-specific risk probability of breast cancer for women with a family history of breast cancer. The Tyrer-Cuzick model [108] tool estimates a 10-year breast cancer risk based on a woman's answers to a series of questions.

Researchers recently utilized Convolutional Neural Network to extract features from medical images to calculate a cancer risk score and aid radiologists in diagnostic decisions [109, 110]. During CNN classification, the class assignment is based on the probabilities

calculated for each of the classes. The empirical probabilities signify how confident the class decision is, and it can be used for the malignancy score calculation in cancer applications.

We propose to calculate cancer risk indices for both TIP and MIP modalities of our bimodal imaging system separately and then combine these together with the BCRAT score from NCI to provide an individualized Multimodal Index for patient breast cancer risk assessment.

7.2 Tactile Index

The cancer probability from Tactile Imaging Probe is called Tactile Index. To calculate Tactile Index, $Index_T$, we take the weighted average of the deep class probability from the depth classification model, medium and large size class probabilities from the size classification model, and stiff class probability from the stiffness classification model, as follows.

$$Index_T = \alpha_1 P_{11} + \alpha_2 P_{12} + \alpha_3 P_{13}, \quad (7.1)$$

where P_{11} corresponds to the deep tumor class probability from CNN TPDMoDelDepth classification, P_{12} is the medium and large tumor class probability from CNN TPDMoDelSize classification, and P_{13} corresponds to the stiff tumor class probability from CNN TPDMoDelStiffness classification. α_1 , α_2 and α_3 are the corresponding weights for depth, size and stiffness probabilities, respectively.

7.3 Spectral Index

The cancer probability from Multispectral Imaging Probe is called Spectral Index. We propose to calculate the Spectral Index, $Index_S$, using three multispectral imaging parameters associated with breast cancer and their probabilities obtained from the asymmetry classification model, texture classification model, and inflammation classification model. Eq. (7.2) shows $Index_S$ calculation.

$$Index_S = \beta_1 P_{21} + \beta_2 P_{22} + \beta_3 P_{23}, \quad (7.2)$$

where P_{21} corresponds to Asymmetric class probability from MPDModelAsymmetry model classification, P_{22} is the Pitted class probability from MPDModelTexture model classification, P_{23} corresponds to the probability of Large Inflammation class probability (more than 2/3 of the area) from CNN MPDModelInflammation model classification. β_1 , β_2 , and β_3 are the corresponding weights for the differential asymmetry, texture, and large inflammation probabilities, respectively.

7.4 NCI Breast Cancer Risk Assessment Tool

The National Cancer Institute made the Breast Cancer Risk Assessment Tool (BCRAT) publicly available to calculate individual risk of developing breast cancer [4]. The probability is largely based on the well-known Gail model [106], which was further developed by Costantino in 1999 [111]. Doctors and medical residents are currently encouraged to use the tool in their practice [104].

BCRAT provides an online calculator, where women can enter information by answering questions about their health history (Fig. 7.1), and receive the predictions for developing breast cancer in five and in ten years in the future (Fig. 7.2). The women with higher than average risk are suggested to talk to their primary care physicians on developing an individualized screening plan.

We incorporate the NCI BCRAT calculation into the Multimodal Breast Cancer Risk Assessment as the individualized health index named BCRAT Index, $Index_{BCRAT}$. Eq. (7.3) shows the index calculation.

$$Index_{BCRAT} = P_{31}, \quad (7.3)$$

where P_{31} corresponds to the 5-year probability of developing breast cancer from BCRAT.

7.5 Multimodal Index

To calculate the Multimodal Index for breast cancer risk assessment, we propose to compute the weighted sum of $Index_T$, $Index_S$, and $Index_{BCRAT}$ as follows.

$$Multimodal\ Index = w_1 Index_T + w_2 Index_S + w_3 Index_{BCRAT}, \quad (7.4)$$

1. Does the woman have a medical history of any breast cancer or of ductal carcinoma in situ (DCIS) or lobular carcinoma in situ (LCIS) or has she received previous radiation therapy to the chest for treatment of Hodgkin lymphoma?
2. Does the woman have a mutation in either the *BRCA1* or *BRCA2* gene, or a diagnosis of a genetic syndrome that may be associated with elevated risk of breast cancer?
3. What is the patient's age?
4. What is the patient's race/ethnicity?
 - a. What is the sub race/ethnicity or place of birth?
5. Has the woman ever had a breast biopsy?
 - a. How many breast biopsies (positive or negative) has the woman had?
 - b. Has the woman ever had a breast biopsy with atypical hyperplasia?
6. What was the woman's age at the time of her first menstrual period?
7. What was the woman's age when she gave birth to her first child?
8. How many of the woman's first-degree relatives (mother, sisters, daughters) have had breast cancer?

Figure 7.1: The BCRAT calculator questions [4]

5-Year Risk of Developing Breast Cancer	
Patient Risk	Average Risk
0.4%	0.3%
Lifetime Risk of Developing Breast Cancer	
Patient Risk	Average Risk
10.4%	12.6%

Figure 7.2: Example of the NCI BCRAT calculator results [4]

where w_1 , w_2 , and w_3 are the weights of the corresponding indices.

Fig. 7.3 shows the block diagram for the Multimodal Index calculation for breast cancer risk assessment. The index will range from 0 to 1, where values closer to zero will correspond to the probability of the tissue with a tumor to be benign, and the values closer to 1 will signify the higher probability of malignancy.

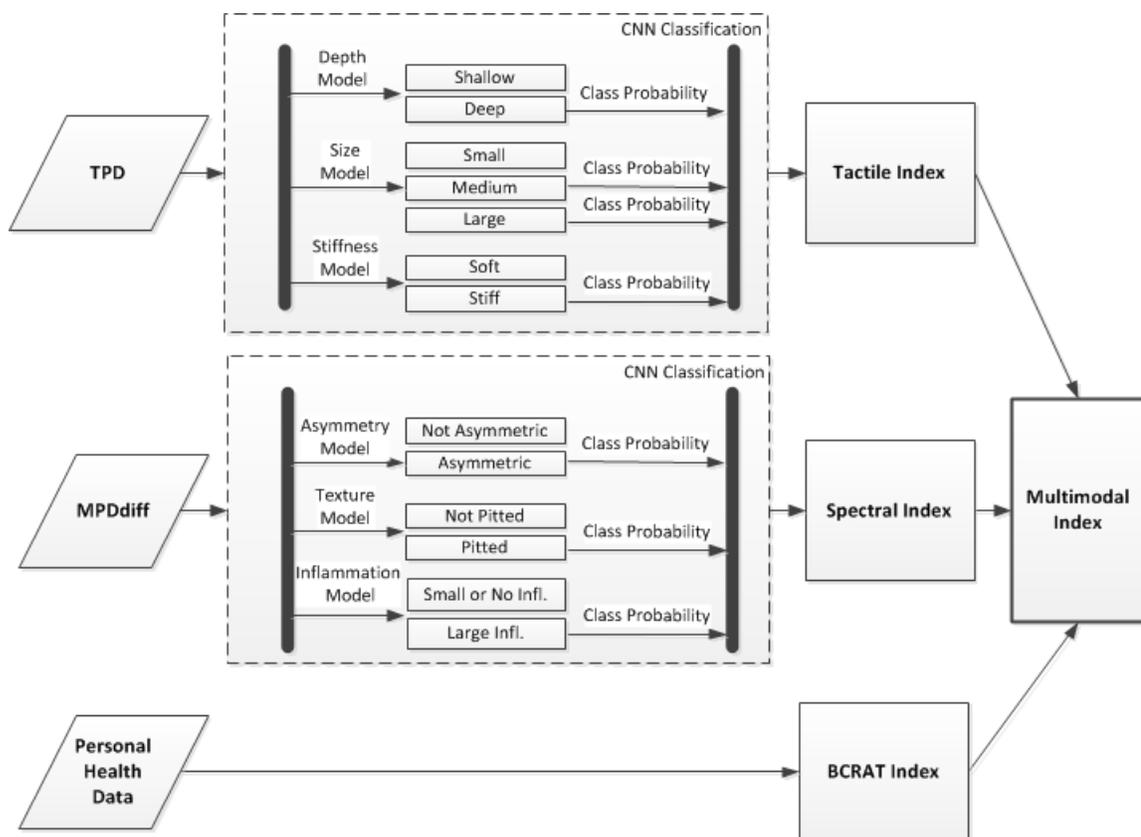


Figure 7.3: Multimodal Index development

7.6 Multimodal Index Results

Here we provide the description of the bimodal imaging phantom for the validation experiments. Further, Multimodal Index calculation results are presented and analyzed.

7.6.1 Bimodal Imaging Phantom

We designed a bimodal imaging phantom to validate the method for Multimodal Index calculation. The requirement for the phantom was the ability to emulate breast tissue's mechanical and optical properties. We accomplished this task by reusing multiple interchangeable layers from the TIP phantom. We combine TIP phantom with MIP phantom by placing MIP phantom layers one at a time on the top of the TIP mechanical property phantom with an inclusion (Fig. 7.4).

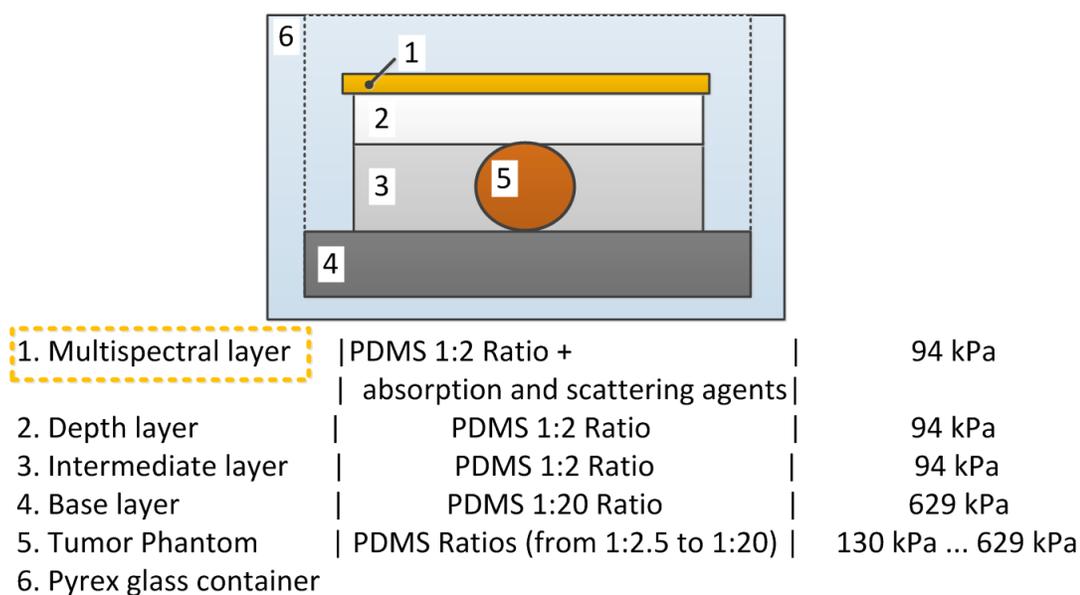


Figure 7.4: Bimodal imaging phantom with optical and mechanical property

Fig. 7.5 shows the implementation of the bimodal imaging phantom with tactile and multispectral imaging parameters. The left side of the figure shows an experimental setup for multispectral imaging of the bimodal phantom. On the right, three views show the tactile imaging phantom layers. We can use both phantom modalities at once for each imaging acquisition. During tactile imaging, a 5 mm thick multispectral layer can be used by itself as a depth layer or can be combined with other depth layers to achieve the desired imaging depth.

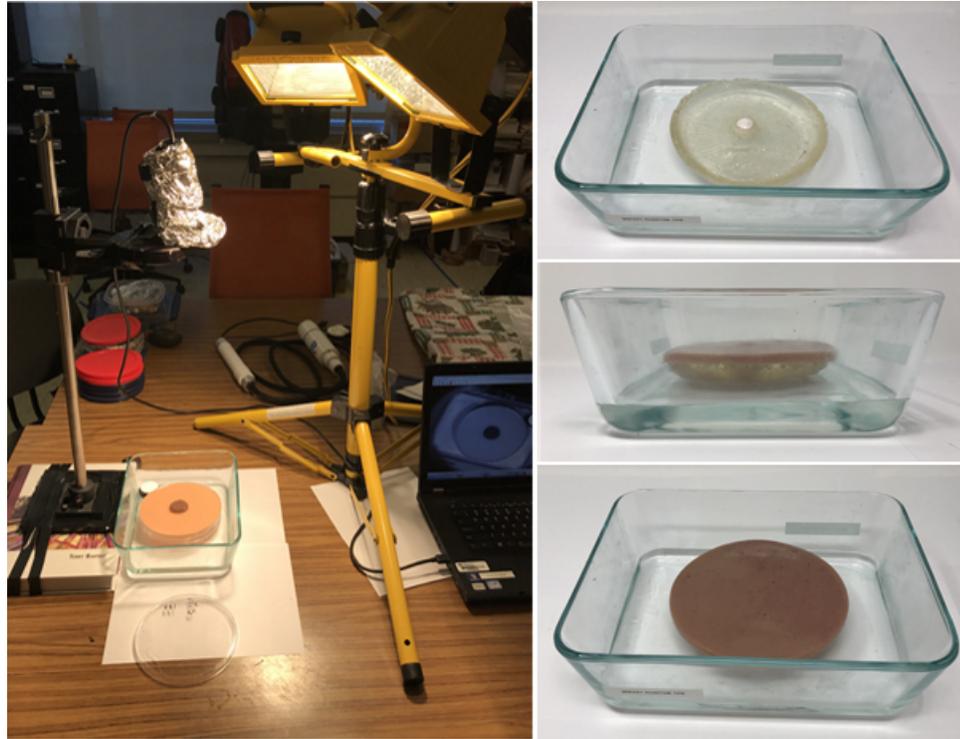


Figure 7.5: Bimodal imaging phantom

7.6.2 Multimodal Index Calculation Dataset

The dataset for the Multimodal Index calculation was developed by combining results from the TIP phantom experiment subset (Section 4.4.3), and the MIP phantom experiment test subset (Section 6.3.2). Larger and stiffer phantom inclusions correspond to larger and stiffer breast tumors. Asymmetry changes, larger area of inflammation, and MIP samples with pitted texture will mimic asymmetry, inflammation, and texture breast tissue changes.

For representative Tactile Index calculations, we used one trial classification results from the TPD CNN models given in Table 4.9. For representative Spectral Index calculations, we used classification results for samples 2, 4, 7, 10, 13, and 15 from the test of MPD CNN models given in Table 6.2. The results are presented and discussed in the next section.

7.6.3 Multimodal Index Calculation Results

In this section, we show the results for Tactile Index, Spectral Index, and Multimodal Index calculations for breast cancer risk assessment.

Tactile Index calculation for a small set of samples from Table 4.9 test results are shown in Table 7.1. For the Tactile Index calculation, we used Equation (7.1), as $0.1 * prob(Deep) + 0.2 * (prob(Med) + prob(Large)) + 0.7 * prob(Stiff)$. We assigned the weights based on our empirical knowledge from a previous publication [34], where stiffness had a weight of 0.7.

From the results in Table 7.1, we note that Tactile Index is lower for softer and smaller inclusions. It increases when the inclusion is stiffer and larger. The increased index value correlates with the assumptions about the malignant tumors being stiffer and larger.

Spectral Index calculation results for a small subset from Table 6.2 set are presented in Table 7.2. For Spectral Index calculation, we used Equation (7.2), as $0.2 * prob(Asymmetric) + 0.2 * prob(Pitted) + 0.6 * prob(Large Inflammation)$. The weights are empirical and require further evaluation in future work. Spectral Index is lower for tissue samples with less inflammation and no texture or asymmetry changes. It is the highest when tissue has a large area of inflammation with asymmetry and pitted texture. The results correlate with the malignancy assumptions for the superficial breast tissues.

Finally, we calculated Multimodal Index, shown in Table 7.3. For the Multimodal Index calculation, we used Equation (7.4), as $0.7 * Spectral Index + 0.3 * Tactile Index$. The weights were assigned empirically. We omitted the NCI BCRAT Index in this experiment because our test data developed purely from phantom data and did not have information about human patients. However, the BCRAT Index can be incorporated into the Multimodal Index calculation for human patients breast cancer risk assessment as Tactile Index and Spectral Index.

The light red-colored cells in Table 7.3 indicate mimicked malignant cases. As expected, the results for the Multimodal Index in these cases are the highest. The light gray color in the last column of the table points to the cases with increased multimodal malignancy risk. The results for Multimodal Index in these cases are close to the middle of the

Table 7.1: Tactile Index calculation results

Sample #	Depth		Stiffness		Size			Tactile Index
	CNN Model Class (Prob.)		CNN Model Class (Prob.)		CNN Model Class (Prob.)			
	Shallow	Deep	Soft	Stiff	Small	Med	Large	
1	0.69	0.31	0.37	0.63	0.24	0.54	0.22	0.62
2	0.73	0.27	0.73	0.27	0.51	0.26	0.22	0.31
3	0.69	0.31	0.27	0.73	0.21	0.57	0.21	0.70
4	0.28	0.72	0.52	0.48	0.37	0.40	0.23	0.54
5	0.73	0.27	0.27	0.73	0.21	0.21	0.58	0.70
6	0.64	0.36	0.69	0.31	0.22	0.55	0.23	0.41
7	0.28	0.72	0.54	0.46	0.49	0.28	0.23	0.50
8	0.29	0.71	0.39	0.61	0.25	0.53	0.22	0.65
9	0.27	0.73	0.29	0.71	0.22	0.57	0.21	0.72
10	0.27	0.73	0.73	0.27	0.21	0.58	0.21	0.42
11	0.40	0.60	0.27	0.73	0.21	0.21	0.58	0.73
12	0.28	0.72	0.56	0.44	0.55	0.23	0.22	0.47

Note: The green color indicates mimicked deep tumors, the gray color highlights the mimicked stiff tumors, the light blue color shows mimicked medium size tumors, and the dark blue color indicates large mimicked tumors. The red font color identifies the incorrectly classified tumors using the CNN models.

Table 7.2: Spectral Index calculation results

Sample #	Asymmetry CNN Model Class (Prob.)		Texture CNN Model Class (Prob.)		Inflammation CNN Model Class (Prob.)		Spectral Index
	Sym.	Asym.	Not Pitted	Pitted	Small Infl.	Large Infl.	
1	1.00	0.00	1.00	0.00	0.99	0.01	0.01
2	1.00	0.00	1.00	0.00	0.02	0.98	0.59
3	1.00	0.00	0.12	0.88	0.23	0.77	0.63
4	0.00	1.00	1.00	0.00	1.00	0.00	0.20
5	0.00	1.00	0.01	0.99	0.97	0.03	0.41
6	0.00	1.00	0.14	0.86	0.00	1.00	0.97
7	1.00	0.00	1.00	0.00	0.99	0.01	0.01
8	1.00	0.00	1.00	0.00	0.02	0.98	0.59
9	1.00	0.00	0.12	0.88	0.23	0.77	0.63
10	0.00	1.00	1.00	0.00	1.00	0.00	0.20
11	0.00	1.00	0.01	0.99	0.97	0.03	0.41
12	0.00	1.00	0.14	0.86	0.00	1.00	0.97

Note: The gray color indicates samples with asymmetry, the green color highlights the pitted samples, and the pink color indicates samples with large mimicked inflammation. All samples are correctly classified using the CNN models.

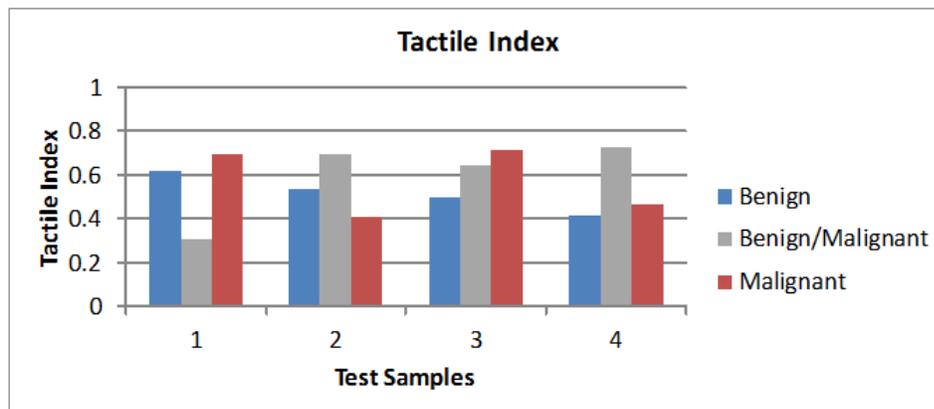
Table 7.3: Multimodal Index calculation results

TPD Sample	MPD Sample	Tactile Index	Spectral Index	Multimodal Index
1 shallow	1	0.62	0.01	0.19
2 shallow	2	0.31	0.59	0.51
3 shallow	3	0.70	0.63	0.65
4 shallow	4	0.54	0.20	0.30
5 shallow	5	0.70	0.41	0.50
6 shallow	6	0.41	0.97	0.80
1 deep	1	0.50	0.01	0.15
2 deep	2	0.65	0.59	0.61
3 deep	3	0.72	0.63	0.66
4 deep	4	0.42	0.20	0.27
5 deep	5	0.73	0.41	0.51
6 deep	6	0.47	0.97	0.82

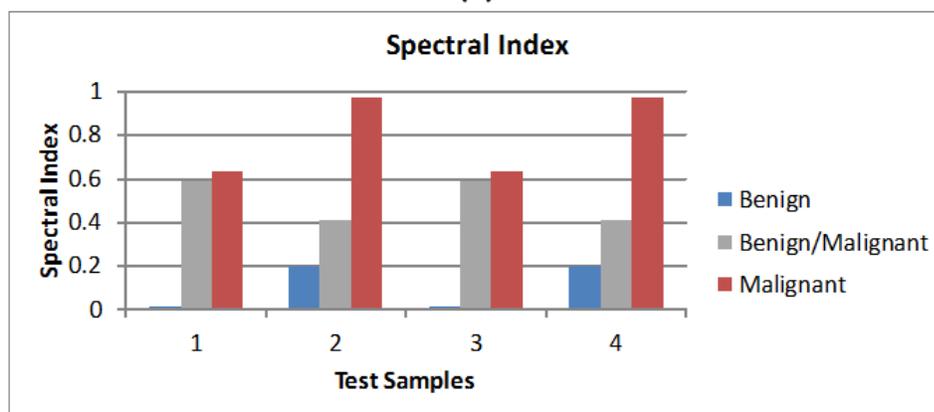
Note: The blue-colored cells indicate mimicked benign cases. The gray-colored cells show the cases with increased chances of malignancy. The light red-colored cells point to the mimicked malignant cases.

scale from 0 to 1. The light blue-colored cells indicate mimicked benign cases. The results for the Multimodal Index in these cases are the lowest.

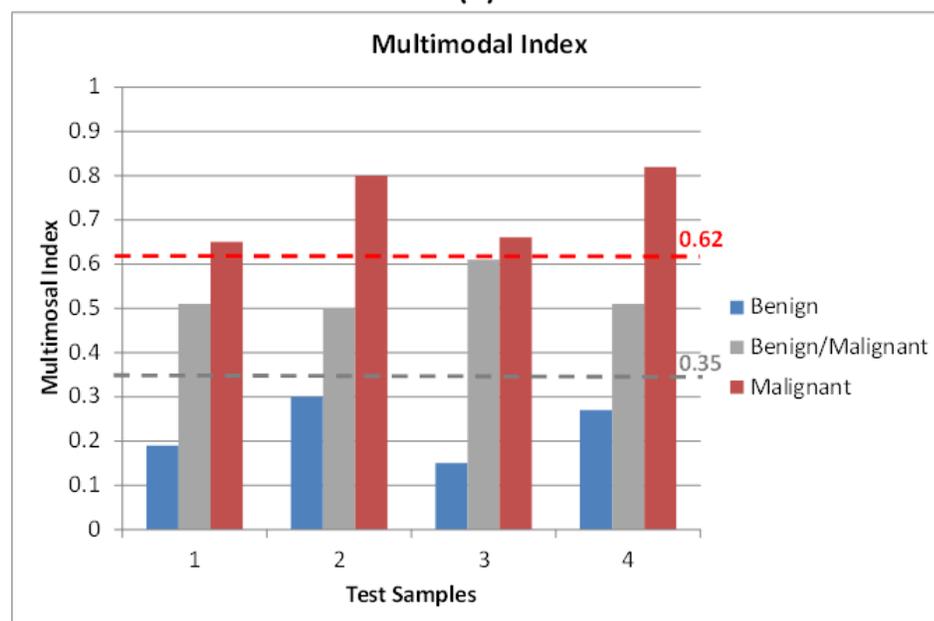
The results for the Multimodal Index calculation are visualized in Fig. 7.6. Each of the mimicked cases has four test samples. Blue bars correspond to mimicked benign samples, red bars show mimicked malignant samples, and gray bars point to test samples with elevated probability of being malignant (benign/malignant cases). We can see from the figure that it is difficult to separate mimicked malignant, malignant/benign, and benign cases using only the Tactile Index (Fig. 7.6(a)) or only the Spectral Index (Fig. 7.6(b)). In Fig. 7.6(a), we note that we cannot draw linear thresholds to separate the cases using only Tactile Index data. In Fig. 7.6(b), we can see that it is possible to have two linear thresholds to separate the cases, yet some of Malignant and Benign/Malignant test samples' results nearly overlap. However, Fig.7.6(c) shows that we can better differentiate these three mimicked cases by calculating Multimodal Index, and separate them with two linear thresholds.



(a)



(b)



(c)

Figure 7.6: Multimodal Index calculation results

The test samples with calculated Multimodal Index below threshold 0.35 correspond to the mimicked benign cases. The test samples with Multimodal Index greater than 0.62 correspond to the mimicked malignant cases. And the samples with calculated Multimodal Index between 0.35 and 0.62 are the mimicked cases with increased malignancy risk.

7.7 Summary

In this chapter, we derived the method to calculate the Multimodal Index for breast cancer risk assessment using the probabilities of the key features from Tactile Imaging Probe and Multispectral Imaging Probe, as well as the probability of malignancy based on the patient's health records from the NCI's Breast Cancer Risk Assessment Tool. We used deep learning capabilities to extract the probabilities associated with the features from imaging data of TIP and MIP modalities and to use it for the highly individualized Multimodal Index for breast cancer risk assessment.

We calculated Multimodal Index for a small yet representative phantom dataset. In this experiment, NCI BCRAT Index was equal to zero because we only used phantom data and no human data. Our Multimodal Index results had consistently higher values for the mimicked malignant cases. The suggested threshold for the malignant samples in this dataset was 0.62. The calculated Multimodal Index values for mimicked benign cases were below 0.35 threshold. We also hypothesize that the incorporation of BCRAT Index in Multimodal Index calculation for human patient cases will help distinguish malignant tumors more accurately. Furthermore, other imaging modalities can be introduced to Multimodal Index calculation in the future. The method can be useful for healthcare providers to assess a patient's breast cancer risk from multiple modalities.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Here we will provide our conclusions for the Ph.D. dissertation and will outline the plan for the future work.

8.1 Conclusions

In this dissertation, we developed the bimodal imaging system and algorithms to capture the tactile and multispectral properties of breast tissues and provide the cancer risk assessment.

- We advanced Tactile Imaging Probe to evaluate tactile characteristics of embedded tumors such as depth, size, and stiffness. Experiments on custom breast tissue mimicking silicone phantoms and *in-vivo* human trials were completed to calculate size and stiffness for embedded tumors. The method required preliminary estimation of tumors' depth and size by an operator to guide the calculations. Further, we proposed the Tactile Profile Diagrams to efficiently capture the tactile property of tumors and encode it in the form of pattern images. Because Convolutional Neural Network gained wide popularity as an excellent method for pattern classification, we applied it to classify Tactile Profile Diagrams. In our experiments, we showed good accuracy in classifying tumors based on their depth, size, and stiffness using Tactile Profile Diagrams and Convolutional Neural Network. The CNN-estimated depth, size, and stiffness of tumors can improve size calculation error and eliminate the need for preliminary estimation of tumors' depth and size by the operators.
- We developed Multispectral Imaging Probe to evaluate superficial optical properties of breast tissues aiming to characterize manifestations of inflammatory breast cancer. The probe utilizes the same hardware as TIP, yet without the tactile sensing element on the front of the camera lens. Instead, MIP uses a mechanical rotary attachment with four bandpass filters. The bands were selected during wavelengths

selection experiments and targeted hemoglobin, oxy-hemoglobin, water, and lipid chromophores. We proposed a method to create Multispectral Profile Diagrams to capture the superficial optical properties as pattern images. MPDs were also classified with CNN models to characterize breast tissue inflammation, texture, and asymmetry parameters. The results showed the possibility of using these methods for superficial breast tissues, yet need to be developed for the higher accuracy on a large and more representative dataset.

- Lastly, we developed a method to calculate the individualized Multimodal Index for breast cancer risk assessment in patients. The probabilities of features obtained from the tactile and multispectral imaging modalities data combined with the individual breast cancer risk probability from National Cancer Institute, give the opportunity to assess the breast tissue and tumor from different points of view and lead to a better individualized patient's breast cancer risk assessment.

8.2 Future Work

In this dissertation, we demonstrated how to acquire bimodal imaging data from breast mimicking phantoms and human patients using tactile and multispectral probes. We also presented methods for extracting important parameters from the bimodal data using interpolation methods and Convolutional Neural Network classifiers. We used the weighted average method with empirically suggested weights to consolidate the parameters and form Tactile, Spectral, and BCRAT indices for the validation of the breast cancer risk assessment method. We envision three possible directions for the future work, as follows.

- Development of Multimodal Index for breast cancer risk assessment using classification techniques for parameters consolidation and malignancy prediction.
- Research on the proper distribution of weights during Multimodal Index calculation to obtain useful and clinically meaningful results for patients [6].
- An extensive literature review on size and stiffness of malignant and benign tumors, asymmetry, texture, and inflammation of malignant and benign breast tissues, com-

bined with a statistically significant amount of patients' clinical records on these mentioned parameters to simulate a larger and more representative realistic dataset for malignancy prediction accuracy improvement (Figure 8.1).

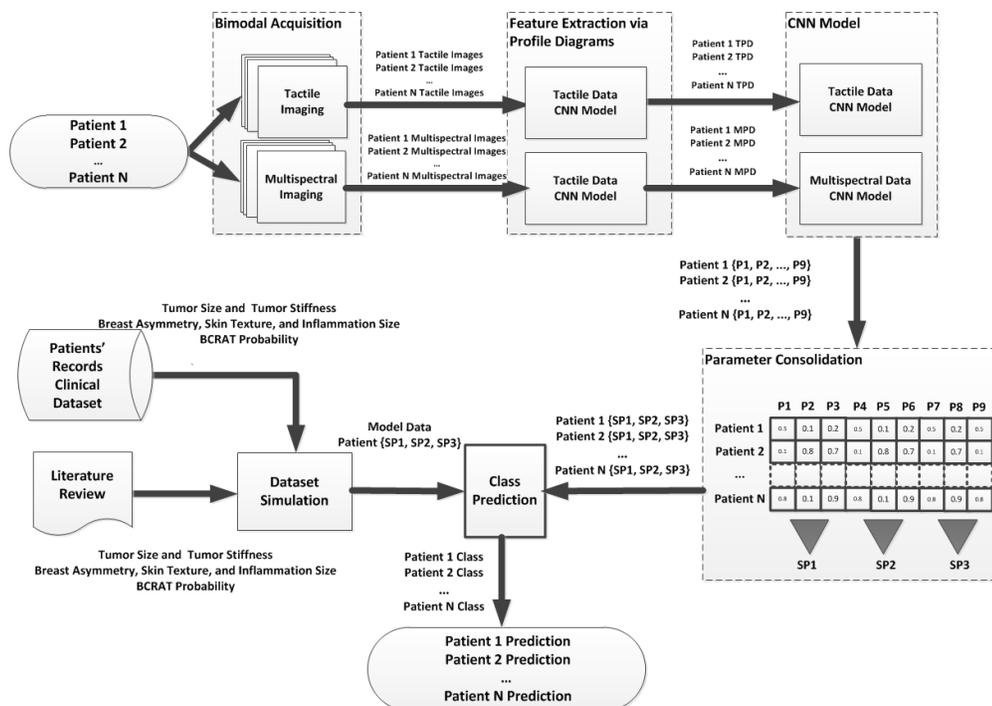


Figure 8.1: Future developments in Multimodal Index methodology

REFERENCES

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Mass: MIT Press, 2017.
- [2] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *Journal of biomedical optics*, vol. 19, no. 1, p. 10901, 2014, Retrieved on 06/07/2021. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/24441941> { % } 5Cnhttp://biomedicaloptics.spiedigitallibrary.org/article.aspx?articleid=1816617
- [3] A. Godavarty, S. Rodriguez, Y.-J. Jung, and S. Gonzalez, “Optical imaging for breast cancer prescreening.” *Breast cancer (Dove Medical Press)*, vol. 7, pp. 193–209, 2015, Retrieved on 03/05/2020. [Online]. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4516032> { & } tool=pmcentrez { & } rendertype=abstract
- [4] National Cancer Institute, “The Breast Cancer Risk Assessment Tool,” 2017, Retrieved on 04/15/2020. [Online]. Available: <https://bcrisktool.cancer.gov/index.html>
- [5] J. Hellawell, “Breast Diseases,” pp. 305–307, 2008.
- [6] A. Aydiner, A. Igci, and A. Soran, *Breast Disease. Diagnosis and Pathology*. Springer, 2017, vol. 1.
- [7] J. R. Harris, J. Pine, Jonathan W., J. Goolsby, E. Moyer, B. Dougherty, and T. Mallon, *Diseases of the Breast*, 5th ed. Philadelphia, Pennsylvania: Wolters Kluwer, 2014, Retrieved on 02/03/2020. [Online]. Available: <https://ebookcentral.proquest.com/lib/templeuniv-ebooks/reader.action?docID=2031620>
- [8] K. Chong, “Education: Clinical review - Benign breast disease,” 2015, Retrieved on 06/10/2021. [Online]. Available: <https://www.gponline.com/clinical-review-benign-breast-disease/womens-health/womens-health/article/1336658>
- [9] S. Dawood, S. D. Merajver, P. Viens, P. B. Vermeulen, S. M. Swain, T. A. Buchholz, L. Y. Dirix, P. H. Levine, A. Lucci, S. Krishnamurthy, F. M. Robertson, W. A. Woodward, W. T. Yang, N. T. Ueno, and M. Cristofanilli, “International expert panel on inflammatory breast cancer: Consensus statement for standardized diagnosis and treatment,” *Annals of Oncology*, vol. 22, no. 3, pp. 515–523, 2011.
- [10] F. M. Robertson, M. Bondy, W. Yang, H. Yamauchi, S. Wiggins, S. Kamrudin, S. Krishnamurthy, H. Le-Petross, L. Bidaut, A. N. Player, S. H. Barsky, W. A. Woodward, T. Buchholz, A. Lucci, N. T. Ueno, and M. Cristofanilli, “Inflammatory Breast Cancer,” *CA Cancer J Clin*, vol. 60, no. 6, pp. 351–375, 2010, Retrieved on 03/03/2021. [Online]. Available: <https://doi.org/10.3322/caac.20082>

- [11] A. M. Gutierrez Barrera, T. M. Fouad, J. Song, R. Webster, N. Elsayegh, A. L. Wood, A. Demir, J. K. Litton, N. T. Ueno, and B. K. Arun, “BRCA mutations in women with inflammatory breast cancer,” *Cancer*, vol. 124, no. 3, pp. 466–474, 2018, Retrieved on 03/03/2021. [Online]. Available: <http://doi.wiley.com/10.1002/cncr.31069>
- [12] H. Yamauchi, W. A. Woodward, V. Valero, R. H. Alvarez, A. Lucci, T. A. Buchholz, T. Iwamoto, S. Krishnamurthy, W. Yang, J. M. Reuben, G. N. Hortobágyi, and N. T. Ueno, “Inflammatory Breast Cancer: What We Know and What We Need to Learn,” *The Oncologist*, vol. 17, no. 7, pp. 891–899, 2012, Retrieved on 03/10/2021. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3399643/>
- [13] J. Kurtz, D. Halaharvi, S. Sarwar, and M. Cripe, “A case of a bilateral inflammatory breast cancer: A case report,” *Breast Journal*, vol. 22, no. 3, pp. 342–346, 2016, Retrieved on 02/12/2021. [Online]. Available: <http://www.embase.com/search/results?subaction=viewrecord{&}from=export{&}id=L608564536{%}5Cnhttp://dx.doi.org/10.1111/tbj.12578{%}5Cnhttp://sfx.library.uu.nl/utrecht?sid=EMBASE{&}issn=15244741{&}id=doi:10.1111{%}2Ftbj.12578{&}atitle=A+case+of+a+bilateral+inflammatory+br>
- [14] A. S. Soliman, C. G. Kleer, K. Mrad, M. Karkouri, S. Omar, H. M. Khaled, A. L. Benider, F. B. Ayed, S. S. Eissa, M. S. Eissa, E. J. McSpadden, A. C. Lo, K. Toy, E. D. Kantor, Q. Xiao, C. Hampton, and S. D. Merajver, “Inflammatory breast cancer in North Africa: Comparison of clinical and molecular epidemiologic characteristics of patients from Egypt, Tunisia, and Morocco,” *Breast Disease*, vol. 33, no. 4, pp. 159–169, 2011.
- [15] D. J. P. van Uden, H. W. M. van Laarhoven, A. H. Westenberg, J. H. W. de Wilt, and C. F. J. M. Blanken-Peters, “Inflammatory breast cancer: An overview,” *Critical Reviews in Oncology/Hematology*, vol. 93, no. 2, pp. 116–126, 2015, Retrieved on 03/03/2021. [Online]. Available: <http://dx.doi.org/10.1016/j.critrevonc.2014.09.003>
- [16] D. Scutt, G. A. Lancaster, and J. T. Manning, “Breast asymmetry and predisposition to breast cancer,” *Breast Cancer Research*, vol. 8, no. 2, pp. 1–7, 2006.
- [17] Y. S. Khan and H. Sajjad, “Anatomy , Thorax , Mammary Gland,” 2021, Retrieved on 06/08/2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK547666/>
- [18] J. Garza-Salazar, A. Meneses-Garcia, and C. Arce-Salinas, *Inflammatory Breast Cancer*. Springer London, 2013.
- [19] National Cancer Institute, “Inflammatory breast cancer questions and answers. Fact sheet,” Bethesda, MD., 2006, Retrieved on 06/09/2021. [Online]. Available: <https://www.cancer.gov/types/breast/ibc-fact-sheet{#}what-are-the-symptoms-of-inflammatory-breast-cancer>

- [20] P. Garg, "Inflammatory breast cancer: a clinical diagnosis," *Singapore Medical Journal*, vol. 55, no. 03, 2014, Retrieved on 03/12/2021. [Online]. Available: <http://www.smj.org.sg/article/inflammatory-breast-cancer-clinical-diagnosis>
- [21] S. Dawood and V. Valero, *Clinical Aspect of Inflammatory Breast Cancer: Diagnosis, Criteria, Controversy*. Dordrecht: Springer Netherlands, 2012, pp. 11–19, Retrieved on 03/10/2021. [Online]. Available: https://doi.org/10.1007/978-94-007-3907-9_{_}3
- [22] American Cancer Society, "Cancer Facts & Figures 2020," pp. 1–10, 2020, Retrieved on 06/08/2021. [Online]. Available: <https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2020/cancer-facts-and-figures-2020.pdf>
- [23] J. Katz, "What is the Gail Model for breast cancer risk assessment?" 2019, Retrieved on 06/07/2021. [Online]. Available: <https://www.medscape.com/answers/1945957-180245/what-is-the-gail-model-for-breast-cancer-risk-assessment>
- [24] E. Yin-Kwee Ng, "Breast imaging: A survey," *World J Clini Oncol*, vol. 2, no. 4, pp. 171–178, 2011, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3100484/pdf/WJCO-2-171.pdf>.
- [25] P. Ricci, E. Maggini, E. Mancuso, P. Lodise, V. Cantisani, and C. Catalano, "Clinical application of breast elastography: State of the art," *European Journal of Radiology*, vol. 83, no. 3, pp. 429–437, 2014, Retrieved on 05/06/2019. [Online]. Available: <http://dx.doi.org/10.1016/j.ejrad.2013.05.007>
- [26] D. Holder, *Electrical Impedance Tomography: Methods, History, and Applications (Institute of Physics Medical Physics Series)*. Bristol: Institute of Physics Publishing, 2005.
- [27] Harvard Medical School, "Radiation risk from medical imaging," 2010, Retrieved on 03/30/2020. [Online]. Available: <https://www.health.harvard.edu/cancer/radiation-risk-from-medical-imaging>
- [28] M. Monga, "Computed tomography - An increasing source of radiation exposure: Editorial comment," *International Braz J Urol*, vol. 33, no. 6, p. 855, 2007.
- [29] S. Namasivayam, M. K. Kalra, W. E. Torres, and W. C. Small, "Adverse reactions to intravenous iodinated contrast media: A primer for radiologists," *Emergency Radiology*, vol. 12, no. 5, pp. 210–215, 2006.
- [30] T. Krouskop, T. Wheeler, F. Kallel, B. Garra, and T. Hall, "Elastic Moduli of Breast and Prostate Tissues Under Compression," *Ultrasonic Imaging*, vol. 20, no. 4, pp. 260–274, 1998.
- [31] P. Wellman, E. Dalton, and D. Krag, "Tactile Imaging of Breast Masses," *Archives of Surgery*, vol. 136, no. 2, pp. 204–208, 2001.

- [32] B. D. Price, a. P. Gibson, L. T. Tan, and G. J. Royle, “An elastically compressible phantom material with mechanical and x-ray attenuation properties equivalent to breast tissue.” *Physics in medicine and biology*, vol. 55, no. 4, pp. 1177–1188, 2010.
- [33] V. Egorov, T. Kearney, S. B. Pollak, C. Rohatgi, N. Sarvazyan, S. Airapetian, S. Browning, and A. Sarvazyan, “Differentiation of benign and malignant breast lesions by mechanical imaging,” *Breast Cancer Research and Treatment*, vol. 118, no. 1, pp. 67–80, 2009.
- [34] V. Oleksyuk, R. Rajan, F. Saleheen, D. Caroline, S. Pascarella, and C. Won, “Risk Score Based Pre-screening of Breast Tumor Using Compression Induced Sensing System,” *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4038–4045, 2018.
- [35] V. Oleksyuk, F. Saleheen, Y. Chen, and C. H. Won, “Tactile Imaging System for inclusion size and stiffness characterization,” *2015 IEEE Signal Processing in Medicine and Biology Symposium - Proceedings*, pp. 1–6, 2015.
- [36] J.-H. Lee and C.-H. Won, “The tactile sensation imaging system for embedded lesion characterization,” *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 2, pp. 452–458, 2013.
- [37] IDS Imaging Development Systems GmbH, “UI3240CP_NIR Datasheet,” 2015, Retrieved on 04/15/2020. [Online]. Available: <https://www.1stvision.com/cameras/IDS/dataman/UI3240CP-NIR-Datasheet.pdf>
- [38] W. Moser, “Third generation tactile imaging system with new interface, calibration method and wear indication,” MS Thesis, Temple University, 2016.
- [39] F. Imani, M. Daoud, M. Moradi, P. Abolmaesumi, and P. Mousavi, “Tissue classification using depth-dependent ultrasound time series analysis: in-vitro animal study,” *SPIE Medical Imaging*, vol. 7968, 2011, Retrieved on 02/06/2020. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=726843>
- [40] N. Uniyal, H. Eskandari, P. Abolmaesumi, S. Sojoudi, P. Gordon, L. Warren, R. Rohling, S. Salcudean, and M. Moradi, “Ultrasound RF time series for classification of breast lesions.” *IEEE transactions on medical imaging*, vol. 34, no. 2, pp. 652–661, 2014, Retrieved on 02/06/2020. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/25350925>
- [41] N. Hatami, Y. Gavet, and J. Debayle, “Classification of Time-Series Images Using Deep Convolutional Neural Networks,” in *The 10th International Conference on Machine Vision (ICMV 2017)*, 2017, Retrieved on 03/12/2020. [Online]. Available: <http://arxiv.org/abs/1710.00886>
- [42] N. Hatami, Y. Gavet, and J. Debayl, “Bag of recurrence patterns representation for time-series classification,” *Pattern Analysis and Applications*, pp. 1–11, 2018, 06/30/2021. [Online]. Available: <https://arxiv.org/pdf/1803.11111.pdf>

- [43] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2015-Janua, pp. 3939–3945, 2015.
- [44] M. D. Riley, *Speech Time-Frequency Representations*, J. Allen, Ed. Boston: Kluwer Academic Publisher, 1989.
- [45] D. M. Howard and D. T. Murphy, *Voice science, acoustics and recording*. San Diego, California: Plural Publishing, Inc., 2007.
- [46] S. A. Fulop, *Speech Spectrum Analysis*. Springer, 2011, Retrieved on 04/12/2020. [Online]. Available: <http://link.springer.com/10.1007/978-3-642-17478-0>
- [47] J. Smith, *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*, 2nd ed. W3K Publishing, 2007, Retrieved on 04/12/2020. [Online]. Available: <http://ccrma.stanford.edu/~jos/mdft/>
- [48] Center for Plasticity of the Brain, “How Your Sense of Touch Works,” 2016, Retrieved on 09/24/2018. [Online]. Available: <https://www.plasticitybraincenters.com/media/how-your-sense-of-touch-works/>
- [49] A. Bicchi, M. Buss, M. O. Ernst, and A. Peer, *The sense of touch and its rendering. Progress in Haptics Research*, B. Siciliano and O. G. Khatib, Eds. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2008.
- [50] J.-H. Lee and C. H. Won, “High Resolution Tactile Imaging Sensor using Total Internal Reflection and Non-Rigid Pattern Matching,” *IEEE SENSORS JOURNAL*, vol. 11, no. 9, pp. 2084 – 2093, 2011.
- [51] E. Mangano, “Preloading, and How It Affects Your Mechanical Test,” 2014, Retrieved on 07/18/2019. [Online]. Available: <https://www.instron.us/en-us/our-company/press-room/blog/2014/august/preloading-and-how-it-affects-your-mechanical-test>
- [52] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986, Retrieved on 04/20/2020. [Online]. Available: <http://ieeexplore.ieee.org.libproxy.temple.edu/stamp/stamp.jsp?tp={&}arnumber=4767851{&}isnumber=4767846>
- [53] MATLAB MathWorks, “Image Processing Toolbox Documentation,” 2019, Retrieved on 6/13/2019. [Online]. Available: [https://www.mathworks.com/help/images/index.html?s\[_\]tid=CRUX\[_\]lftnav](https://www.mathworks.com/help/images/index.html?s[_]tid=CRUX[_]lftnav)
- [54] D. B. Lenat and R. V. Guha, *Building Large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [55] A. Gulli and S. Pal, *Deep Learning with Keras*. Birmingham: Packt Publishing, 2017.

- [56] Y. Lecun, "Generalization and network design strategies. Technical Report CRG-TR-89-4," University of Toronto, Toronto, Tech. Rep., 1989, Retrieved on 06/30/2021. [Online]. Available: <http://yann.lecun.com/exdb/publis/pdf/lecun-89.pdf>
- [57] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [58] Z. Jiao, X. Gao, Y. Wang, and J. Li, "A deep feature based framework for breast masses classification," pp. 221–231, 2016, Retrieved on 05/01/2020. [Online]. Available: <http://dx.doi.org/10.1016/j.neucom.2016.02.060>
- [59] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–14, 2016, Retrieved on 06/05/2021. [Online]. Available: <https://arxiv.org/pdf/1511.07289.pdf>
- [60] Y. T. Zhou and R. Chellappa, "Computation of optical flow using a neural network," *IEEE 1988 International Conference on Neural Networks*, pp. 71–78 vol.2, 1988.
- [61] G. A. Walters, "Max-pooling," 2019, Retrieved on 12/04/2020. [Online]. Available: <https://austingwalters.com/convolutional-neural-networks-cnn-to-classify-sentences/max-pooling/>
- [62] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.
- [63] F. Chollet, *Deep learning with Python*. Manning Publications Co., 2018.
- [64] G. Lamouche, B. F. Kennedy, K. M. Kennedy, C.-E. Bisailon, A. Curatolo, G. Campbell, V. Pazos, and D. D. Sampson, "Review of tissue simulating phantoms with controllable optical, mechanical and structural properties for use in optical coherence tomography," *Biomedical Optics Express*, vol. 3, no. 6, p. 1381, 2012, Retrieved on 05/06/2021. [Online]. Available: <https://www.osapublishing.org/boe/abstract.cfm?uri=boe-3-6-1381>
- [65] T. Boehm, A. Hochmuth, A. Malich, J. R. Reichenbach, M. Fleck, and W. A. Kaiser, "Contrast-enhanced near-infrared laser mammography Contrast-Enhanced Near-Infrared Laser Mammography with a prototype breast scanner: Feasibility study with tissue phantoms and preliminary results of imaging experimental tumors," *Investigative Radiology*, vol. 36, no. 10, pp. 573–581, 2001.
- [66] A. M. D. Grand, S. J. Lomnes, D. S. Lee, M. Pietrzykowski, S. Ohnishi, T. G. Morgan, A. Gogbashian, R. G. Laurence, and J. V. Frangioni, "Tissue-Like Phantoms for Near-Infrared Fluorescence Imaging System Assessment and the Training of Surgeons," *J Biomed Opt*, vol. 11, no. 1, pp. 1–21, 2008.

- [67] C. A. Giller, H. Liu, P. Gurnani, S. Victor, U. Yazdani, and D. C. German, "Validation of a near-infrared probe for detection of thin intracranial white matter structures," *Journal of Neurosurgery*, vol. 98, no. 6, pp. 1299–1306, 2003, Retrieved on 03/05/2020. [Online]. Available: <http://thejns.org/doi/10.3171/jns.2003.98.6.1299>
- [68] R. Choe, "Diffuse optical tomography and spectroscopy of breast cancer and fetal brain," Dissertation, University of Pennsylvania, 2005.
- [69] J. C. Hebden, B. D. Price, A. P. Gibson, and G. Royle, "A soft deformable tissue-equivalent phantom for diffuse optical tomography," *Physics in Medicine and Biology*, vol. 51, no. 21, pp. 5581–5590, 2006.
- [70] L. Sallaway, S. Magee, J. Shi, O. Lehmann, F. Quivira, K. Tgavalekos, D. H. Brooks, S. Muftu, W. Meleis, R. H. Moore, D. Kopans, and K. T. Wan, "Detecting Solid Masses in Phantom Breast Using Mechanical Indentation," *Experimental Mechanics*, vol. 54, no. 6, pp. 935–942, 2014.
- [71] J. R. Mourant, T. Fuselier, J. Boyer, T. M. Johnson, and I. J. Bigio, "Predictions and measurements of scattering and absorption over broad wavelength ranges in tissue phantoms," *Applied Optics*, vol. 36, no. 4, p. 949, 1997, Retrieved on 05/05/2020. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/18250760>{% }5Cn<https://www.osapublishing.org/abstract.cfm?URI=ao-36-4-949>
- [72] V. Egorov and A. P. Sarvazyan, "Mechanical imaging of the breast," *IEEE Transactions on Medical Imaging*, vol. 27, no. 9, pp. 1275–1287, 2008.
- [73] A. Sahu, F. Saleheen, V. Oleksyuk, C. McGoverin, N. Pleshko, A. Hossein Harati Nejad Torbati, J. Picone, S. Member, K. Sorenmo, and C.-H. Won, "Characterization of Mammary Tumors Using Noninvasive Tactile and Hyperspectral Sensors," *IEEE SENSORS JOURNAL*, vol. 14, no. 10, pp. 3337 – 3344, 2014.
- [74] V. Oleksyuk, "Young's Modulus Estimation of PDMS Samples using Instron Material Testing Machine," Temple University, Philadelphia PA, Tech. Rep., 2019.
- [75] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, no. 106, pp. 1–16, 2013, Retrieved on 05/20/2021. [Online]. Available: <https://bmcbioinformatics.biomedcentral.com/track/pdf/10.1186/1471-2105-14-106.pdf>
- [76] J. Qin, K. Chao, M. S. Kim, R. Lu, and T. F. Burks, "Hyperspectral and multispectral imaging for evaluating food safety and quality," *Journal of Food Engineering*, vol. 118, no. 2, pp. 157–171, 2013, Retrieved on 03/04/2020. [Online]. Available: <http://dx.doi.org/10.1016/j.jfoodeng.2013.04.001>
- [77] S. Nioka and B. Chance, "NIR spectroscopic detection of breast cancer," *Technology in Cancer Research and Treatment*, vol. 4, no. 5, pp. 497–512, 2005.

- [78] R. H. Wilson, K. P. Nadeau, F. B. Jaworski, B. J. Tromberg, and A. J. Durkin, "Review of short-wave infrared spectroscopy and imaging methods for biological tissue characterization," *Journal of Biomedical Optics*, vol. 20, no. 3, p. 030901, 2015.
- [79] O. Y. Hay, M. Cohen, I. Nitzan, Y. Kasirer, S. Shahroor-Karni, Y. Yitzhaky, S. Engelberg, and M. Nitzan, "Pulse oximetry with two infrared wavelengths without calibration in extracted arterial blood," *Sensors (Switzerland)*, vol. 18, no. 10, 2018.
- [80] J. Ruiz, F. Nouizi, J. Cho, J. Zheng, Y. Li, J.-H. Chen, M.-Y. Su, and G. Gulsen, "Breast density quantification using structured-light-based diffuse optical tomography simulations," *Applied Optics*, vol. 56, no. 25, p. 7146, 2017.
- [81] Y. Yang, N. Liu, A. Sassaroli, and S. Fantini, "Near-infrared spectral imaging of the female breast for quantitative oximetry in optical mammography," *Applied Optics*, vol. 48, no. 10, 2009.
- [82] S. L. Jacques, "Optical properties of biological tissues: a review," *Physics in Medicine and Biology*, vol. 58, no. 11, pp. R37–R61, 2013, Retrieved on 05/05/2020. [Online]. Available: <http://stacks.iop.org/0031-9155/58/i=11/a=R37?key=crossref.e58b67eb50f9f6507dd5e75b7744fa07>
- [83] QImaging, "Retiga EXi Camera Datasheet," 2014, Retrieved on 04/15/2020. [Online]. Available: https://meyerinst.com/library/uploads/RetigaEXi_{_}fast1394.pdf
- [84] A. Cerussi, "Main NIR functional absorbers in tissues: deoxy hemoglobin, oxy hemoglobin, water and lipids," 2009, Retrieved on 11/08/2020. [Online]. Available: <https://sites.google.com/site/dosiatbli/what-is-dosi-/general-description>
- [85] J. Modersitzki, *Numerical Methods for Image Registration*. New York: Oxford University Press, 2004.
- [86] R. Gonzalez and R. Woods, *Digital Image Processing. Upper Saddle River, N.J: Prentice Hall; 2002.*, 3rd ed. Pearson/Prentice Hall, 2008.
- [87] M. A. Calin, S. V. Parasca, D. Savastru, and D. Manea, "Hyperspectral Imaging in the Medical Field: Present and Future," *Applied Spectroscopy Reviews*, vol. 49, no. 6, pp. 435–447, 2014, Retrieved on 10/20/2020. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84891297229{&}partnerID=tZOtx3y1>
- [88] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in Hyperspectral Image and Signal Processing: A Comprehensive Overview of the State of the Art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, 2017.
- [89] T. A. Wilson, "Perceptual-based image fusion for hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 1007–1017, 1997.

- [90] R. Raghavendra and C. Busch, "Novel image fusion scheme based on dependency measure for robust multispectral palmprint recognition," *Pattern Recognition*, vol. 47, no. 6, pp. 2205–2221, 2014, Retrieved on 07/12/2020. [Online]. Available: <http://dx.doi.org/10.1016/j.patcog.2013.12.011>
- [91] S. Sonnad, "A survey on fusion of multispectral and panchromatic images for high spatial and spectral information," *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 177–180, 2016.
- [92] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and Multispectral Data Fusion: A Comparative Review," *IEEE Geoscience and Remote Sensing Magazine*, no. june, pp. 1–25, 2017.
- [93] F. M. Riese and S. Keller, *Supervised, semi-supervised, and unsupervised learning for hyperspectral regression*. Springer International Publishing, 2020.
- [94] M. Boisserie-Lacroix, M. Debled, C. Tunon De Lara, G. Hurtevent, M. Asad-Syed, and S. Ferron, "The inflammatory breast: Management, decision-making algorithms, therapeutic principles," *Diagnostic and Interventional Imaging*, vol. 93, no. 2, pp. 126–136, 2012.
- [95] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*, 2nd ed. O'Reilly Media Inc., 2019.
- [96] G. Fiaschetti, J. E. Browne, M. Cavagnaro, L. Farina, and G. Ruvio, "Tissue Mimicking Materials for Multi-Modality Breast Phantoms," *2018 2nd URSI Atlantic Radio Science Meeting, AT-RASC 2018*, no. June, 2018.
- [97] D. A. Miranda, K. L. Cristiano, and J. C. Gutiérrez, "Breast phantom for mammary tissue characterization by near infrared spectroscopy," *Journal of Physics: Conference Series*, vol. 466, no. 1, pp. 3–6, 2013.
- [98] G. M. Spirou, A. A. Oraevsky, I. Alex Vitkin, and W. M. Whelan, "Optical and acoustic properties at 1064 nm of polyvinyl chloride-plastisol for use as a tissue phantom in biomedical optoacoustics," *Physics in Medicine and Biology*, vol. 50, no. 14, 2005.
- [99] A. I. Chen, M. L. Balter, M. I. Chen, D. Gross, S. K. Alam, T. J. Maguire, and M. L. Yarmush, "Multilayered tissue mimicking skin and vessel phantoms with tunable mechanical, optical, and acoustic properties," *Med. Phys.*, vol. 43, no. 6, pp. 3117–3131, 2016.
- [100] F. Saleheen and C. H. Won, "Bimodal Dynamic Imaging System for Embedded Inclusion Characterization," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 6062–6071, 2016.

- [101] R. R. Anderson, J. a. Parrish, B. R. Rox Anderson, and M. Parrish, "The Optics of Human Skin," *The Journal of investigative dermatology*, vol. 77, no. 1, pp. 13–9, 1981, Retrieved on 07/12/2020. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/22520927>
- [102] H. R. Kang, *Computational Color Technology*. SPIE, 2006, Retrieved on 07/12/2020. [Online]. Available: <https://doi.org/10.1117/3.660835.ch16>(<https://doi.org/10.1117/3.660835.ch16>)
- [103] Harrick Scientific Products, "What is Kubelka-Munk?" 2018, Retrieved on 07/12/2020. [Online]. Available: <http://mmrc.caltech.edu/FTIR/Literature/Kubelka-Munk.pdf>
- [104] S. Yadav, S. Hartkop, P. Y. Cardenas, R. Ladkany, A. Halalau, S. Shoichet, M. Madens, and D. Zakalik, "Utilization of a breast cancer risk assessment tool by internal medicine residents in a primary care clinic: Impact of an educational program," *BMC Cancer*, vol. 19, no. 1, pp. 1–7, 2019.
- [105] C. Ming, V. Viassolo, N. Probst-Hensch, P. O. Chappuis, I. D. Dinov, and M. C. Katapodi, "Machine learning techniques for personalized breast cancer risk prediction: Comparison with the BCRAT and BOADICEA models," *Breast Cancer Research*, vol. 21, no. 1, pp. 1–11, 2019.
- [106] M. H. Gail, L. A. Brinton, D. P. Byar, D. K. Corle, S. B. Green, C. Schairer, and J. J. Mulvihill, "Projecting individualized probabilities of developing breast cancer for white females who are being examined annually," *Journal of the National Cancer Institute*, vol. 81, no. 24, pp. 1879–1886, 1989.
- [107] E. B. Claus, N. Risch, and W. D. Thompson, "Autosomal dominant inheritance of early-onset breast cancer. Implications for risk prediction." *Cancer*, vol. 73, no. 3, pp. 641–651, 1994.
- [108] J. Tyrer, S. Duffy, and J. Cuzick, "A breast cancer prediction model incorporating familial and personal risk factors." *Stat Med*, vol. 23, no. 7, pp. 1111–1130, 2004.
- [109] M. Kallenberg, K. Petersen, M. Nielsen, A. Y. Ng, P. Diao, C. Igel, C. M. Vachon, K. Holland, R. R. Winkel, N. Karssemeijer, and M. Lillholm, "Unsupervised Deep Learning Applied to Breast Density Segmentation and Mammographic Risk Scoring," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1322–1331, 2016.
- [110] F. Heinemann, G. Birk, and B. Stierstorfer, "Deep learning enables pathologist-like scoring of NASH models," *Scientific Reports*, vol. 9, no. 1, pp. 1–10, 2019.
- [111] J. P. Costantino, M. H. Gail, D. Pee, S. Anderson, C. K. Redmond, J. Benichou, and H. S. Wieand, "Validation studies for models projecting the risk of invasive and total breast cancer incidence," *Journal of the National Cancer Institute*, vol. 91, no. 18, pp. 1541–1548, 1999.

APPENDIX A

HARDWARE CHARACTERISTICS OF TIP PROTOTYPES

Three generations of TIP prototypes are summarized in Table A.1.

Table A.1: Comparative hardware characteristics for three generations of TIP prototypes

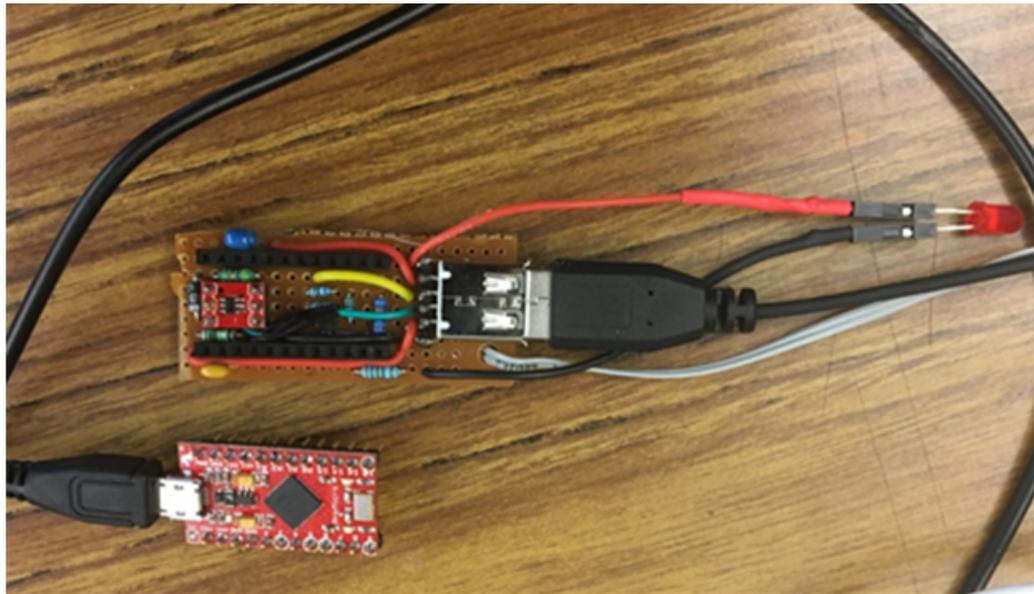
Parameters	TIP 1E	TIP 3E	TIP 4E
Camera	AVT Guppy F038C	IDS UI-1220SE-M-GL	IDS UI-3240CP-NIR
Interface	IEEE 1394a Sony ICX428	USB 2.0 Aptina	USB 3.0 E2V
Sensor	CCD Type 1/2" Mono	CMOS Type 1/3" Mono	CMOS Type 1/1.84" Mono
Cell size, $\mu\text{m} \times \mu\text{m}$	8.4 \times 9.8	6 \times 6	5.3 \times 5.3
Image size, pixels \times pixels	768 \times 492	752 \times 480	1024 \times 1280
Pixel depth,bit		8	
Lens	Kowa 16mm / F1.4 C2/3"	Kowa CCTV 6mm / F1.2 C1/2"	Schneider 12mm / F1.4 C2/3" VIS-NIR
LEDs	4 \times White D 3 mm 1500 MCD	4 \times White D 3 mm 1200 MCD	4 \times White D 3 mm 1100 MCD
Spacers	#8 \times 1/4" Aluminum Spacers 0.170" ID \times 0.250" OD \times 0.250" Matte		
LED Circuit control	1K potentiometer	Arduino Pro Micro 5V/16MHz	
Case	-	3D printed design	
Sensing element size, mm (L \times W \times H)	23 \times 20 \times 12		
Power source	9V	USB	
Force sensor	Mark-10 (external)	FC22 (built in)	

APPENDIX B

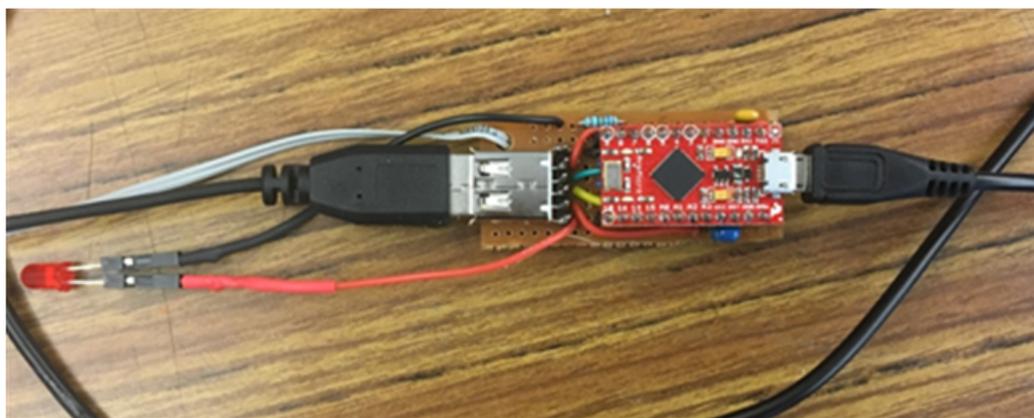
TIP MICROCONTROLLER CIRCUIT IMPLEMENTATION

Microcontroller Hardware Implementation

The circuit design of the microcontroller circuit was given Fig. 3.2. Here we present hardware implementation of the circuit. Fig. B.1(a) shows the circuit implementation and the wired microcontroller. Fig. B.1(b) illustrates how the microcontroller is attached on the top of the circuit to minimize the design size.



(a)



(b)

Figure B.1: TIP Microcontroller circuit implementation

Microcontroller Firmware Installation

For TIP 4E, the tactile images have to be collected with force reading from the load cell. At the same time, the LED brightness should be controlled. To ensure these functionalities, a firmware code is required to be loaded into an Arduino microcontroller.

Pre-requisites:

1. Arduino Pro-Micro Dev 12640.
2. Arduino IDE version 1.6.10 (download from <https://www.arduino.cc/en/Main/OldSoftwareReleases>).
3. Follow the instruction in this url as an example: <https://learn.adafruit.com/adafruit-feather-32u4-basic-proto/using-with-arduino-ide>.

Steps for uploading the firmware:

1. Open Arduino IDE
2. Find the file in the folder:
 - Documents -> Arduino -> TIS_Firmware_v1_3.
3. Open the file from IDE:
 - File -> Open -> TIS_Firmware_v1_3.ino.
4. Then select:
 - Tool - Board: "Adafruit Feather 32u4".
5. Check if the COM port is correct.
6. Upload the program

Microcontroller Firmware Script

```
// Filename: TIS_Firmware_v1_3.ino
#include <TimerOne.h>

//#include <TimerOne.h>
#include <avr/io.h>
#include <avr/interrupt.h>
/*
 * Version 1.3
 * Supports TimerOne-based LED dimming.
```

```

    Send "b" to brighten and "d" to dim the LEDs
    * Serial comms are at 57.6kbaud
    * The force pin is read 5 times and averaged before being output.
    *
    */
//Declare pins for PWM out, Force in
const int pwmpin=9;
const int forceInPin = A0;
const int switchPin = 7;
//const int debugPin = 8;

//Delare other variable
char state=0; //state for FSM
int trigger_mode = 0;
int force_readings[5]={0,0,0,0,0}; //5-force array for quick sum/average
int force_reading; //final force value to commit
uint16_t force_buffer[1005] = {0}; //Buffer to collect force
                                values to download after a capture
String force_reading_STR = "0"; //Force gauge reading as string
float dutyCycle = 99.0; //dimming variable
volatile unsigned int triggerCount= 1;
volatile unsigned int switchState=1;
int i=0; //for loop counter

ISR(INT6_vect){
    switchState = digitalRead(switchPin);
}

void setup() {
    pinMode(pwmpin, OUTPUT); //set the I/O pin to output
    Timer1.initialize(100); //set timer1 to 10Khz (100 us)
    Timer1.pwm(pwmpin,128); //set duty cycle to 25% (out ot 1023)
    /*32u4 Datasheet 11.0.2 Interrupt mode setup
    *ISC61 ISC60 Triggered By
    * 0 0 INT6 low
    * 0 1 Any logical change on INT6
    * 1 0 Falling edge between two INT6 samples
    * 1 1 Rising edge between two INT6 samples
    */
    EICRB |= (1<<ISC60)|(1<<ISC61); //External Interrupt Control
    Register B. Reg A controls ints 0 to 3. 1/1=Rising Edge detection
    EIMSK |= (1<<INT6); //Ext. Int. Mask register.
    This activates the interrupt on Pin 7 for ProMirco.

```

```

    It is Pin1 on the 32u4
    noInterrupts();
    Serial.begin(57600);    //Start 57600kbps serial
    interrupts();
}

void loop() {
    //Begin Serial Communication
    delay(1);

    /*Watch for switch interrupt
    * Upon interrupt, take a 5-averaged force reading
    * and insert it into the force buffer. If this is
    * software mode, send an ascii 'p' to signal a photo
    */
    if(switchState == 1){
        if(trigger_mode==0){
            //digitalWrite(debugPin, HIGH);
            switchState = 0; //clear switch state
            force_readings[0] = analogRead(forceInPin);
            force_readings[1] = analogRead(forceInPin);
            force_readings[2] = analogRead(forceInPin);
            force_readings[3] = analogRead(forceInPin);
            force_readings[4] = analogRead(forceInPin);
            force_reading = 0.2* int(addnum(force_readings));
            force_buffer[triggerCount] = force_reading;
            //digitalWrite(debugPin, LOW);
            triggerCount++;
        }
        if(trigger_mode==1){
            delay(600);
            Serial.print('p');
        }
    }

    while (Serial.available())
    {
        state = Serial.read(); // used to read incoming data

        /*This section defines the functions you can access
        * by sending an ASCII request over serial.
        * The Print function always has lower priority
        * than any timer interrupt, so the dimming should be safe
        */
    }
}

```

```

switch(state)// see what was sent to the board
{
  //Trigger Mode: send an 's' or 'h'
  //to toggle between the flash trigger or
  //acquiring when an 'r' is sent
  case 's': //software trigger
    trigger_mode = 1;
    //Serial.print("st");
  break;
  case 'h': //hardware trigger
    trigger_mode = 0;
    //Serial.print("ht");
  break;

  /*Brightness: Sending a 'b' will
  * increment the Brightness by one step and 'd' will
  * decrement it by one step and return
  * the current level with a 'b' on the end
  */
  case 'b': //brighten LED
    if (dutyCycle > 0)
    {
      dutyCycle = --dutyCycle; //decrement PWM duty cycle by 1
      Timer1.pwm(pwmpin, (dutyCycle / 100) * 1023);
    }
    // return the brightness
    Serial.print(String(100-(int)dutyCycle)+"b");
  break;
  case 'd': //dim LED
    if (dutyCycle < 99)
    {
      dutyCycle = ++dutyCycle; //increment PWM duty cycle by 1
      Timer1.pwm(pwmpin, (dutyCycle / 100) * 1023);
    }
    // return the brightness
    Serial.print(String(100-(int)dutyCycle)+"b");
  break;

  /*Force Request: This is the less accurate force
  * function that is called by the GUI to update the
  * live force readout. It is called with 'f' every ~250ms.
  * 5 measurements are taken and averaged to account for
  * noise. The force should always be 4 digits appended with 'f'
  */
}

```

```

case 'f': // if f was sent, return force reading
    force_reading = analogRead(forceInPin);

    if (force_reading < 1000){
        force_reading_STR = String(force_reading);
        force_reading_STR = "0"+force_reading_STR+"f";
        Serial.print(force_reading_STR);
    }
    else{
        force_reading_STR = String(force_reading);
        force_reading_STR = force_reading_STR+"f";
        Serial.print(force_reading_STR);
    }
break;

/*Reading: This is the software trigger control
* for the force measurement with 'r'. It takes 5 readings,
* sums them, then takes an average. Again, it always
* returns 4 digits with a 'r' on the end.
*/
case 'r': // if r was sent, return force reading with reading flag
    force_readings[0] = analogRead(forceInPin);
    force_readings[1] = analogRead(forceInPin);
    force_readings[2] = analogRead(forceInPin);
    force_readings[3] = analogRead(forceInPin);
    force_readings[4] = analogRead(forceInPin);
    force_reading = 0.2*int(addnum(force_readings));

    if (force_reading < 1000){
        force_reading_STR = String(force_reading);
        force_reading_STR = "0"+force_reading_STR+"r";
        Serial.print(force_reading_STR);
    }
    else{
        force_reading_STR = String(force_reading);
        force_reading_STR = force_reading_STR+"r";
        Serial.print(force_reading_STR);
    }
break;

/*Download Force Buffer: Sending a 'q' will
* start returning every force from the earliest

```

```

* to the most recent. The last value is not a force,
* but the total number of readings. This will
* give a checksum for botched transfers.
*/
case 'q': //query force list
    i=0;
    while(force_buffer[i] != 0){
        force_reading_STR = String(force_buffer[i]);
        if (force_buffer[i]<1000 && force_buffer[i]>99){
            force_reading_STR = "0"+force_reading_STR+"q";
        }
        if (force_buffer[i]<100){
            force_reading_STR = "00"+force_reading_STR+"q";
        }
        if (force_buffer[i]>=1000){
            force_reading_STR = force_reading_STR+"q";
        }

        Serial.print(force_reading_STR);
        delay(2); //it gets jumbled on the other end w/o this
        i++;
    }
    Serial.print(String(triggerCount)+"t");
break;

/*Clear Force Buffer: Sending an x
* will reset the buffer and the trigger count and return 'x'
*/
case 'x': //clear force list
    i=0;
    while(force_buffer[i] != 0){
        force_buffer[i] = 0;
        i++;
    }
    triggerCount = 0;
    Serial.print("x");
break;

//default case is just exit the loop. Should never happen.
default:
    break;

} //end case
delay(1); //for loop stability

```

```
    } //end while
} //end loop

//summing function to create a 5-force reading average
//pointing to the array object prevents needless copying
int addnum(int *ptr)
{
    int index, total=0;
    for(index = 0; index < 5; index++){
        if(*(ptr+index)<1000){
            total += *(ptr+index);}
    }
    return(total);
}
```

APPENDIX C

TIP LEDs ALIGNMENT PROCEDURE

Materials and Tools

The list of materials and tools for LED alignment is given in Table C.1.

Table C.1: List of materials and tools

Number	Description	Manuf./Destr.	Q-ty
1	CMOS Camera IDS UI-3240CP-NIR	1stVision	1
2	Schneider Lens (21-1001951) 2/3", 12mm, 1.4	1stVision	1
3	USB 3.0 Cable	Amazon	1
4	FC22 Force Sensor	Phidgets	1
5	Standard LEDs Through Hole White Clear 1100mcd	Mouser El.	4
6	Arduino Pro Micro - 5V/16MHz	Sparkfun	1
8	Heat-Resistant Borosilicate Glass 1-3/4" Diameter, 1/8" Thick	McMaster-Carr	1
9	#8 × 1/4" Aluminum Spacers 0.170" ID × 0.250" OD × 0.250" L (unpolished, not shiny)	WidgetCo	4
10	3D Printed Case Tube, Tube Cap, Camera holder, Spectral tip, TIS tip	Temple University	1

Alignment Procedure

To align the LED light, one has to obtain parts listed in Table C.1 and strictly follow the directions. The camera and its configuration highly influence the output of tactile image quality.

Camera configuration

The focal length of the used lens is fixed. Aperture knob has to be adjusted to the 1.4. The lens tube has to touch the glass plate. Use the TIS4E_parameters.ini settings for the IDS camera, given in the next section.

Spacers

The spacers are attached to the front-end as shown on Fig. C.1. They are perpendicular (90 degrees) to the corresponding side of the PDMS probe surface. It is important to use the listed unpolished spacers (Table C.1).

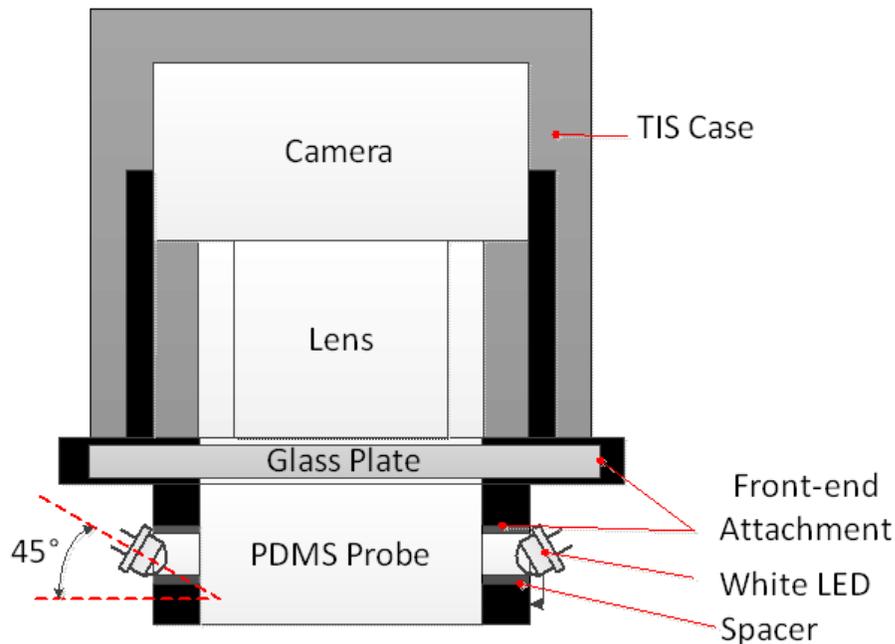


Figure C.1: LED and spacers configuration schematics in TIP design

LEDs

LEDs are placed into spacers at the far end. The direction of LED light is towards the outer contact surface of the PDMS probe. Fig. 1 shows the LEDs alignment. The LEDs are tilted approximately 45 degrees towards the outer surface of PDMS. LEDs have to be dimmed to very low level (97 - 99 level using Arduino serial port configuration).

GUI parameters

LED brightness at 50. Can save up to 230 images (TactileQtGUI.pro).

Fig. C.2 demonstrates the hardware implementation of LEDs alignment.



Figure C.2: Alignment implementation

Camera Parameters Initialization Software

[Versions]

ueye_api_64.dll=4.81.12
ueye_usb_64.sys=4.81.0012
ueye_boot_64.sys=4.81.0012

[Sensor]

Sensor=UI324xCP-NIR
Sensor bit depth=0
Sensor source gain=0
FPN correction mode=0
Black reference mode=0
Sensor digital gain=0

[Image size]

Start X=0
Start Y=0
Start X absolute=1
Start Y absolute=1
Width=1280
Height=1024
Binning=0
Subsampling=0

[Scaler]

Mode=0
Factor=1.000000

[Multi AOI]

Enabled=0
Mode=1
x1=0
x2=0
x3=0
x4=0
y1=0
y2=0
y3=0
y4=0

[Shutter]

Mode=2

Linescan number=0

[Log Mode]

Mode=3

Manual value=0

Manual gain=0

[Timing]

Pixelclock=30

Extended pixelclock range=0

Framerate=19.992143

Exposure=0.999579

Long exposure=0

Dual exposure ratio=0

[Selected Converter]

IS_SET_CM_RGB32=1

IS_SET_CM_RGB24=1

IS_SET_CM_RGB16=1

IS_SET_CM_RGB15=1

IS_SET_CM_Y8=1

IS_SET_CM_RGB8=1

IS_SET_CM_BAYER=8

IS_SET_CM_UYVY=1

IS_SET_CM_UYVY_MONO=1

IS_SET_CM_UYVY_BAYER=1

IS_CM_CBYCRY_PACKED=0

IS_SET_CM_RGBY=8

IS_SET_CM_RGB30=1

IS_SET_CM_Y12=1

IS_SET_CM_BAYER12=8

IS_SET_CM_Y16=1

IS_SET_CM_BAYER16=8

IS_CM_BGR12_UNPACKED=1

IS_CM_BGRA12_UNPACKED=1

IS_CM_JPEG=0

IS_CM_SENSOR_RAW10=8

IS_CM_MONO10=1

IS_CM_BGR10_UNPACKED=1

IS_CM_RGBA8_PACKED=1
IS_CM_RGB8_PACKED=1
IS_CM_RGBY8_PACKED=8
IS_CM_RGB10V2_PACKED=8
IS_CM_RGB12_UNPACKED=1
IS_CM_RGBA12_UNPACKED=1
IS_CM_RGB10_UNPACKED=1
IS_CM_RGB8_PLANAR=1

[Parameters]

Colormode=6
Gamma=1.000000
Hardware Gamma=0
Blacklevel Mode=0
Blacklevel Offset=0
Hotpixel Mode=2
Hotpixel Threshold=0
Sensor Hotpixel=1
GlobalShutter=0
AllowRawWithLut=0

[Gain]

Master=0
Red=0
Green=0
Blue=0
GainBoost=0

[Processing]

EdgeEnhancementFactor=0
RopEffect=0
Whitebalance=0
Whitebalance Red=1.000000
Whitebalance Green=1.000000
Whitebalance Blue=1.000000
Color correction=0
Color_correction_factor=1.000000
Color_correction_satU=100
Color_correction_satV=100
Bayer Conversion=1
JpegCompression=0
NoiseMode=0

ImageEffect=0
LscModel=0
WideDynamicRange=0

[Auto features]
Auto Framerate control=0
Brightness exposure control=0
Brightness gain control=0
Auto Framerate Sensor control=0
Brightness exposure Sensor control=0
Brightness gain Sensor control=0
Brightness exposure Sensor control photometry=0
Brightness gain Sensor control photometry=0
Brightness control once=0
Brightness reference=128
Brightness speed=50
Brightness max gain=100
Brightness max exposure=49.975579
Brightness Aoi Left=0
Brightness Aoi Top=0
Brightness Aoi Width=1280
Brightness Aoi Height=1024
Brightness Hysteresis=2
AutoImageControlMode=2
AutoImageControlPeakWhiteChannel=0
AutoImageControlExposureMinimum=0.000000
Auto WB control=0
Auto WB type=2
Auto WB RGB color model=1
Auto WB RGB color temperature=0
Auto WB offsetR=0
Auto WB offsetB=0
Auto WB gainMin=0
Auto WB gainMax=100
Auto WB speed=50
Auto WB Aoi Left=0
Auto WB Aoi Top=0
Auto WB Aoi Width=1280
Auto WB Aoi Height=1024
Auto WB Once=0
Auto WB Hysteresis=2
Brightness Skip Frames Trigger Mode=4
Brightness Skip Frames Freerun Mode=4
Auto WB Skip Frames Trigger Mode=4

Auto WB Skip Frames Freerun Mode=4

[Trigger and Flash]

Trigger mode=0
Trigger timeout=200
Trigger delay=0
Trigger debounce mode=0
Trigger debounce delay time=1
Trigger burst size=1
Trigger prescaler frame=1
Trigger prescaler line=1
Trigger input=1
Flash strobe=0
Flash delay=0
Flash duration=0
Flash auto freerun=0
PWM mode=0
PWM frequency=20000000
PWM dutycycle=20000000
GPIO state=0
GPIO direction=3
GPIO1 Config=2
GPIO2 Config=2

[Sequence AOI]

NumberUsedAOI=0
StartX1=0
StartY1=0
Gain1=0
Exposure1=0.000000
ReadoutCycle1=1
BinningMode1=0
SubsamplingMode1=0
ScalerFactor1=0.000000
DetachImageParameter1=0
StartX2=0
StartY2=0
Gain2=0
Exposure2=0.000000
ReadoutCycle2=1
BinningMode2=0
SubsamplingMode2=0
ScalerFactor2=0.000000

DetachImageParameter2=0
StartX3=0
StartY3=0
Gain3=0
Exposure3=0.000000
ReadoutCycle3=1
BinningMode3=0
SubsamplingMode3=0
ScalerFactor3=0.000000
DetachImageParameter3=0

[Vertical AOI Merge Mode]
Mode=0
Position=0
Additional Position=0
Height=1

[Memory]
Camera memory mode=0

APPENDIX D

YOUNG'S MODULUS TEST OF PDMS SAMPLES WITH INSTRON 5944

Here, Young's moduli of Polydimethyl siloxane (PDMS) samples were obtained using the compression technique with the Instron 5944 testing machine. The purpose of this experiment was to measure the Young's Modulus of PDMS phantom components used in Tactile Imaging Probe (TIP) experiments. PDMS of seventeen different mixing ratios were tested. The mean Young's modulus of PDMS was found to be 32.76 kPa, 37.41 kPa, 94.41 kPa, 129.70 kPa, 177.35 kPa, 196.58 kPa, 237.00 kPa, 281.20 kPa, 271.05 kPa, 315.84 kPa, 376.38 kPa, 416.12 kPa, 418.17 kPa, 444.18 kPa, 506.96 kPa, 532.69 kPa, 569.10 kPa, 583.75 kPa, 628.84 kPa for these seventeen samples.

Polydimethyl siloxane (PDMS) is a commercially available silicone rubber. The molecular formula of PDMS is $(C_2H_6OSi)_n$. It is low cost, nontoxic, chemically resistant and stable against humidity and temperature variation. It is composed of two materials – Base agent (A) and Curing Agent (B). We used RTV 6136-D1 80LB KIT (R.S. Hughes, Baltimore, MD), a low viscosity silicone dielectric gel, to prepare PDMS. Two components A and B were mixed in a different mixing ratio by weight. As a curing agent (B) increases in A : B ratio mixture, the resulting PDMS samples become harder.

PDMS samples were made with a wide range of A : B ratios to satisfy the requirements of TIP breast phantom preparation. The mixed silicone material was poured into 37 mm \times 37 mm tall plastic molds. The height of the poured silicone in each mold was 19 mm. After curing, each PDMS cuboid was removed from molds and cut into four approximately identical pieces for replication in the experiment (Figure 4.3).

A compression experiment with PDMS samples was done using INSTRON 5944. For the test, we used a 1 kN load cell with a crosshead speed of 100 mm/min, and two 50 mm compression anvils attached to the system. A sample was placed on the middle of the bottom anvil for compression. The data capture frequency was set to 100 pts/s. The pre-compression load is required to be applied on a sample so that the compression plate touches the complete test material surface. We used a 0.2 N pre-compression load. The range of the compressive strain of each sample was set to the 5%-25% range.

The load was applied until the test material deforms up to 30% of its height before compression. From load cell signal force and vertical displacement were obtained. Using (2) and (3), stress and strain can be calculated. So for one mixture ratio, there are four test samples. Each sample was tested twice. Therefore, there are eight Young's moduli calculated using Instron's software (Bluhill 3) for each PDMS ratio. In this experiment, we considered 5-25% strain data for the calculation.

The Young's moduli of PDMSs found from the compression experiment are listed in the following tables. The linear region is observed from 5-25% strain, and linear curve fitting was performed using Instron measuring machine software (Bluehill 3).

Table D.1: PDMS Young's Modulus results

PDMS Mixing Ratio (Base : Curing)	Sample	Young's Modulus, kPa	Mean Young's Modulus, kPa	Standard deviation of the Mean, kPa
1 : 1.4	1	34.78	32.76	3.65
	2	35.07		
	3	36.49		
	4	37.17		
	5	31.30		
	6	30.88		
	7	29.58		
	8	26.82		
1 : 1.5	1	35.54	37.41	1.36
	2	36.75		
	3	36.95		
	4	35.93		
	5	38.19		
	6	37.66		
	7	38.99		
	8	39.28		
1 : 2	1	89.98	94.41	2.75
	2	90.62		
	3	94.30		
	4	96.83		
	5	94.91		
	6	94.76		
	7	97.12		
	8	96.74		
1 : 2.5	1	129.93	129.70	2.23
	2	127.29		
	3	132.19		
	4	127.91		
	5	129.03		
	6	127.01		
	7	132.30		
	8	131.94		
1 : 3	1	174.29	177.35	6.45
	2	173.69		
	3	170.82		
	4	169.65		
	5	178.57		
	6	186.76		
	7	179.06		
	8	185.93		
1 : 3.5	1	189.66	196.58	5.55
	2	199.47		
	3	191.81		
	4	191.80		
	5	198.04		
	6	207.04		
	7	196.89		
	8	197.95		

PDMS Mixing Ratio (Base : Curing)	Sample	Young's Modulus, kPa	Mean Young's Modulus, kPa	Standard deviation of the Mean, kPa
1 : 4	1	242.49	237.00	9.51
	2	243.39		
	3	230.58		
	4	231.11		
	5	247.81		
	6	248.45		
	7	226.11		
	8	226.04		
1 : 4.5	1	283.24	281.20	9.14
	2	281.78		
	3	266.28		
	4	276.05		
	5	292.99		
	6	293.73		
	7	275.34		
	8	280.17		
1 : 5	1	280.70	271.05	13.28
	2	291.38		
	3	268.18		
	4	281.12		
	5	253.03		
	6	253.98		
	7	269.53		
	8	270.49		
1 : 5.5	1	314.92	315.84	7.29
	2	317.42		
	3	321.20		
	4	320.01		
	5	327.63		
	6	312.32		
	7	306.26		
	8	306.94		
1 : 6	1	394.26	376.38	11.35
	2	393.53		
	3	369.43		
	4	371.12		
	5	372.12		
	6	376.46		
	7	363.96		
	8	370.07		
1 : 7	1	391.46	416.12	23.36
	2	380.78		
	3	404.25		
	4	406.08		
	5	435.78		
	6	437.35		
	7	431.94		
	8	441.31		

PDMS Mixing Ratio (Base : Curing)	Sample	Young's Modulus, kPa	Mean Young's Modulus, kPa	Standard deviation of the Mean, kPa
1 : 8	1	401.95	418.17	32.34
	2	402.52		
	3	440.45		
	4	441.23		
	5	452.78		
	6	452.42		
	7	373.57		
	8	380.45		
1 : 9	1	450.11	444.18	27.67
	2	450.23		
	3	406.44		
	4	408.11		
	5	439.56		
	6	439.63		
	7	478.68		
	8	480.66		
1 : 10	1	448.61	506.96	35.34
	2	451.83		
	3	530.65		
	4	531.62		
	5	526.90		
	6	526.82		
	7	516.93		
	8	522.34		
1 : 12.5	1	610.27	532.69	59.25
	2	613.02		
	3	487.58		
	4	493.39		
	5	471.76		
	6	472.19		
	7	557.68		
	8	555.60		
1 : 15	1	505.11	569.10	67.44
	2	506.09		
	3	600.78		
	4	600.04		
	5	513.49		
	6	512.98		
	7	651.29		
	8	663.00		
1 : 17.5	1	561.19	583.75	17.42
	2	612.73		
	3	587.50		
	4	587.78		
	5	576.36		
	6	560.27		
	7	592.86		
	8	591.33		

PDMS Mixing Ratio (Base : Curing)	Sample	Young's Modulus, kPa	Mean Young's Modulus, kPa	Standard deviation of the Mean, kPa
1 : 20	1	635.32	628.84	28.36
	2	638.65		
	3	616.75		
	4	623.53		
	5	584.63		
	6	601.69		
	7	666.34		
	8	663.78		

APPENDIX E

TPD CNN MODELS SOFTWARE IMPLEMENTATION

Depth Classification Model

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 22 11:17:26 2018
@author: tuc44192
## used linkes www.tensorflow.org/tutorials/keras/classification
"""
import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers.normalization import BatchNormalization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, Reshape, Lambda
from tensorflow.keras.layers import Conv2D, MaxPooling2D

from matplotlib import pyplot as plt
import os
import numpy as np

from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "TPD-CNN-Classification-2Classes_Depth1Added_Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.98)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle
##add train/validation datasets
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_train_1Depth_Added_Diss.tpd","rb")
X_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_train_1Depth_Added_Diss.tpd","rb")
```

```

y_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_test_1Depth_Added_Diss.tpd", "rb")
X_test = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_test_1Depth_Added_Diss.tpd", "rb")
y_test = pickle.load(pickle_in)
#
class_names = ['1_shallow', '2_deep']

print('Data structure')
print('Training')
print(X_train.shape)
print(len(y_train))
print('Validation')
print(X_test.shape)
print(len(y_test))

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(103,128))
plt.colorbar()
plt.grid(False)
plt.savefig('example_pic_Diss_Depth1Added_50_UPSAMPLE.png', dpi=1000)
plt.show()

## Normalization
#####
X_train = X_train / 255.0
X_test = X_test / 255.0
#####
## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn
# import the ADASYN object.
# pip install --upgrade scikit-learn

## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

# create the object to resample the majority class.
## create the object.

```

```

over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

nsamples_, nx_, ny_, nz_ = X_test.shape
d2_X_test = X_test.reshape((nsamples_,nx_*ny_*nz_))
print(d2_X_test.shape)

## fit the object to the training data.
x_train_over,y_train_over=over_sampler.fit_sample(d2_X_train,y_train)
x_test_over,y_test_over=over_sampler.fit_sample(d2_X_test, y_test)

##reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

nsamples2_, n_ = x_test_over.shape
X_test_over = x_test_over.reshape((nsamples2_,nx_,ny_,nz_))
print(X_test_over.shape)

# re-assignment
X_train = X_train_over
y_train = y_train_over
X_test = X_test_over
y_test = y_test_over

## Show examples of the dataset
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[i]])
plt.savefig('5by5_Diss_Depth1Added_50_UPSAMPLE.png',dpi = 300)
plt.show()

#####
# Model Development

```

```
model = Sequential()

# first layer
model.add(Conv2D(128, (11,11), input_shape=X_train.shape[1:]))
model.add(Activation('elu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
# second layer
model.add(Conv2D(128, (9, 9)))
model.add(Activation('softmax'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (5,5)))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(2)) #amount of classes
model.add(Activation('softmax'))
#model.add(BatchNormalization())

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adamax',
              metrics=['accuracy'])
```

```

model.summary()

#####
history = model.fit(X_train, y_train, batch_size=200, epochs=50,
validation_data = (X_test, y_test), verbose=1)
['accuracy', 'loss', 'val_accuracy', 'val_loss']

test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)
print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='lower right')
plt.savefig("Accuracy_plot_CNNModel_Depth1Added_Diss_
Model_02042021_50_UPSAMPLE.png",dpi = 300);
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='upper right')
plt.savefig("Loss_plot_CNNModel_Depth1Added_Diss_
Model_02042021_50_UPSAMPLE.png",dpi = 300);
plt.show()

##### SAVE
#save whole model
model.save('myCNN_Depth1Added_Diss_Model5_02072021_50_UPSAMPLE.h5')
#####

### Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.

```

```

layers.Softmax())
predictions = probability_model.predict(X_test)

#def plot_image(i, predictions_array, true_label):
def plot_image(i, predictions_array, true_label, img):
    predictions_array, true_label, img = predictions_array, true_label[i],
    img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img.reshape(103,128), cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_array)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'red'

    plt.xlabel("{} {:.2f}% ({}).format(class_names[predicted_label],
                                     100*np.max(predictions_array),
                                     class_names[true_label]),
               color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(2))
    plt.yticks([])
    thisplot = plt.bar(range(2), predictions_array, color="#777777")
    plt.ylim([0, 1])
    predicted_label = np.argmax(predictions_array)

    thisplot[predicted_label].set_color('red')
    thisplot[true_label].set_color('blue')

# show multiple images with predictions
num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(i, predictions[i], y_test, X_train)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)

```

```

plt.subplot(num_rows, 2*num_cols, 2*i+2)
plot_value_array(i, predictions[i], y_test)
plt.tight_layout()
plt.savefig('Prediictions_gridDiss_
Depth1Added_Model15_50_UPSAMPLE.png',dpi = 250)
plt.show()

# Test on images from the set-aside dataset
## add test samples (paper set)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/Test_EMBCpaper.tpd","rb")
X_test_paper = pickle.load(pickle_in)
X_test_paper = X_test_paper / 255.0

### Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions_paper = probability_model.predict(X_test_paper)
#
print(predictions_paper)

```

Size Classification Model

```

# -*- coding: utf-8 -*-
"""
Created on Mon Oct 22 11:17:26 2018
@author: tuc44192
## used linkes www.tensorflow.org/tutorials/keras/classification
"""
import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()
#from tensorflow.keras.datasets import cifar10
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers.normalization import BatchNormalization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, Reshape, Lambda
from tensorflow.keras.layers import Conv2D, MaxPooling2D

from matplotlib import pyplot as plt
import os
import numpy as np

```

```

from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "TPD-CNN-Classification-3Classes_Size_Added_Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.99)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle
##add train/validation samples
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_train_1Size_Added_Diss.tpd", "rb")
X_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_train_1Size_Added_Diss.tpd", "rb")
y_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_test_1Size_Added_Diss.tpd", "rb")
X_test = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_test_1Size_Added_Diss.tpd", "rb")
y_test = pickle.load(pickle_in)

class_names = ['1_small', '2_medium', '3_large']

print('Data structure')
print('Training')
print(X_train.shape)
print(len(y_train))
print('Validation')
print(X_test.shape)
print(len(y_test))

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(103,128))
plt.colorbar()
plt.grid(False)
plt.savefig('example_pic_Diss_Size1_Added5_50_Upsample.png',dpi=1000)
plt.show()

```

```

## Normalization
#####
X_train = X_train / 255.0
X_test = X_test / 255.0
#####

## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn
# pip install --upgrade scikit-learn

## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

# create the object to resample the majority class.
over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

nsamples_, nx_, ny_, nz_ = X_test.shape
d2_X_test = X_test.reshape((nsamples_,nx_*ny_*nz_))
print(d2_X_test.shape)

## fit the object to the training data.
x_train_over,y_train_over=over_sampler.fit_sample(d2_X_train,y_train)
x_test_over y_test_over=over_sampler.fit_sample(d2_X_test,y_test)

#reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

nsamples2_, n_ = x_test_over.shape
X_test_over = x_test_over.reshape((nsamples2_,nx_,ny_,nz_))
print(X_test_over.shape)

# re-assignment
X_train = X_train_over
y_train = y_train_over

```

```

X_test = X_test_over
y_test = y_test_over

## Show examples of the dataset
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[i]])
plt.savefig('5by5_Diss_Size1_Added5_50_Upsample.png',dpi = 300)
plt.show()

# Model Development
model = Sequential()

# first layer
model.add(Conv2D(128, (11, 11), input_shape=X_train.shape[1:]))
model.add(Activation('elu'))
model.add(MaxPooling2D(pool_size=(3, 3)))
# second layer
model.add(Conv2D(128, (9, 9)))
model.add(Activation('softmax'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
# Third layer
model.add(Conv2D(256, (5, 5)))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

```

```

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(3)) #amount of classes
model.add(Activation('softmax'))
#model.add(BatchNormalization())

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adamax',
              metrics=['accuracy'])

model.summary()

#####
history = model.fit(X_train, y_train, batch_size=200, epochs=50,
                  validation_data = (X_test, y_test), verbose=1)
['accuracy', 'loss', 'val_accuracy', 'val_loss']

test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)
print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='lower right')
plt.savefig("Accuracy_plot_CNNModel_Size1_
Added_Diss5_50_Upsample.png",dpi = 300);
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')

```

```

plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='upper right')
plt.savefig("Loss_plot_CNNModel_Size1_
Added_Diss5_50_Upsample.png",dpi = 300);
plt.show()

## Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions = probability_model.predict(X_test)

#####save a whole model

model.save('myCNN_Size1Added_Diss_Modelv5__50_02072021_Upsample.h5')
#####

# show predictionas and labels
# Plot the first X test images, their predicted labels,
and the true labels.
# Color correct predictions in blue and incorrect predictions in red.
#def plot_image(i, predictions_array, true_label):
def plot_image(i, predictions_array, true_label, img):
    predictions_array, true_label, img = predictions_array,
    true_label[i], img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img.reshape(103,128), cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_array)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'red'

    plt.xlabel("{} {:.2f}% ({})" .format(class_names[predicted_label],
                                        100*np.max(predictions_array),
                                        class_names[true_label]),
                color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(3))

```

```

plt.yticks([])
thisplot = plt.bar(range(3), predictions_array, color="#777777")
plt.ylim([0, 2])
predicted_label = np.argmax(predictions_array)

thisplot[predicted_label].set_color('red')
thisplot[true_label].set_color('blue')

#multiple images with predictions
num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(i, predictions[i], y_test, X_train)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(i, predictions[i], y_test)
plt.tight_layout()
plt.savefig('Prediictions_gridDiss_Size1Added_50_Upsample.png',dpi=250)
plt.show()

#test on images from other datasets
##add test samples (paper set)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/Test_EMBCpaper.tpd","rb")
X_test_paper = pickle.load(pickle_in)
X_test_paper =X_test_paper / 255.0

### Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions_paper = probability_model.predict(X_test_paper)
#
print(predictions_paper)

```

Stiffness Classification Model

```

# -*- coding: utf-8 -*-
"""
Created on Mon Oct 22 11:17:26 2018
@author: tuc44192
## used linkes www.tensorflow.org/tutorials/keras/classification

```

```

"""
import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers.normalization import BatchNormalization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, Reshape, Lambda
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from matplotlib import pyplot as plt
import os
import numpy as np
from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "TPD-CNN-Classification-2Classes_Stiffness1Added_Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.99)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle
##add train/validation samples
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_train_1Stiffness_Added_Diss.tpd","rb")
X_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_train_1Stiffness_Added_Diss.tpd","rb")
y_train = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
X_test_1Stiffness_Added_Diss.tpd","rb")
X_test = pickle.load(pickle_in)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/
y_test_1Stiffness_Added_Diss.tpd","rb")
y_test = pickle.load(pickle_in)
#
class_names = ['1_soft', '2_stiff']

print('Data structure')
print('Training')
print(X_train.shape)

```

```

print(len(y_train))
print('Validation')
print(X_test.shape)
print(len(y_test))

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(103,128))
plt.colorbar()
plt.grid(False)
plt.savefig('example_pic_Diss_Stiffness1_
Added_UPRAND_v7_25.png',dpi = 1000)
plt.show()

## Normalization
#####
X_train = X_train / 255.0
X_test = X_test / 255.0
#####

*****Upsampling Random
## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn
# pip install --upgrade scikit-learn

## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

# create the object to resample the majority class.
## create the object.
over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

nsamples_, nx_, ny_, nz_ = X_test.shape
d2_X_test = X_test.reshape((nsamples_,nx_*ny_*nz_))
print(d2_X_test.shape)

## fit the object to the training data.

```

```

x_train_over,y_train_over=over_sampler.fit_sample(d2_X_train,y_train)
x_test_over,y_test_over=over_sampler.fit_sample(d2_X_test,y_test)

#reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

nsamples2_, n_ = x_test_over.shape
X_test_over = x_test_over.reshape((nsamples2_,nx_,ny_,nz_))
print(X_test_over.shape)

# re-assignment
X_train = X_train_over
y_train = y_train_over

X_test = X_test_over
y_test = y_test_over

## Show examples of the dataset
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[i]])
plt.savefig('5by5_Diss_Stiffness3_Added_UPRAND_v7_25.png',dpi = 300)
plt.show()

# Model Development
model = Sequential()

# first layer
model.add(Conv2D(128, (11, 11), input_shape=X_train.shape[1:]))
model.add(Activation('elu'))
model.add(MaxPooling2D(pool_size=(3, 3)))
# second layer
model.add(Conv2D(128, (9, 9)))
model.add(Activation('softmax'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
# Third layer
model.add(Conv2D(256, (5, 5)))

```

```

model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(128))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(2)) #amount of classes
model.add(Activation('softmax'))
#model.add(BatchNormalization())

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adamax',
              metrics=['accuracy'])

model.summary()

#####
history = model.fit(X_train, y_train, batch_size=200, epochs=25,
                  validation_data = (X_test, y_test), verbose=1)
['accuracy', 'loss', 'val_accuracy', 'val_loss']

test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)
print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

```

```

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='lower right')
plt.savefig("Accuracy_plot_CNNModel_Stiffness1_
Added_UPRAND_v7_25.png",dpi = 300);
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='upper right')
plt.savefig("Loss_plot_CNNModel_Stiffness1_
Added_UPRAND_v7_25.png",dpi = 300);
plt.show()

### Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions = probability_model.predict(X_test)

##### SAVE
#save a whole model
model.save('myCNN_Stiffness1Added__Added__UPRAND_25_Diss_
Modelv7_02052021.h5')

#####

# show predictionas and labels
# Plot the first X test images, their predicted labels, and the true labels.
# Color correct predictions in blue and incorrect predictions in red.
#def plot_image(i, predictions_array, true_label):
def plot_image(i, predictions_array, true_label, img):
    predictions_array, true_label, img = predictions_array, true_label[i],
    img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

```

```

plt.imshow(img.reshape(103,128), cmap=plt.cm.binary)

predicted_label = np.argmax(predictions_array)
if predicted_label == true_label:
    color = 'blue'
else:
    color = 'red'

plt.xlabel("{} {:.20f}% ({}).format(class_names[predicted_label],
                                   100*np.max(predictions_array),
                                   class_names[true_label]),
          color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(2))
    plt.yticks([])
    thisplot = plt.bar(range(2), predictions_array, color="#777777")
    plt.ylim([0, 1])
    predicted_label = np.argmax(predictions_array)

    thisplot[predicted_label].set_color('red')
    thisplot[true_label].set_color('blue')

#multiple images with predictions
num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(i, predictions[i], y_test, X_train)
    plt.imshow(X_train[i].reshape(103,128), cmap=plt.cm.binary)
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(i, predictions[i], y_test)
plt.tight_layout()
plt.savefig('Prediictions_gridDiss_Stiffness1_
Added_APRAND_v7_25.png',dpi = 250)
plt.show()

##add test samples (paper set)
pickle_in = open("F:/Vira/PhantomTest for CNN/Dissertation Data Set/
Dissertation_Data/PICKLES with added data/Test_EMBCpaper.tpd","rb")
X_test_paper = pickle.load(pickle_in)

```

```
X_test_paper =X_test_paper / 255.0

### Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions_paper = probability_model.predict(X_test_paper)
#
print(predictions_paper)
```

APPENDIX F

MPD CNN MODELS SOFTWARE IMPLEMENTATION

Asymmetry Classification Model

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 22 11:17:26 2018
@author: tuc44192
"""
import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers.normalization import BatchNormalization
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, Reshape, Lambda
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from matplotlib import pyplot as plt
import os
import numpy as np
from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "MPD-CNN-Classification-2Classes_AsymmetryBalanced_Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.99)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle
##add train/valid/test samples

pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/xMPDdiff_Asymmetry_Training.tpd","rb")
X_train = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/yMPDdiff_Asymmetry_Training.tpd","rb")
y_train = pickle.load(pickle_in)

pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/xMPDdiff_Asymmetry_Validation.tpd","rb")
X_val = pickle.load(pickle_in)
```

```

pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/yMPDdiff_Asymmetry_Validation.tpd","rb")
y_val = pickle.load(pickle_in)

pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/xMPDdiff_Asymmetry_Test.tpd","rb")
X_test = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/Python/
Multispectral Models/yMPDdiff_Asymmetry_Test.tpd","rb")
y_test = pickle.load(pickle_in)
#
class_names = ['1_Symmetric', '2_Asymmetric']

print('Data structure')

print('Training')
print(X_train.shape)
print(len(y_train))

print('Validation')
print(X_val.shape)
print(len(y_val))

print('Test')
print(X_test.shape)
print(len(y_test))

nk = X_train.shape[0]
nl = X_test.shape[0]

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(171,171))
plt.colorbar()
plt.grid(False)
plt.savefig('Example_pic_Diss_MPDAsymmetry.png',dpi = 1000)
plt.show()

## Normalization
#####
X_train = X_train / 255.0
X_val    = X_val   / 255.0
X_test   = X_test  / 255.0
#####
*****Upsampling Random

```

```

## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn
# pip install --upgrade scikit-learn
## - - - - -
## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

# create the object to resample the majority class.
over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

## fit the object to the training data.
x_train_over,y_train_over=over_sampler.fit_sample(d2_X_train,y_train)

#reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

#reassignment
X_train = X_train_over
y_train = y_train_over

## Show examples of the dataset
nk = X_train.shape[0]
*****Upsampling Random upsampling
## Show examples of the dataset
from numpy import random
#random.randint(153504, size=(25)): #
k= random.randint(nk, size=(25))
plt.figure(figsize=(10,10))
for i in range(25): # random.randint(153504, size=(25)): #
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[k[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[k[i]]])
plt.savefig('5by5_Diss__MPDAsymmetry.png',dpi = 300)

```

```
plt.show()

# Model Development
model = Sequential()

# first layer
model.add(Conv2D(32, (3, 3), input_shape=X_train.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
# second layer
model.add(Conv2D(16, (5, 5)))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
# second layer
model.add(Conv2D(8, (9, 9)))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.3))

model.add(Dense(8))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(16))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(32))
model.add(Activation('relu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

model.add(Dense(2)) #amount of classes
model.add(Activation('softmax'))
#model.add(BatchNormalization())

model.compile(loss='sparse_categorical_crossentropy',
optimizer='adamax',
```

```

        metrics=['accuracy'])

model.summary()

# Implementing Callback
from tensorflow.keras.callbacks import ModelCheckpoint
filepath = "SavedAsymmetry-{epoch:02d}.h5"

checkpoint = ModelCheckpoint(filepath, monitor='accuracy', verbose=0,
save_best_only=False, mode='auto', save_weights_only=False,
save_freq = 1)

history = model.fit(X_train, y_train, batch_size=100, epochs=5, v
alidation_data = (X_val, y_val), verbose=1, callbacks=[checkpoint])
['accuracy', 'loss', 'val_accuracy', 'val_loss']

# Validate also on the separate test data
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)
print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='lower right')
plt.savefig("Accuracy_plot_Diss_MPDAsymmetry.png",dpi = 300);
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='upper right')
plt.savefig("Loss_plot_Diss_MPDAsymmetry.png",dpi = 300);
plt.show()

## Make predictions

```

```

probability_model = tf.keras.Sequential([model,tf.keras.
layers.Softmax()])
predictions = probability_model.predict(X_test)

##### SAVE
#save a whole model
model.save('Model_Diss_MPDAsymmetry_04092021.h5')
#####

# show predictionas and labels
# Plot the first X test images, their predicted labels, and the true
  labels.
# Color correct predictions in blue and incorrect predictions in red.
#def plot_image(i, predictions_array, true_label):
def plot_image(i, predictions_array, true_label, img):
    predictions_array,true_label,img =predictions_array,
    true_label[i], img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img.reshape(171,171), cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_array)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'red'

    plt.xlabel("{} {:.2f}% ({})."format(class_names[predicted_label],
                                        100*np.max(predictions_array),
                                        class_names[true_label]),
              color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(2))
    plt.yticks([])
    thisplot = plt.bar(range(2), predictions_array, color="#777777")
    plt.ylim([0, 1])
    predicted_label = np.argmax(predictions_array)

    thisplot[predicted_label].set_color('red')
    thisplot[true_label].set_color('blue')

```

```

num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
l = random.randint(n1, size=(num_images))
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(l[i], predictions[l[i]], y_test, X_test)
    plt.imshow(X_test[l[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(l[i], predictions[l[i]], y_test)
plt.tight_layout()
plt.savefig('Prediictions_grid_Diss_MPDAsymmetry.png',dpi = 250)
plt.show()

```

Texture Classification Model

```

# -*- coding: utf-8 -*-
"""
Created in 2020
@author: tuc44192
"""

import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten,Reshape, Lambda, BatchNormalization
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from matplotlib import pyplot as plt
import os
import numpy as np

from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "MPD-CNN-Classification-2Classes-TextureModel-Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.99)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle

```

```
##add train/val/test samples

# Training 0.7
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Texture_Training.tpd","rb")
X_train = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Texture_Training.tpd","rb")
y_train = pickle.load(pickle_in)

# Validation 0.2
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Texture_Validation.tpd","rb")
X_val = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Texture_Validation.tpd","rb")
y_val = pickle.load(pickle_in)

# Test 0.1
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Texture_Test.tpd","rb")
X_test = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Texture_Test.tpd","rb")
y_test = pickle.load(pickle_in)

class_names = ["1_NotPitted", "2_Pitted"]

print('Data structure')

print('Training')
print(X_train.shape)
print(len(y_train))

print('Validation')
print(X_val.shape)
print(len(y_val))

print('Test')
print(X_test.shape)
print(len(y_test))

nk = X_train.shape[0]
nl = X_test.shape[0]
```

```

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(171,171))
plt.colorbar()
plt.grid(False)
plt.savefig('Example_pic_Diss_MPDTexture.png',dpi = 1000)
plt.show()

## Normalization
#####
X_train = X_train / 255.0
X_val   = X_val   / 255.0
X_test  = X_test  / 255.0
#####
*****Upsampling Random
## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn
# pip install --upgrade scikit-learn
## - - - - -
## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

# create the object to resample the majority class.
over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

## fit the object to the training data.
x_train_over,y_train_over=over_sampler.fit_sample(d2_X_train,y_train)

#reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

# re-assignment
X_train = X_train_over
y_train = y_train_over

## Show examples of the dataset

```

```

from numpy import random
k= random.randint(nk, size=(25))
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[k[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[k[i]]])
plt.savefig('5by5_Diss_MPDTexture.png',dpi = 300)
plt.show()

# Model Development
model = Sequential()

# first layer
model.add(Conv2D(8, (3, 3), input_shape=X_train.shape[1:]))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# second layer
model.add(Conv2D(8, (3, 3)))
model.add(Activation('elu'))
model.add(Conv2D(8, (3, 3)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(3, 3)))

# third layer
model.add(Conv2D(8, (5, 5)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(3, 3)))

model.add(Flatten())
model.add(Dropout(0.2))

model.add(Dense(256))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.4))

model.add(Dense(128))

```

```

model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.4))
#
model.add(Dense(64))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.4))

model.add(Dense(2)) #amount of classes
model.add(Activation('softmax'))

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

model.summary()

# Implementing Callback
from tensorflow.keras.callbacks import ModelCheckpoint
filepath = "SavedModel_Texture-{epoch:02d}f3.h5"

checkpoint = ModelCheckpoint(filepath, monitor='accuracy', verbose=0,
                             save_best_only=False, mode='auto', save_weights_only=False, save_freq = 1)

#####

history = model.fit(X_train, y_train, batch_size=100, epochs=20,
                   shuffle=True,
                   validation_data = (X_val, y_val), verbose=1, callbacks=[checkpoint])
['accuracy', 'loss', 'val_accuracy', 'val_loss']

# Validate also on the separate test data
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)
print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')

```



```

        class_names[true_label]),
        color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(2))
    plt.yticks([])
    thisplot = plt.bar(range(2), predictions_array, color="#777777")
    plt.ylim([0, 1])
    predicted_label = np.argmax(predictions_array)

    thisplot[predicted_label].set_color('red')
    thisplot[true_label].set_color('blue')

num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
l = random.randint(n1, size=(num_images))
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(l[i], predictions[l[i]], y_test, X_test)
    plt.imshow(X_test[l[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(l[i], predictions[l[i]], y_test)
plt.tight_layout()
plt.savefig('Prediictions_grid_Diss_MPDTexturedf3.png',dpi = 250)
plt.show()

```

Inflammation Classification Model

```

# -*- coding: utf-8 -*-
"""
Created in 2021
@author: tuc44192
"""
import tensorflow as tf
tf.keras.backend.clear_session()
tf.reset_default_graph()
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, Reshape, Lambda, BatchNormalization

```

```

from tensorflow.keras.layers import Conv2D, MaxPooling2D,
    ActivityRegularization
#Activity Regularizer: Tries to reduce the layer's output y,
    thus will reduce the weights and adjust bias so  $Wx+b$  is smallest.
from matplotlib import pyplot as plt
import os
import numpy as np
from tensorflow.keras.callbacks import TensorBoard
import time

NAME = "MPD-CNN-
Classification-2Classes_50,75_Large_Inflammation_Dissertation"

gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.99)
sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

import pickle
##add train/validation/test sets

# Training
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Inflammation_Training.tpd", "rb")
X_train = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Inflammation_Training.tpd", "rb")
y_train = pickle.load(pickle_in)

# Validation 0.2
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Inflammation_Validation.tpd", "rb")
X_val = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Inflammation_Validation.tpd", "rb")
y_val = pickle.load(pickle_in)

# Test
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/xMPDdiff_Inflammation_Test.tpd", "rb")
X_test = pickle.load(pickle_in)
pickle_in = open("C:/Users/CSNAP/Documents/Vira/CNN Data Sets/Code/
Python/Multispectral Models/yMPDdiff_Inflammation_Test.tpd", "rb")
y_test = pickle.load(pickle_in)
#
class_names = ['1_LessInflammation', '2_MoreInflammation']

```

```

print('Data structure')

print('Training')
print(X_train.shape)
print(len(y_train))

print('Validation')
print(X_val.shape)
print(len(y_val))

print('Test')
print(X_test.shape)
print(len(y_test))

nk = X_train.shape[0]
nl = X_test.shape[0]

## Data pre-processing
plt.figure()
plt.imshow(X_train[0].reshape(171,171))
plt.colorbar()
plt.grid(False)
plt.savefig('Example_pic_Diss_MPDInflammation.png',dpi=1000)
plt.show()

## Normalization
X_train = X_train.astype('float32')
X_val    = X_val.astype('float32')
X_test   = X_test.astype('float32')
#####
X_train = X_train / 255.0
X_val   = X_val   / 255.0
X_test  = X_test  / 255.0
#####

*****Upsampling Random
## using pip
#pip install -U imbalanced-learn
## using conda
# conda install -c conda-forge imbalanced-learn

# pip install --upgrade scikit-learn

## import the Random Over Sampler object.
from imblearn.over_sampling import RandomOverSampler

```

```

# create the object to resample the majority class.
## create the object.
over_sampler = RandomOverSampler()

# reshape from 4d to 2d
nsamples, nx, ny, nz = X_train.shape
d2_X_train = X_train.reshape((nsamples,nx*ny*nz))
print(d2_X_train.shape)

## fit the object to the training data.
x_train_over, y_train_over=over_sampler.fit_sample(d2_X_train,y_train)

#reshape back to 4d
nsamples2, n = x_train_over.shape
X_train_over = x_train_over.reshape((nsamples2,nx,ny,nz))
print(X_train_over.shape)

# reassignment
X_train = X_train_over
y_train = y_train_over

nk = X_train.shape[0]

*****Upsampling
## Show examples of the dataset
from numpy import random
k= random.randint(nk, size=(25))
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[k[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.xlabel(class_names[y_train[k[i]]])
plt.savefig('5by5_Diss__MPDInflammation.png',dpi = 300)
plt.show()

# Model Development

model = Sequential()
# First layer
model.add(Conv2D(8, (3, 3), input_shape=X_train.shape[1:]))
#model.add(ActivityRegularization(l1=0, l2=0.001))

```

```
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(3, 3)))

# Second Layer
model.add(Conv2D(8, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Conv2D(8, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Third Layer
model.add(Conv2D(8, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Conv2D(8, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Conv2D(8, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

# Fourth Layer
model.add(Conv2D(16, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Conv2D(16, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Conv2D(16, (3, 3)))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```

# Flatten
model.add(Flatten())
model.add(Dropout(0.2))

## Dense 1
model.add(Dense(256))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
#model.add(BatchNormalization())
model.add(Dropout(0.2))

# Dense 2
model.add(Dense(64))
#model.add(ActivityRegularization(l1=0, l2=0.001))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Dense Classification
model.add(Dense(2)) #amount of classes
#model.add(ActivityRegularization(l1=0, l2=0.01))
model.add(Activation('softmax'))
#model.add(BatchNormalization())
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adamax',
              metrics=['accuracy'])

model.summary()

# Implementing Callback
from tensorflow.keras.callbacks import ModelCheckpoint
filepath = "Saved_MPDInflammation_Diss6-{epoch:02d}.h5"

checkpoint = ModelCheckpoint(filepath, monitor='accuracy', verbose=0,
save_best_only=False, mode='auto', save_weights_only=False,
save_freq = 1)
history = model.fit(X_train, y_train, batch_size=50, epochs=20,
                    shuffle=True,
                    validation_data = (X_val, y_val), verbose=1, callbacks=[checkpoint])
['accuracy', 'loss', 'val_accuracy', 'val_loss']

# Validate also on the separate test data
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=1)
print('\nTest accuracy', test_acc)

```

```

print('Test loss \n', test_loss)

# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Validation set'], loc='lower right')
plt.savefig("Accuracy_plot_Diss_MPDInflammation6.png",dpi = 300);
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Validation set'], loc='upper right')
plt.savefig("Loss_plot_Diss_MPDInflammation6.png",dpi = 300);
plt.show()

## Make predictions
probability_model = tf.keras.Sequential([model,tf.keras.layers.Softmax()])
predictions = probability_model.predict(X_test)

##### SAVE
#save a whole model
model.save('Model_Diss_MPDInflammation6_04122021.h5')
#####

# show predictionas and labels
# Plot the first X test images, their predicted labels,
and the true labels.
# Color correct predictions in blue and incorrect predictions in red.
#def plot_image(i, predictions_array, true_label):
def plot_image(i, predictions_array, true_label, img):
    predictions_array, true_label, img = predictions_array,
    true_label[i], img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

```

```

plt.imshow(img.reshape(171,171), cmap=plt.cm.binary)

predicted_label = np.argmax(predictions_array)
if predicted_label == true_label:
    color = 'blue'
else:
    color = 'red'

plt.xlabel("{} {:.2f}% ({})" .format(class_names[predicted_label],
                                     100*np.max(predictions_array),
                                     class_names[true_label]),
          color=color)

def plot_value_array(i, predictions_array, true_label):
    predictions_array, true_label = predictions_array, true_label[i]
    plt.grid(False)
    plt.xticks(range(2))
    plt.yticks([])
    thisplot = plt.bar(range(2), predictions_array, color="#777777")
    plt.ylim([0, 1])
    predicted_label = np.argmax(predictions_array)

    thisplot[predicted_label].set_color('red')
    thisplot[true_label].set_color('blue')

num_rows = 5
num_cols = 3
num_images = num_rows*num_cols
l = random.randint(n1, size=(num_images))
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(l[i], predictions[l[i]], y_test, X_test)
    plt.imshow(X_test[l[i]].reshape(171,171), cmap=plt.cm.binary)
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(l[i], predictions[l[i]], y_test)
plt.tight_layout()
plt.savefig('Prediictions_grid_Diss_MPDInflammation6.png',dpi = 250)
plt.show()

```