

Graduate Theses, Dissertations, and Problem Reports

2016

# Imaging White Blood Cells using a Snapshot Hyper-Spectral Imaging System

Christopher John Robison

Follow this and additional works at: https://researchrepository.wvu.edu/etd

# **Recommended Citation**

Robison, Christopher John, "Imaging White Blood Cells using a Snapshot Hyper-Spectral Imaging System" (2016). *Graduate Theses, Dissertations, and Problem Reports*. 6519. https://researchrepository.wvu.edu/etd/6519

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Imaging White Blood Cells using a Snapshot Hyper-Spectral Imaging System Christopher John Robison

Thesis submitted

to the Benjamin M. Statler College of Engineering and Mineral Resources

at West Virginia University

In partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Jeremy M. Dawson, Ph.D., Chair

Thirimachos Bourlai, Ph.D.

Christopher J. Kolanko, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2016

Keywords: Hyper-Spectral Imaging, Medical Imaging,

**Copyright 2016 Christopher John Robison** 

#### Abstract

# Imaging White Blood Cells using a Snapshot Hyper-Spectral Imaging System by Christopher Robison

Automated white blood cell (WBC) counting systems process an extracted whole blood sample and provide a cell count. A step that would not be ideal for onsite screening of individuals in triage or at a security gate. Snapshot Hyper-Spectral imaging systems are capable of capturing several spectral bands simultaneously, offering co-registered images of a target. With appropriate optics, these systems are potentially able to image blood cells in vivo as they flow through a vessel, eliminating the need for a blood draw and sample staining. Our group has evaluated the capability of a commercial Snapshot Hyper-Spectral imaging system, specifically the Arrow system from Rebellion Photonics, in differentiating between white and red blood cells on unstained and sealed blood smear slides. We evaluated the imaging capabilities of this hyperspectral camera as a platform to build an automated blood cell counting system. Hyperspectral data consisting of 25, 443x313 hyperspectral bands with ~3nm spacing were captured over the range of 419 to 494nm. Open-source hyperspectral datacube analysis tools, used primarily in Geographic Information Systems (GIS) applications, indicate that white blood cells' features are most prominent in the 428-442nm band for blood samples viewed under 20x and 50x magnification over a varying range of illumination intensities. The system has shown to successfully segment blood cells based on their spectral-spatial information. These images could potentially be used in subsequent automated white blood cell segmentation and counting algorithms for performing in vivo white blood cell counting.

# **DEDICATION**

I would like to dedicate this thesis work to my advisory committee members: Dr. Jeremy Dawson, Dr. Thirimachos Bourlai, and Dr. Christopher Kolanko. Without your expertise and guidance, none of this work would be possible. I would like to thank James Robison, Janet Robison, Curtis Robison, and Jessica Robison for their constant encouragement as I don't know where I would be without their support. I would also like to thank my friends and colleagues who have lent a helping hand at many points during the course of my graduate career. I would notably like to thank the following:

Kyle Smith, Anand Kadiyala, Matthew Pachol, Cameron Whitelam, Amanda Holbert, James Searls, Michael Martin, Seth Leffel, Brianna Maze, Jordan Martin, Curtis Sheets III, Ronnie Sanford, Stallone Sabatier, Morgan Trester, Jacob Tyo, the CyberWVU team, the LCSEE Nanophotonics group, and LCSEE faculty members for their hard work keeping this department a well-oiled machine.

# ACKNOWLEDGMENTS

The author wishes to acknowledge Rebellion Photonics, WVU LCSEE Nanophotonics, and West Virginia University for their contributions to this work.

# **Table of Contents**

CHAPTER 1: INTRODUCTION	1
1.1 PROBLEM STATEMENT:	2
1.2 MANUAL BLOOD CELL COUNTING: BACKGROUND AND LIMITATIONS	8
1.2.1 BLOOD COUNT LIMITATIONS	.10
1.3 BLOOD COUNT AUTOMATION	11
1.3.1 RAPID BLOOD COUNTING SYSTEMS	.12
1.3.2 PHOTOTHERMAL TOMOGRAPHY METHODS OF CELL DIFFERENTIATION	.13
1.3.3 OTHER SYSTEMS	.13
1.4 REMOTE SENSING IMPACTS ON BLOOD CELL COUNTING	14
1.4.1 Hyperspectral Imagery	.15
1.4.2 BLOOD TISSUE OPTICAL PROPERTIES	.17
1.5 SIMILAR APPLICATIONS	17
1.5.1 BLOOD CELL DETECTION AND CLASSIFICATION USING AN RGB IMAGE PROCESSING	1,
APPROACH	17
1.5.1.1 A Framework for WBC Segmentation in Microscopic Blood Images using Digital Imag	. 17 је
Processing	18
1.5.1.2 A New White Blood Call Segmentation using Mean Shift Filter and Region Growing	. 10
Algorithm	19
1.5.1.2 White Blood Call Segmentation by Color Space Based K Means Clustering	.10
1.5.1.5 While Blood Cell Seglicination by Color-Space-Dased K-Means Clustering	. 19
1.5.1.4 Visual Saliency Attention for Localization and Segmentation in Stanled Leukocyte	10
	. 19
1.5.1.5 A Summary of KGB Blood Cell Detection Methods.	. 19
1.5.2 BLOOD CELL DETECTION AND CLASSIFICATION USING HYPERSPECTRAL IMAGING SYSTE	MS
	.20
1.5.2.1 Blood Cell Classification Using a Hyperspectral Imaging Technique	.20
1.5.2.2 Leukocyte identification based on molecular HS imaging	.22
1.5.2.3 Hyperspectral Image Classification via contextual deep learning	.23
<b>1.6 System Proposal</b>	25
1.6.1 System Operation	.26
1.6.2 System Design	.27
1.6.3 System Impact:	.28
1.7 THESIS CONTRIBUTION	30
1.8 THESIS ORGANIZATION	33
CHAPTER 2: THEORY	.34
2.1 BLOOD CELLS & THE HUMAN IMMUNE SYSTEM	35
2.1.1 Optical Properties of Leukocytes	.37
2.2 ENDMEMBER STORAGE DATABASE	39
2.2.1 HIERARCHICAL DATA FORMAT	.40
2.3 BLOOD CELL SEGMENTATION TECHNIQUES	40
2.4 Hyperspectral Imaging Systems:	41
2.4.1 LINE SCAN HYPERSPECTRAL IMAGING SYSTEMS	41
2.4.2. SPECTRAL SCANNING HYPERSPECTRAL IMAGING SYSTEMS	43
2.4.3 SNAPSHOT HYPERSPECTRAL IMAGING SYSTEMS	44
2.4.4 DATACHEF	47
2.5.5 SPECTRAL MIXING	. – / 48
2.5 STECTIVE MILLING	. <del>4</del> 0
2.0 IIII ERGE EUIKAL LUAGE I KEIKUULGOING 2.6.1 East Eolided Toanseodm Filteding	<b>47</b> 50
2.0.1 LAST FOURIER TRANSFORM FILTERING	. 50
2.0.2 ENDIVIEWIBER CONSTRUCTION	1

2.7 FEATURE EXTRACTION TECHNIQUES	52
2.7.1 SPECTRAL ANGLE MAPPER	
2.7.2 PRINCIPAL COMPONENT ANALYSIS	53
2.8 CLASSIFIERS	55
2.8.1 NAÏVE BAYES CLASSIFIER	55
2.8.2 GAUSSIAN MAXIMUM LIKELIHOOD CLASSIFIER	55
2.8.3 SUPPORT VECTOR MACHINE	57
2.9 RESEARCH TOOLS	58
2.9.1 REBELLION PHOTONICS CRYSTAL	59
2.9.2 PYTHON PACKAGES	59
2.9.2.1 Numerical Python	59
2.9.2.2 Scientific Python	60
2.9.2.3 SciKit	60
2.9.2.4 Spectral Python	60
2.9.3 JETBRAINS PYCHARM	61
CHAPTER 3: IMAGING SYSTEM CONFIGURATION AND DATA CAPTURE	62
3.1 RESEARCH OBJECTIVE	63
3.2 DATA CAPTURE AND PROCESSING	65
3.2.1 HSI System Setup:	67
3.2.1.1 Alternative System Considerations and Justifications	69
3.2.2 HSI CALIBRATION AND TARGET SIZE ESTIMATION	
3.2.3 BLOOD SAMPLE PREPARATION	72
3.2.4 BLOOD SAMPLE HYPERSPECTRAL DATA CAPTURE:	74
3.2.5 DATASET CONSTRUCTION AND PREPARATION	78
3.2.6 DATA PREPROCESSING	79
3.2.7 ILLUMINATION FACTORS:	
3.2.8 ENDMEMBER UN-MIXING AND CONSTRUCTION	
CHAPTER 4: IMAGE PROCESSING & RESULTS	85
4.1 NAÏVE BAYES CLASSIFIER	86
4.2 GAUSSIAN MAXIMUM LIKELIHOOD CLASSIFIER	89
4.2.1 FAST FOURIER TRANSFORM FILTERING, PCA, & GAUSSIAN MAXIMUM LIKELIHOOD	
CLASSIFIER	
CHAPTER 5: CONCLUSIONS & FUTURE WORK	99
5.1 RESULTS SUMMARY AND CONCLUSIONS	100
5.2 FUTURE WORK	106
5.1.1 DEEP LEARNING NEURAL NETWORK	106
5.1.2 IMPROVEMENTS IN SNAPSHOT HSI TECHNOLOGY	108
REFERENCES	112
APPENDIX A: SCRIPT PROVIDED BY REBELLION PHOTONICS TO DECODE BSO FILES CAPTU	JRED
BY ARROW SYSTEM	118
APPENDIX B' BSO TO HDES CONVERSION SCRIPT	123
	178
	<u> 120</u> 122
	<u>133</u>
APPENDIA E. REDELLIUN PROTUNICS ARROW SNAPSHUT MTPERSPECTRAL IMAGER PRO	420
	139
APPENDIX F: PRODUCT SPECIFICATIONS FOR SCHOTT ACE LIGHT SOURCE	<u>143</u>
APPENDIX G: HARDWARE SPECIFICATIONS	<u>146</u>

# Table of Figures

<b>FIGURE 1:</b> PROCESS FLOW OF PROPOSED SYSTEM. A SUBJECT ENTERS A CHECKPOINT WHERE A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM CAPTURES BLOOD FLOWING THROUGH SUPERFICIAL BLOOD VESSELS IN THE SCLERA. BASED ON STATISTICAL AVERAGE BLOOD CELL CONCENTRATIONS IN THE BODY DETERMINE IF A SUBJECT IS HEALTHY.THE CONTRIBUTIONS OF THIS WORK RESIDE WITHIN THE SUBJECT COLLECTION AND CELL DETECTION PORTIONS OF THIS THIS SYSTEM DESIGN.	4
FIGURE 2: BASIC REPRESENTATION OF THE HUMAN EYE (LEFT), THE SCLERA IS THE THICK WHITE TISSUE LAYER SURROUNDING THE EYE AND IS FED OXYGEN AND NUTRIENTS VIA SCLERAL BLOOD VESSELS (RIGHT)	6
FIGURE 3: PROCESS FLOW FOR MANUAL BLOOD CELL COUNTING	9
FIGURE 4: DIAGRAM OF BACK-PROPAGATED NEURAL NETWORK WHERE SPECTRAL, SPATIAL, AND SIMILARITY FEATURES ARE USED AS INPUT LAYER	21
FIGURE 5: A PROCESS FLOW OF THE FINAL SYSTEM	26
FIGURE 6: DIAGRAM ILLUSTRATING PROCESS FLOW OF THE RESEARCHCONDUCTED TOWARD THE FINAL SYSTEM	32
FIGURE 7: BLOCK DIAGRAM ILLUSTRATING DESIGN PROBLEMS PRESENTED AND ADDRESSED FROM THE ILLUSTRATION ABOVE.	32
FIGURE 8: LEUKOCYTE FAMILY TREE, THE MOST COMMON OF LEUKOCYTES (BASOPHIL, NEUTROPHIL, EOSINOPHIL AND MACROPHAGES) ALL DERIVE FROM MYEOBLAST CELLS. LYMPHOCYTES ARE LESS COMMON BUT VITAL CONTRIBUTORS THAT WORK TOGETHER AND PROVIDE A MECHANISM THAT LABELS AND REMEMBERS PATHOGENS TO INCREASE THE EFFECTIVE UTILITY OF THE OTHER WBCS [77].	36
FIGURE 9: EXAMPLE IMAGE SHOWING THAT THERE IS MORE INFORMATION AVAILABLE THAN WHAT CAN BE MEASURED FROM A STANDARD RGB CAMERA	41
FIGURE 10: AN DIAGRAM ILLUSTRATING THE BASIC COMPONENTS USED IN A LINE-SCAN HYPERSPECTRAL IMAGING SYSTEM. A SLIT SHAPED APERATURE ALLOWS A SMALL PORTION OF THE INCOMMING LIGHT FROM A SUBJECT TO BE SPLIT INTO SEPARATE WAVELENGHTS VIA A PRISM. AN IMAGING SENSOR RECORDS INFORMATION FROM EACH INDIVIDUAL WAVELENGTH AND STORES THE INFORMATION INTO A DATACUBE [64].	42
FIGURE 11: DIAGRAM ILLUSTRATING THE OPTOMECHANICS BEHIND A AN ACOUSTO OPTICAL TUNEABLE FILTER [85] THE LIGHT FROM THE SUBJET TRAVELS THROUGH A PIEZOELECTRIC CRYSTAL THAT ALTERS IT'S GEOMETRY SLIGHTLY TO ALTER THE ANGLE OF INCIDENCE FOR EACH WAVELENGTH AS IT ENTERS THE CRYSTAL. THE CRYSTAL IS TUNED TO DISTORT IN SUCH A WAY THAT ONLY SPECIFIC WAVELENGHTS EXIT ALONG THE 0-ORDER PLANE TOWARD A MONOCHROMATIC IMAGING SENSOR [64]	44

FIGURE 12: (LEFT) AN IMAGE MAPPER IS A PRECISION GROUND MIRRORED	
SURFACE THAT DIRECTS SEGMENTS OF AN IMAGE [18] (RIGHT) INTO A	
COLLECTING LENS AND PROJECTS THE SEGMENTED IMNAGE INTO A PRISM	
ARRAY [87]	45
FIGURE 13: THE LARGE IMAGE PLANE ON THE RIGHT SHOWS THE ELEMENTAL	
(RAW) IMAGE DATA CAPTURED BY A SNAPSHOT HSI. THIS DATA CONSISTS	
OF BOTH SPECTRAL AND SPATIAL INFORMATION THAT MUST BE	
RECOMBINED TO FORM A DATASTRUCTURE THAT CAN BE EASILY	
INTERPRETED BY A USER OR IMAGE PROCESSING ALGORITHM. WHEN THE	
SCENE INFORMATION IS FINALLY CAPTURED BY THE SENSOR, THE DATA	
RECOMBINED INTO A DATACUBE VIA A LOOKUP TABLE CREATED WHEN THE	
SYSTEM IS CALIBRATED.[87]	46
FIGURE 14: MICROSCOPE IMAGES OF A SNAPHSHOT HYPERSPECTRAL IMAGING	
SENSOR WITH INDIVIDUAL BAND FILTERS OVER EACH SUBPIXEL OF THE	
SENSOR. THIS DESIGN ENSURES THAT EVERY PIXEL OF THE SENSOR IS	
UTILIZED FOR DATA CAPTURE, A FEATURE THAT IMAGE MAPPING OPTICAL	
HARDWARE CANNOT PROVIDE [90]	47
FIGURE 15: DATACUBE CONSTRUCTION PER UNIT TIME FOR EACH HSI CAPTURE	
TECHNIQUE.	48
FIGURE 16: EXAMPLE OF FFT FILTERING ON LENA [99].	50
FIGURE 17: AN EXAMPLE OF SPECTRAL ANGLE (A) AND SPECTRAL DISTANCE (D)	
[107]	53
FIGURE 18: THIS FLOWCHART DEPICTS THE OVERALL PROCESS FLOW BEGINNING	
FROM, CONFIGURATION AND ADJUSTMENT OF THE SNAPSHOT HSI PRIOR TO	
DATA CAPTURE, THE DATA CAPTURE PROCESS, FOLLOWED BY INSERTING	
DATA & METADATA INTO A DATABASE, AND FINALLY HYPERSPECTRAL	
IMAGE ANALYSIS INVOLVING FEATURE EXTRACTION AND BLOOD CELL	
DETECTION	63
FIGURE 19: SPECTRAL MIXING OCCURS WHEN THE SPECTRAL INFORMATION OF	
TWO DIFFERENT MATERIALS (RBCS AND WBCS AS SEEN ABOVE) POPULATES	
THE SAME PIXEL ON THE SENSOR OF THE HSI. THE RESULTING PIXEL	
SPECTRUM IS A CONVOLUTION OF THE TWO ENDMEMBER SPECTRA FOR	
EACH MATERIAL. SPECTRAL UNMIXING ALGORITHMS FIND PROBABILITY	
DISTRIBUTIONS THAT BEST REPRESENT THE PRESENCE OF EACH TARGET	
ENDMEMBER IN THE PIXEL [94]	66
FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM	
FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]	67
FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]	67
<ul><li>FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]</li><li>FIGURE 21: (A) A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM MOUNTED</li></ul>	67
<ul> <li>FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]</li> <li>FIGURE 21: (A) A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM MOUNTED ABOVE A MICROSCOPE. (B) A DIAGRAM DEPICTING THE SETUP USED TO</li> </ul>	67
<ul> <li>FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]</li> <li>FIGURE 21: (A) A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM MOUNTED ABOVE A MICROSCOPE. (B) A DIAGRAM DEPICTING THE SETUP USED TO COLLECT DATA.</li> </ul>	67
<ul> <li>FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]</li> <li>FIGURE 21: (A) A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM MOUNTED ABOVE A MICROSCOPE. (B) A DIAGRAM DEPICTING THE SETUP USED TO COLLECT DATA.</li> </ul>	67
<ul> <li>FIGURE 20: THE ARROW SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM DEVELOPED AND MANUFACTURED BY REBELLION PHOTONICS [18], [118]</li> <li>FIGURE 21: (A) A SNAPSHOT HYPERSPECTRAL IMAGING SYSTEM MOUNTED ABOVE A MICROSCOPE. (B) A DIAGRAM DEPICTING THE SETUP USED TO COLLECT DATA.</li> <li>FIGURE 22: PIXEL RESOLUTION FOR X20 (LEFT) AND X50 (RIGHT) OBJECTIVE</li> </ul>	67

FIGURE 23: METERED SCALE USED TO DETECT DISTORTIONS AS WELL AS	
MEASURE THE EXACT DIMENSIONS OF THE FRAME	71
FIGURE 24: THE BUFFY COAT IS A THIN LAYER OF WHITE BLOOD CELLS WHICH	
MAKE UP LESS THAN 1 PERCENT OF A BLOOD SAMPLE'S TOTAL VOLUME [121]	73
FIGURE 25: REBELLION PHOTONICS CRYSTAL INTERFACE. BANDS CAN BE	
SELECTED ON THE LEFT WITH A 2D SPATIAL DISPLAY IN THE MIDDLE. THE	
GRAPH ON THE TOP RIGHT CORRELATES TO THE PIXEL VECTOR SELECTED	
BY THE CROSSHAIRS IN THE BOTTOM RIGHT IMAGE	75
FIGURE 26. A CAMPLE DAND FROM A DATACURE WITH NO FLATFIELD	
FIGURE 20: A SAMPLE BAND FROM A DATACUBE WITH NO FLAT FIELD CODDECTION ADDITED (LEET). THE SAME SAMPLE DAND IN THE ADJACENT	
CORRECTION APPLIED (LEFT). THE SAME SAMPLE BAND IN THE ADJACENT	70
IMAGE AFTER FLAT FIELD CORRECTION IS APPLIED (RIGHT).	/6
FIGURE 27: BANDS FROM A DATACUBE IN STAINED BLOOD SLIDE DATASET. THE	
THREE IMAGES ABOVE REPRESENT BANDS THAT CLOSEST RESEMBLE THE	
RED (LEFT: 471) GREEN (MIDDLE: 555) AND BLUE (RIGHT: 601) CHANNELS OF	
AN RGB IMAGE:	77
FICURE 28. AN EXAMPLE OF A SPECTRAL BAND USED TO MANUALLY MASK DATA	
FIGURE 20. AT LIAR ARY CONSTRUCTION WITH ONE WHITE BLOOD CELL	
I ARELED IN GREEN (LEET): THE SAME IMAGE MASKED WITH DIFFERENT RGR	
VALUES TO SIGNIEV WHICH CLASS THE OBJECTS BELONG TO (RIGHT)	78
VALUES TO SIGNIET WHICH CLASS THE OBJECTS BLEONG TO (RIGHT)	70
FIGURE 29: AN EXAMPLE SHOWING AN EXTRACTED BAND IMAGE (507NM) AND	
ITS MANUALLY SEGMENTED VERSION USED FOR TRAINING AND TESTING	80
FIGURE 30. DIAGRAM II LUSTRATING THE TRANSFORMATIONS APPLIED TO THE	
DATA FACH CUBE IN THE SET WAS ROTATED AND STACKED ALONG THE	
SPECTRAL AXIS A TO ENABLE EFFICIENT DATA RETRIEVAL DURING	
PROCESSING AND ANALYSIS	81
FIGURE 31: MEAN PIXEL VECTOR VALUES FOR WHITE BLOOD CELLS AND RED	
BLOOD CELLS WITH RESPECTIVE STANDARD DEVIATIONS FOR BOTH. THE	
BLACK BOX SHOWS THE ROI WHERE A DETECTABLE DIFFERENTIATION	
BETWEEN CELL TYPES IS SEEN	82
FIGURE 32: OUTPUT SPECTRUM OF HALOGEN LAMP USED IN LEICA MICROSCOPE.	
THE RED BOX OUTLINES THE OUTPUT SPECTRUM OF THE ILLUMINATION	
SOURCE WITHIN THE TARGET REGION OF THE HSI (INSIDE). THE OUTLINED	
REGION IS FURTHER EXPANDED TO SHOW THE DRASTIC VARIANCE IN	
ILLUMINATION INTENSITY THROUGHOUT THE TARGET SPECTRUM (OUTSIDE)	83
FIGURE 22. DOD VIGIDI E IMAGE (CARTURED WITTI A MOROGODE MOUNTED	
FIGURE 33: KUB VISIBLE IMAGE (CAPTURED WITH A MICRUSCUPE MOUNTED	
CANON 5D MK II) OF A KEGION CONTAINING THKEE WBCS AMONG A DACKODOLIND OF DDCS - NOTE THAT THERE IS VERY LITTLE CONTRACT	
DAUNUKUUND UF KBUS. NUTE THAT THEKE IS VEKY LITTLE UUNTKAST Detween the two tydes of cells due to spectral mixing of	
DETWEEN THE TWO TIPES OF CELLS DUE TO SPECTKAL MIAING OF DACKCDOIND DDCS DETWEEN THE WDCS AND ILLIMINATION SOUDCE	07
DACKOROUND RDCS DE I WEEN THE WBCS AND ILLUMIINATION SOURCE	

FIGURE 34: VECTO FROM T CALCU ERROR	THE SAM ERROR VALUES (HIGHER IS BETTER) BETWEEN THE PIXEL RS OF THE DATACUBE AND THE REFERENCE VECTOR CALCULATED THE SPECTRAL LIBRARY. THE ERROR VALUES USED IN THE LATION FROM THE SAME EXAMPLE IN <b>FIGURE 35</b> ON THE LEFT, VALUES FROM THE SAME EXAMPLE IN <b>FIGURE 36</b> ARE ON THE RIGHT	87
FIGURE 35: LINEAF APPLIE	CLASSIFICATION RESULTS OF IMAGE SEEN IN FIGURE 29 BY NAÏVE CLASSIFIER CLASSIFIER (LEFT) AND NAÏVE LINEAR CLASSIFIER D TO SAM FEATURES (RIGHT)	88
FIGURE 36: INFORM SPECTE	AN EXAMPLE OF POOR PERFORMANCE FROM RAW SPECTRAL IATION COMPARED TO SAM FEATURES EXTRACTED FROM THE AL INFORMATION	88
FIGURE 37: Maske Incori	EARLY TESTS OF GMLC OUTPUT FOR AN IMAGE WITH BACKGROUND D TO BE IGNORED (TOP LEFT, TOP RIGHT). PIXEL VECTORS RECTLY CLASSIFIED (BOTTOM)	90
FIGURE 38: BAND I ARE RE THE MA INFORM SEEN IN	THE ABSOLUTE VALUE OR REAL COMPONENT OF THE 2D FFT FROM A N THE HYPERSPECTRAL DATA. THE STRIATIONS SEEN IN THE DATA PRESENTED HERE AS THE CENTRAL VERTICAL LINE IN THE IMAGE. ASK (RIGHT) REMOVES THIS FEATURE FROM THE FREQUENCY MATION OF THE FFT AND THE IMAGE IS THEN RECONSTRUCTED AS N FIGURE 39.	91
FIGURE 39: HORIZO HYPER	UN-PROCESSED HYPERSPECTRAL DATA PRIOR TO FILTERING ONTAL STRIATIONS FROM THE DATA (LEFT). THE SAME SPECTRAL DATA AFTER FFT FILTERING (RIGHT)	92
FIGURE 40: DATAS BACKG	THE COVARIANCE MATRIX AFTER APPLYING PCA TO THE TRAINING ET USED IN THE EXPERIMENT ILLUSTRATED IN <b>FIGURE 37</b> EXCLUDING ROUND (NON-RBC & NON-WBC) FEATURES	93
FIGURE 41: (RIGHT IN FIGU	PCA COMPONENTS OF RAW (LEFT) AND FFT PROCESSED DATA ). CLASSES ARE REPRESENTED BY THEIR RESPECTIVE COLORS SHOWN J <b>RE 42-45</b>	94
FIGURE 42: HYPER	RESULTS OF GAUSSIAN CLASSIFIER ON RAW, UNPROCESSED SPECTRAL DATA.	95
FIGURE 43: UNPRO	RESULTS OF GAUSSIAN CLASSIFER ON PCA FEATURES OF RAW, CESSED HYPERSPECTRAL DATA	95
FIGURE 44: HYPER	RESULTS OF GAUSSIAN CLASSIFER ON FFT FILTERED SPECTRAL DATA	96
FIGURE 45: FILTER	RESULTS OF GAUSSIAN CLASSIFER ON PCA FEATURES OF FFT ED DATA	96

FIGURE 46: OBSERVABLE IMPROVEMENTS IN CLASSIFICATION PERFORMANCE OF	
USING PCA SELECTED SPECTRAL FEATURES FROM THE RAW	
HYPERSPECTRAL DATA (LEFT) AND THE FFT FILTERED HYPERSPECTRAL	
DATA (RIGHT)	97
FIGURE 47: APPLICATION OF A EROSION (A MATHEMATICAL MORPHOLOGY	
ALGORITHM) TO THE RESULTS TO CLEAN UP SPARSE ERRORS IN THE DATA	
(LEFT). COMPARED TO THE MANUALLY GENERATED GROUND TRUTH	
(RIGHT) AS WELL AS RESULTS SEEN IN FIGURE 46.	97
FIGURE 48: ACCURACY METRICS OF THE GAUSSIAN MAXIMUM LIKELIHOOD	
CLASSIFIER BASED ON THE SEVERAL PRE-PROCESSING TECHNIQUES APPLIED	
TO THE DATA PRIOR TO CLASSIFICATION.	105
FIGURE 49. ACCURACY METRICS OF GAUSSIAN MAXIMUM LIKELIHOOD	
CLASSIFIER AFTER THE FROSION POST PROCESSING TECHNIQUES WERE	
APPI IED TO THE CLASSIFICATION RESULTS	105
	105
FIGURE 50: CLASSIFICATION RESULTS FROM A FULLY CONNECTED NEURAL	
NETWORK TRAINED FROM ROUGHLY 30 UNIQUE SAMPLE DATA-CUBES OF	
WHICH MILL TIPLE DATA-CUBES WERE CAPTURED OVER A PERIOD OF 20	
SECONDS THIS RESULT IS FROM A SINGLE HIDDEN LAYER FULLY	
CONNECTED NEURAL NETWORK A MORE ADVANCED NETWORK IMPROVED	
PREPROCESSING AND MOST IMPORANTLY A LARGE DATASET MAY SURPASS	
PERFORMANCE REPORTED IN THIS WORK	107
I EKI OKWANCE KEI OKTED IN THIS WORK.	107
FIGURE 51: ILLUSTRATION OF THE BUCCAL (LABIAL) MUCOSA THE REGIOUN	
BEHIND THELIP WHICH IS VERY VASCULAR AND IS ANOTHER POTENTIAL	
IMAGING SITE	109
FIGURE 52: ILLUSTRATION SHOWING LIMITATIONS OF NON-SPECIALIZED	
MICROSCOPE OBJECTIVES, NOTE HOW THE DEPTH OF FIELD (DOF) ONLY	
OCCUPIES A SMALL VOLUME OF THE CAPILLARY TUBE. ALSO NOT THE	
FOCAL PLANE OF THE OBJECTIVE WIDENS AS THE DISTANCE FROM THE	
FOCAL POINT OR WORKING DISTANCE.	109
FIGURE 53: EXAMPLE DIAGRAM DEMONSTRATING THE DESIRED SYSTEM SETUP	
FOR PERFORMING ON SITE COMPLETE BLOOD COUNTS.	110

# Chapter 1: Introduction

In medicine, doctors extract a blood sample from a patient and analyze the sample to provide an estimated count of leukocytes per volume of blood. When the human auto-immune response activates, hormones are released to instruct stem cells in the bone marrow to form into white blood cells (Leukocytes) and travel to the site of activation. This response creates an elevated number of leukocytes in the body and is a telltale sign of a foreign pathogen or disease in the body [1]. The number of leukocytes in the body exists within a distribution that is common to all humans [2]. Using this, a doctor could diagnose a patient for certain illnesses if they find that the blood cell count distributions, in a patient, to be outside the norm.

Often, blood cell counts are performed manually by preparing a blood sample diluted to two concentrations (1:10 & 1:100 hematocrit) in a calibrated glass chamber and viewed through a Leica semiconductor grade microscope [2]. This process is time-consuming and resource intensive as it requires a skilled operator to accurately count the types of cells present in the sample [3]. In scenarios where a large number of samples need to be processed, these facilities need to have a robust and automated method of accurately counting red and white blood cells for a large number of subjects as quickly as possible.

# **1.1 Problem statement:**

There exists a performance bottleneck in blood counting procedures in that, generally, a sample of blood must be prepared and examined by a certified operator or technician. There have been other attempts to automate this process but they too suffer from similar limitations [4]–[7]. These limitations arise from methods which rely on technologies that require a sample *in vitro* ( i.e. out of the body). For example, a label or staining agent that contaminates or kills the blood sample, or the supervision of a qualified professional to ensure accurate counting of blood cells. Currently, there is neither a manual nor automated method of cell counting that could be used to perform a blood count *in vivo* (within the body).

The work performed in this thesis can be categorized by three topics. A hyperspectral image data collection of unstained blood on sealed hematology slides. The labeling and analysis of this data to determine if spectral features of RBCs and WBCs can be used in binary and multiclass classification algorithms. Finally, the development of a methodology that can detect white blood cells based on their spectral signature, without the need for stains or labels. This system design is inspired by the need to provide a point of care solution in scenarios with a large influx of individuals for example mass exposure to a radiation source. As the WBC count of a human decreases rapidly after moderate to severe acute radiation exposure [8]. The use case of such a system is not specific to radiological disasters; such a system could also be used in various applications, such as rapid blood counts for cancer patients or industrial workers exposed to hazardous materials (e.g. benzene [9]) on a regular basis.



**Figure 1:** Process flow of proposed system. A subject enters a checkpoint where a snapshot hyperspectral imaging system captures blood flowing through superficial blood vessels in the sclera. Based on statistical average blood cell concentrations in the body determine if a subject is healthy. The contributions of this work reside within the subject collection and cell detection portions of this this system design.

The diagram in (**Figure 1**) illustrates the overall functionality of the proposed system. A subject approaches the imaging interface where an operator ensures the target area of the sclera is in focus and properly exposed. The imaging device in the system is a hyperspectral camera, a camera capable of capturing the spectral information of a scene. The data is then processed to reduce noise as well as detect cells based on their spectral signatures. With enough of these spectra Identified the system segments out individual cells and counts them over a fixed period of time. Based on

vessel diameter and blood flow rate, a blood count can be performed for viable volume within an acceptable data capture duration. With the count completed, the system tells the operator whether the counting operation shows adequate results or the subject may require further attention. However, there are specific requirements to be met in order for this system to be viable. They are as follows:

- 1.) A system capable of capturing images of blood cells as they pass through a blood vessel with little to no blurring (i.e. fast integration time or shutter speed)
  - a. no marker or label will need to be injected into the subject to aid cell detection.
- A high enough frame rate in order to enable cell tracking; thus providing a way to avoid counting the same cell twice.
- 3.) A spectral bandwidth in which the peak absorption wavelengths of the target endmembers (cells) is unique to each target to be classified.
- 4.) A self-contained system that is relatively lightweight or portable that can be deployed with minimal setup.
- 5.) A data processing and classification technique that may perform accurately under varying illumination intensities.

These requirements present new and unique challenges that will require clinical testing, IRB approval, and a robust dataset. Some of the challenges set forth have been met from the work of systems implementing similar technologies [10]–[12]. However, there are many other prerequisites that must be met before any testing can be planned or approved. Possibly the most important aspect to be explored is to determine if blood cells can be distinguished from one another *in vitro* and *in vivo*.

While the system will be designed to count blood cells from hyperspectral video of blood moving through blood vessels. The system needs to first be able to count cells in a single frame. If the

system cannot accomplish this task, then it would be moot to experiment with moving cells first. Thus, the system design was broken down and isolated to its simplest components and capabilities.

The problems posed in this research do not only reside within the scope of image processing. For instance, there are several biological limitations that must be considered in order for the proposed device to be as non-invasive as possible. To image blood cells *in vivo* (without physically removing blood, such as imaging blood flow in scleral vessels) this system would need to be able to image blood vessels in an easy to access location on the body. Blood vessels lie  $250-425\mu m$  under the surface of the skin, which would then require a strong illumination source to penetrate deep enough into the skin to image the blood cells.





**Figure 2:** Basic representation of the human eye (left), the sclera is the thick white tissue layer surrounding the eye and is fed oxygen and nutrients via scleral blood vessels (right)

Alternatively, *in vivo* imaging could be performed at a location where blood vessels are extremely close to the surface of the skin. One such body part containing these blood vessels would be located in the white of the eye also known as the sclera. The blood vessels in the sclera (the Superficial episcleral network, see **Figure 2**) would be an optimal choice as they are relatively thick and reside very close to the surface of the scleral tissue (25-90 µm), giving nearly a perfect window from

which to observe blood cells without the increased spectral noise from intermediate tissues [13], [14]. However, other stabilization systems would need to be incorporated into this system to counteract the saccades (*i.e.* constant movements) in the eye [14].

The goal of this research effort is to develop a system design which will apply a snapshot hyperspectral imaging system (HSI), an imaging technique well used in remote sensing applications[11], [15]–[18], to identify and detect white blood cells (WBC). More specifically the HSI will need work in an application required to detect WBCs and RBCs that are alive and unstained. This system design would not require the staining process used in many of the other methods, thus limiting the materials and sample preparation needed in other system designs.

This thesis addresses the baseproblems of a need for an automated blood cell counting method that has the potential to perform *in vivo*. The tasks that were accomplished to meet the goal of this research effort include:

- **Task 1:** Setup of the Arrow system and calibrate the system to evaluating imaging capability in top down and bottom up illumination using stained blood cells.
- Task 2: Imaging of blood cells on histology slides to determine spectra of interest. These target spectra will contain the greatest divergence between the absorption spectra of RBCs and WBCs
- **Task 3:** Implement linear binary classification and nonlinear, multi-classification algorithms on of WBCs, RBCs, and background from the acquired data

The problems addressed in this thesis work have set the foundation for developing and testing an imaging system based on this method. The information gathered from this research will benefit greatly towards the future development of an automated system capable of performing a complete blood count *in vivo*. This method was developed utilizing a snapshot HSI equipped to simultaneously capture 31 bands within the range 461-641nm or 25 bands within the range of 419-494nm. This imaging system, attached to the camera port of a microscope, captured data of

unstained blood samples on microscope slides. Several commonly used segmentation techniques were applied to the data to determine if WBCs and RBCs are in fact discernable. Non-linear classification techniques were implemented to separate the two cell types, which display positive results with an accuracy >80% on non-ideal imaging examples. Ultimately, this work shows that these cell types are indeed separable without the use of a stain or label.

# **1.2 Manual Blood Cell Counting: Background and Limitations**

Manual blood cell counting first involves the extraction of a blood sample from a patient. Blood samples are typically extracted from one of the three major veins in the arm (the median cubital vein, Cephalic vein, and Basilic vein) [19]. Once a sample is extracted, it is 'spun down' in a centrifuge to force red blood cells to the bottom of the sample chamber. The liquid matrix or plasma is displaced by the red cells and rests on top of the 'plug' of blood cells. Resting between the plasma and plug is the buffy coat, a layer of platelets and white blood cells which make up about 1% of the sample's total volume [2].



Figure 3: Process flow for manual blood cell counting

After a blood sample is extracted from a patient, two steps must be performed before any counting can be conducted (**Figure 3**). The blood must first be diluted to a specified concentration, a 1 mL sample of blood is diluted with 10 mL of additive solution and 1  $\mu$ L of the diluted blood is then extracted [2]. The cell count is then represented as the number of cells per unit volume, more specifically, cubic millimeters (mm<sup>3</sup>). Despite the genetic diversity seen throughout human populations, the concentrations of each cell type per mm<sup>3</sup> of blood are very consistent (**Table 1**).

Manual leukocyte counts require more preparation since their concentrations within a volume of blood is orders of magnitude less than that of erythrocytes (RBCs) (**Table 1**). Not only are several additional procedures required, but specialized equipment is required as well [2]. A hemocytometer, a precision measurement instrument resembling two glass microscope slides with a grid etched on the bottom slide, is used to aid the operator in counting leukocytes per volume of blood. A few additional steps require the use of a solution that lyses (i.e. destroys) RBCs leaving only leukocytes so that they are not obscured when the solution settles on the slide. Not only is this procedure time consuming and painstaking for the operator, each step which involves a human operator can create opportunities for errors to be introduced.

**Table 1:** Typical blood cell values in young adults [2], note that while the concentration of erythrocytes varies by sex, the concentrations of leukocytes and platelets remain the same.

ERYTHROCYTES         5.21 x 10 <sup>6</sup> /mm <sup>3</sup> (Male) 4.60 x 10 <sup>6</sup> /mm <sup>3</sup> (Female)           LEUKOCYTES         7.8 x 10 <sup>3</sup> /mm <sup>3</sup> PLATELETS         311 x 10 <sup>3</sup> /mm <sup>3</sup>	CELL TYPE	CONCENTRATION		
LEUKOCYTES         4.60 x 10 <sup>6</sup> /mm <sup>3</sup> (Female)           PLATELETS         7.8 x 10 <sup>3</sup> /mm <sup>3</sup>		5.21 x 10 <sup>6</sup> /mm <sup>3</sup> (Male)		
LEUKOCYTES         7.8 x 10 <sup>3</sup> /mm <sup>3</sup> PLATELETS         311 x 10 <sup>3</sup> /mm <sup>3</sup>	ERTIAKUCTIES	4.60 x 10 <sup>6</sup> /mm <sup>3</sup> (Female)		
<b>PLATELETS</b> 311 x 10 <sup>3</sup> /mm <sup>3</sup>	LEUKOCYTES	7.8 x 10 <sup>3</sup> /mm <sup>3</sup>		
	PLATELETS	311 x 10 <sup>3</sup> /mm <sup>3</sup>		

## **1.2.1 Blood Count Limitations**

A complete blood count, while a relatively simple diagnostic tool, is a complex task to perform on a large number of patients. The ability to provide an accurate blood count rapidly for a large population could allow healthcare professionals to focus their efforts on patients that require immediate care [20]. One such example is an accident in 1987 in Goiânia, Brazil where over 200 people were exposed to a radioactive Caesium-137 source [8]. During the accident, over 100,000 people visited local hospitals to be tested for radiation exposure, commonly diagnosed using white blood cell count. Local football stadiums were used as triage centers to aid the hospital's capacity exceeding number of patients. The goal of these centers was to provide rapid exposure assessment and treatment to those exposed. Acquiring an accurate count of WBCs in an individual subject is a time-consuming process, with throughput largely limited by the number of available personnel [20].

Acute Myeloid Leukemia, a form of blood cancer that affects the myeloid family of leukocytes, is another example where rapid high volume blood cell counts can make a significant impact. There exists sufficient evidence that Benzene, a widely used chemical, causes acute myeloid leukemia [21]. This disease, if not detected early, can be fatal. However, myeloid leukemia can be prevented if physical checkups and blood work are performed frequently for individuals who are exposed to benzene on a daily basis. The need for frequent blood work could be expensive and invasive to individuals.

Many cancer treatments, chemotherapy, for instance, place a heavy strain on the body. A prescribed amount of localized radiation and drugs destroy cancer cells while simultaneously destroying healthy tissue. Keeping track of a patient's blood cell count is crucial in order to balance the strength of a treatment and the damage done to the patient's body [22]. This application could move beyond specific types of cancer screening and be used toward monitoring blood counts in patients undergoing chemotherapy. Because of these motivations, an automated system capable of screening for anomalous blood cell counts could provide an early warning system and save patients' lives.

# **1.3 Blood Count Automation**

Several automated blood counting systems have been developed to address this problem; as automating this process can provide an accurate estimate of a blood cell count that could be used for further diagnosis [4], [6], [23]–[25]. One method to expedite this process is to implement an imaging system capable of scaling throughput to provide a complete blood cell count for a large number of patients. Such systems could relieve the demand for personnel to process a high number of blood samples generated from a scenario similar to Goiânia. Capabilities of these systems should

not be limited to radiation exposure, but also include situations where quickly identifying individuals with abnormal blood cell counts would improve quarantine practices during epidemics or environmental disasters.

The fixation of cell stains involves the use of alcohols which kill the cells in the sample [25]. This protocol is common for external testing and diagnosis; it is applicable in all but large-scale, high throughput screening scenarios. To provide an alternate, scalable technique that provides an accurate blood cell count. Newer automated methods should be explored which fit a use case outside the scope of current systems. In addition, if blood cell counting could be performed *in vivo*, throughput could be greatly improved. Due to the destructive nature of cell staining by drying out the sample, the lack of scalability, and the cost of onsite expertise, the need for these automated systems grows [4], [26].

# 1.3.1 Rapid Blood Counting Systems

One approach, namely, the Rapid Automated Biodosimetry Tool (RABIT), is an automated WBC analysis system capable of processing 6,000 up to 30,000 samples per day. The system runs two fully automated, mature, biodosimetry assays, both of which are used to test for radiation exposure. However, the RABIT still processes blood samples collected by examiners in bulk, only expediting the analysis portion of providing a blood cell count for thousands of subjects. To decrease the time taken to process a sample, a system capable of imaging blood cells *in vivo* should be efficient, portable and easy to use.

Since blood counts were first utilized as a means to diagnose diseases or radiation exposure, many fields of research have explored solutions to improve count accuracy as well as scalability. The first and easiest solution was to simply stain the cells in order to allow lab technicians to see the different blood cells more clearly [27]. Within the past two decades, automated systems have been developed to perform white blood cell counts but require blood samples to be taken from a patient

and processed. Most systems are designed to perform rapid blood counts. However, some specific systems, the RABIT for example, were designed to operate in scenarios with a high throughput requirement [4].

#### 1.3.2 Photothermal Tomography Methods of Cell Differentiation

Photothermal flow cytometry[26], an *in vitro* method, is aimed at imaging cells flowing through a microvessel in a single-file fashion. These systems use lasers, emitting near-infrared (NIR) wavelengths (800nm-1.6µm) to excite cells and measure their fluorescence (if labeled) or light scattering effects to determine cell structure and type. However, due to the relatively high power of these laser systems, localized temperature increases of cellular structures occurs. Photothermal flow cytometry, while effective, is damaging to tissues if used for extended periods of time[28].

Photothermal optical coherence tomography (OCT) is a technique used to image blood *in vivo* beneath the epidermis. This imaging technique is generally used for measuring blood oxygenation [29] as well as blood flow as a diagnostic tool to determine a patient's overall health. However, spatial imaging quality could pose an issue when attempting to count individual blood cells. For example, optical coherence tomography systems usually operate around the 1050nm, or near infrared band. This band is chosen due to the low absorption value of water at this particular wavelength. However, whole blood is roughly 55% water which has a higher absorption coefficient in the shorter wavelengths of the visible spectrum. Since OCT requires reflectance interference to resolve an image, the depth at which the system can observe is severely decreased [30].

## 1.3.3 Other Systems

The Retinal Function Imager, an *in vivo* multispectral approach, was also explored to measure blood flow, oxygenation and metabolic signals in the retina. This system is comprised of a Charge-Coupled Device (CCD) camera, an imaging system with both a fast integration time and high spatial resolution, attached to a multispectral filter system. The filter system, composed of four fast moving filters, exploits a natural chromophore in erythrocytes to allow for a non-invasive label that can be used for calculating blood flow, metric which is very useful in diagnosing diseases and injuries such as hemorrhages or micro aneurysms in the retina [31].

The systems mentioned previously aren't the only ones developed to automate cell counting. There is a large market for systems that employ various different technologies that measure differing properties of blood samples *in vitro* to determine a cell count. These systems include methods involving electrical impedance [32], [33], light scattering [6], microfluidic devices [34], electromagnetic radiation conductivity (reflectance/absorption) and cytochemistry [2] [23]. Above all, each of these systems relies on disposable materials to conduct testing in a safe and sanitary manner. These disposable components, while relatively inexpensive, lead to a recurring cost over the lifetime of the systems [3]. Additionally, single-use supplies limit the capacity of a facility to process samples based on the inventory on hand at any given time.

# 1.4 Remote sensing impacts on Blood Cell Counting

The field of remote sensing has expanded beyond astronomy and geology to a wide range of fields thanks to advances in imaging systems and computer processing power [35]. Remote sensing equipment typically consists of highly sensitive sensors and produces a substantial amount of data. Remote spectrometry has been used to great success in geological surveys to detect mineral content in otherwise inaccessible regions. These systems observe and analyze the natural reflectance of sunlight from a remote material. Due to physical properties at the atomic/molecular level, each material has a unique reflectance spectrum that can be used as a specific signature [36]. These signatures can allow these systems to detect a material without requiring a physical sample nor chemical marker.

Up until the advent of general purpose graphical processor unit (GPGPU) computing in the past decade [37], processing this large amount of data would require significant computational resources to record and analyze. With GPGPU computing, remote sensing and data acquisition systems can be scaled down from requiring the processing power of a supercomputer to a desktop workstation. Consequentially, microscopy-based diagnostics, including blood cell counting, now have a new technology to incorporate into the micro-imaging domain.

### **1.4.1 Hyperspectral Imagery**

A subfield of remote sensing and remote spectrometry, Hyperspectral Imaging Systems (HSI) are cameras that can capture spectral information within their field of view. These systems differ from multispectral systems in the fact that the number of bands (wavelengths) captured in the spectrum have a narrow spectral resolution and represent a near continuous sample of a specified bandwidth [15]. These systems allow for the capture of spatial information alongside spectral information, a relationship essential in remote sensing when determining the exact location of a specific material. Since these systems can capture the spectral and spatial properties of a material, the segmentation algorithms used to exploit these properties reside in three general paradigms. The first of involves extracting features from both spectral and spatial information before classification methods are applied. This method is beneficial for scientists to incorporate specific features in classification. However, since these features are 'hand-crafted', they require human knowledge in order to train [38]. The second method foregoes extraction of spatial features prior to classification and instead uses contextual information inferred from the spatial data to classify spectral features; as seen in [39]. The final paradigm uses the spatial information after classification to act as a boundary condition. Such a method extracts features from spectral and spatial information but treats them independently. While this technique does reduce dimensionality, it does also dampen any strong correlations between spectral and spatial features [40].

Since the advent of small and inexpensive imaging components, the applications of hyperspectral imagery have branched out into the agricultural [17] and medical fields [11], [12], [18]. Systems that provide automated quality control on food processing lines and pharmaceutical production labs have become increasingly popular [41], [42]. The manufacture and testing of thin-film materials are aided by hyperspectral imagery to detect flaws and measure film thicknesses[16], [43].

While there are many industries and research fields that utilize HSI systems, there is a previously unoccupied niche in some industries, namely medical imaging. HSI systems have been applied to several different branches of medicine [44]. From applications in detecting damaged tissue in burn patients to the detection of plaques in arterial walls [45], [46]. These systems aid doctors and medical professionals in identifying specific regions of interest that may otherwise go unnoticed or undetected. Regions of interest might include less severe burns surrounded by heavily damaged tissue or plaques in adjacent arterial walls that might be overlooked when targeting larger deposits. These systems can be incorporated as a valuable tool used on a daily basis to increase the accuracy and precision of treatments for patients as a second method to verify a medical professional's diagnosis.

The ability for these systems to be able to differentiate healthy from damaged tissues without the need for a biopsy or label presents some significant advantages over current systems or methods. For example, consider systems described in the above paragraph. The advantage of a system that can detect and differentiate burned or damaged tissue from healthy tissue could decrease the area where skin grafts would be applied. The reduction in the area of grafted skin will improve a patient's recovery and potentially reduce chances for infection [47], [48]. Likewise, the ability to detect smaller plaque deposits that would have otherwise gone unnoticed would aid surgeons in treating patients with a higher success rate. These systems and their applications in the medical field hold enough merit to explore applying this technology to blood count automation.

#### **1.4.2 Blood Tissue Optical Properties**

Blood is one of the main tissues in the human body, it consists of blood cells and their fluid matrix, plasma [2]. Although, it is the fluidic nature of blood which presents a difficult task in modeling its optical properties within the body. As blood is a heterogeneous mixture its optical properties too are heterogeneous, meaning that changes in the population density of red cells in the plasma can change rapidly thus altering optical properties as well [49]. Like other tissues explored and examined with the devices seen in [45], [46], the same could be done with blood tissue.

Another optical property of blood that is worth mentioning is its light scattering properties. Due to their pseudo-toroidal shape, RBCs tend to scatter light anisotropically [50]. Meaning the intensity of the scattering effect is dependent on the direction photons enter the cell. However, based on findings in [50] these scattering properties do not change based on the oxygenation of hemoglobin in RBCs nor if the cells were deformed in shape so long as if volume and hemoglobin properties are kept constant. This last piece of information justifies why some datasets were used in training and testing. These recorded optical properties of blood can be used as a solid reference to explore detecting cells via a hyperspectral imaging technique.

# **1.5 Similar Applications**

In the past, several attempts have been made to image white blood cells either as a means to assist in counting blood cells as well as performing a full blood analysis. A common approach has been to use RGB color imaging systems to differentiate cell types present in the blood. The following sections summarize prior research in the pursuit of detecting blood cells.

#### 1.5.1 Blood Cell Detection and Classification Using an RGB Image Processing Approach

The research methods described in the following section are approaching the task of detecting white blood cells using color (RGB) imaging systems. The image processing techniques, applied to the

data, demonstrate how the structure of WBCs possess unique features that allow for accurate segmentation of the cells' cytoplasm and nuclei. However, the spectral features of RBCs and WBCs that best distinguish the two types exist in a narrow band between 420-440nm [30], [49]–[51].

#### 1.5.1.1 A Framework for WBC Segmentation in Microscopic Blood Images using Digital Image Processing

Digital image processing techniques have provided significant improvements in a wide range of imaging based applications. The methods proposed in this paper explore the use of spatial features obtained from the blood cells to distinguish RBCs and WBCs. The RGB image data was converted to grayscale images before any segmentation methods were applied. The algorithms used implemented the use of active contour models both geometric [52] and parametric [53] to detect the contours of the nuclei in WBCs as they are not present in RBCs. The applied methods produced a method capable of segmenting the nuclei of WBCs with 92% accuracy and cytoplasm with 78% accuracy [7].

#### 1.5.1.2 A New White Blood Cell Segmentation using Mean Shift Filter and Region Growing Algorithm

A method proposed in this research implements the use of the CIE color space [54] of the image to detect blood cells based upon their color in the image. By transforming into this color space, it is possible to utilize a general intensity value (L) as well as the color information (a & b) independently. The results of such methods produced an efficient method of de-noising and segmenting blood cells from RGB images. However, performance results of these experiments are based solely on the manual selection of segmentation parameters, there is no general model to segment cells without fine tuning [55].

#### 1.5.1.3 White Blood Cell Segmentation by Color-Space-Based K-Means Clustering

The research proposed in this work was to develop a method to detect white blood cells using colorspace information of the image. Implementing an unsupervised learning technique (K-means) allows for the detection of optimal features in a given color space, in this case, the CMYK color space [54]. This clustering technique, when tested on a dataset of over 600 stained blood slide images produced an average accuracy for all WBC cell types at 95.7% [56].

## 1.5.1.4 Visual Saliency Attention for Localization and Segmentation in Stained Leukocyte Images

A fast hierarchical framework developed for localization and segmentation of leukocytes is proposed in this work. A general description of the methods implemented in this framework includes mean shift filtering to remove background noise as well as watershed segmentation [57]. A saliency model, which is based on the contrast of features in the cell compared to its neighbors, based on the impurities or DNA in the nuclei of WBCs is used to provide localization of a WBC in the image. The next methods utilize the spatial information of the cells in a watershed segmentation technique to differentiate WBC cytoplasm from nuclei. This framework is very accurate with an average performance of 94.1% across all WBC types in a stained blood slide image.

#### 1.5.1.5 A Summary of RGB Blood Cell Detection Methods

source	method	domain	performance	data
[7]	active contour model	spatial	92.0%	stained blood slides
[55]	Mean Shift filter & Region Growing	color	-	stained blood slides
[56]	K-Means clustering	color	95.7%	stained blood slides
[58]	Mean Shift & watershed segmentation	Saliency, spatial	94.1%	stained blood slides

Table 2: A summary of performance from RGB WBC detection methods

In terms of blood cell detection, the methods demonstrated in the previously described papers are significant. However, the data used in these experiments were images of stained blood samples [7], [55], [59]–[62], [56]. The goals of this thesis research are to detect blood cells without the use of a stain. Thus, this work is not comparable within the scope of this thesis research (**Table 2**). The justification of which is that the philosophy behind cell staining, in general, is to improve contrast between difficult to distinguish structures within a cell (e.g. WBC nuclei and the cytoplasm surrounding them). Unstained RBCs and WBCs are very difficult to distinguish due to the fact that there is little contrast between cell structures without the use of a label or stain [27], [63]. If a stain cannot be applied to the cells in order to allow for the application of *in vivo* cell detection, then an alternative image capture method must be explored.

### 1.5.2 Blood Cell Detection and Classification Using Hyperspectral Imaging Systems

The works described in the previous section all implement segmentation algorithms used in grayscale or RGB images. However, hyperspectral systems have a significant advantage in that many materials have unique spectral signatures that can be detected and exploited by an HSI. While many digital image processing segmentation algorithms can be applied to HSI data, they are not exploiting the spectral information that may provide a better separability between two similar objects in the spatial domain [17], [44], [64].

### 1.5.2.1 Blood Cell Classification Using a Hyperspectral Imaging Technique

Relevant research in this particular field has focused on classifying blood cells using a line-scan hyperspectral imaging system. Using a system with 240 bands and a spectral range from 400-860nm, the system could cover the whole visible and some of the near infra-red wavelengths [11]. The setup design in this research involved imaging histology slides containing blood cells taken from first-time leukemia patients prior to their first chemotherapy. The images captured with the system were then preprocessed via noise reduction, baseline subtraction, and transmittance conversion. The research incorporated features from both the spectral and spatial domains. Through analysis of the reflectance spectra of the cells, it was determined that several specific features could be enough to classify red and white blood cells.



**Figure 4:** Diagram of Back-Propagated Neural Network where spectral, spatial, and similarity features are used as input layer

The classification method implemented a multi-layered, back-propagated neural network trained with all of the features extracted from the data in preprocessing (**Figure 4**). Features containing spectral, spatial as well as similarity data were passed into the network on a per-pixel vector basis. Training of this network was performed using a modified version of the gradient descent algorithm, a stoichiometric algorithm that commonly converges to a local minimum rather than an optimal or global minimum. This modified training method incorporated the use of an adaptive genetic algorithm to converge toward optimal weights with a higher "immunity" toward local minima.

The performance of this modified neural network was evaluated using a three-fold cross validation where each sample was subdivided into three subsamples, two of these subsets were used in training where the remaining subset was used for validation. With an average performance of 87.5% and 92.5% for White and Red blood cells respectively; there lies room for improvement mentioned in future works. Some improvements mentioned are increasing feature extraction accuracy and some modifications to the classifier to improve performance [11].

This system was developed and tested around a spatial or 'push broom' scanning hyperspectral camera. The system tested and performed very well on images from blood slides, however, tests with cells moving in blood flow may pose a challenge due to the scanning HSI's capacity to only capture image data along a single axis. This limitation would prevent this method from being employed in a system designed to capture hyperspectral images of cells moving through a blood vessel

## 1.5.2.2 Leukocyte identification based on molecular HS imaging

Prior works explored several feature extraction, as well as classification, techniques in an attempt to find an algorithm that could be applied to identifying blood cells. The focus of this research was the identification of morphological differences of cells infected with different diseases [12]. Blood smears were imaged utilizing a scanning hyperspectral imaging system with an Acousto-Optical Tuneable Filter (AOTF). With a spectral range from 550nm to 1000nm consisting of 80 bands, the system was capable of imaging in the visible green to near-infrared wavelengths.

The data used was calibrated to compensate for the output spectrum of the illumination source by solving for the gray correction coefficient, a technique commonly used for white-balancing images [65]. Once white-balancing operations were performed, the data was then normalized and ready to be sent through a classifier. Principal Component Analysis (PCA) was performed on the spatial portion of the data, and a K-means unsupervised classifier was used to segment the most important

principal components. The spectral features of leukocytes nuclei and cytoplasm were extracted using the fuzzy C-means classifier [66]. The output of these two classifiers was combined into a weighted voting system similar to soft-max and the highest voted label was selected as the output. This algorithm combining bot spectral and spatial information performed with 92% and 85% accuracy in distinguishing nuclei and cytoplasm of white blood cells respectively. The computation cost of this algorithm, roughly 26s on an Intel Core i5, is expensive compared to the Spectral Angle Mapper with a computation time of 1.1s on the same system.

The system implemented in [12] utilized an AOTF spectral scanning HSI. A system capable of capturing a large number of bands but not any two bands simultaneously. This poses a limitation in the system for capturing fast moving targets. Temporal distortions, the difference of the position of an object from one band to the next, occur in spectral scanning HSIs and cannot meet the requirements of an *in vivo* blood cell counting system. However, the methods implemented show an unsupervised classification algorithm that may be used to validate results.

# 1.5.2.3 Hyperspectral Image Classification via contextual deep learning

Contextual deep learning has also been used in conjunction with hyperspectral imaging for object classification. Research presented in [5] describes that most hyperspectral imaging techniques only utilize spectral information on a pixel vector basis. However, the spatial component of these images possesses a significant amount of useful information. The scope of the research presented explores first of three major categories of hyperspectral image classification spectral-spatial classification where spectral and spatial features are extracted before classification. A deep learning approach was implemented to utilize both spectral and spatial features in such a way that minimizes the Hughes phenomenon, otherwise known as the curse of dimensionality [67].

The proposed method in this research was to implement a new deep network design called Contextual Deep Learning (CDL) [5], it is a uniquely structured neural network consisting of 3 hidden layers. The first layer merges spectral information from adjacent pixels to add a spatial component to each pixel vector. The next group of layers train extensively on each pixel vector to learn each spectra from the input. The third layer is a smoothing layer which learns the gradient of the spatial information in the selected neighborhood for the pixel vector.

CDL networks performed exceptionally well in instances where training examples are limited either in number or in resolution. With an average accuracy score of 98.26% on the AVIRIS Indian Pines dataset, CDL networks outperform prior network architectures[5], [39]. CDL networks outperform in scenarios with low-resolution hyperspectral data as well, although with a neighborhood window of a smaller size to compensate for the limited resolution.

The data processing methods explored in [5] implement both spatial and spectral data to extract features that do not require advanced classification methods. Using a neural network to extract features from the data can ensure that only useful features are used. Useful features that may validate manual segmentation methods performed by a technician with little to no histology knowledge. The main improvement contributed from this work, however, is implementing a deep neural network capable of learning spectral features with respect to the spatial location to neighboring pixels. A practical approach to distinguish spectrally similar or mixed materials such as whole blood.
#### 1.6 System Proposal

Previously developed systems possess limitations in terms of performing a complete blood cell count. For example, most automated systems designed for high processing output require blood samples to be collected from subjects prior to analysis. This approach, while advancements and design improvements have increased their efficiency and effectiveness for large scale screening, unfortunately sacrifices its ability to manage small sample sets as well as other manual or automated designs. As a small sample size would still require the same amount of preparation and analysis steps, the advantage of large scale systems is their ability to process samples in parallel. Therefore, as the sample size decreases for these massively parallel systems, so too does the efficiency.

The proposed system would perform in scenarios where samples would be collected and processed asynchronously (i.e. independently). One sample would not have to wait for a minimum sample size to be reached before it could be processed. An example application would be a "checkpoint" style screening process where samples can be quickly processed sequentially and independently from other samples. This allows for benign samples to be screened and directed accordingly, thus deallocating vital resources that may be used for anomalous or positively detected samples.

The challenges associated with employing HSI in *in vivo* and *in vitro* blood cell counting resemble the common issues found in remote sensing applications[68], [69]. However, there is another challenge involving spectral un-mixing of the endmembers detected *in vivo* due to the fact that the cells in the connective tissues of blood vessels and dermis also possess similar contents to leukocytes. In order to overcome this obstacle, a HSI system needs to be incorporated into this system that meets or exceeds minimum requirements to reasonably achieve the ultimate goal of a portable, rapid, and automated blood count system.

# 1.6.1 System Operation

The intended operational flow of the system (**Figure 5**) is to provide a complete blood count for a subject or patient in a rapid and comfortable procedure. To begin, a subject is seated or in a prone position while a trained medical professional locates suitable blood vessels in the sclera to image. Once a suitable imaging site has been located, the device then measures the dimensions of blood cells and vessels in the image. Once initial measurements have been correctly determined, using blood cells themselves as a point of reference as their average diameter is fairly constant in humans [2]. Using these estimated dimensions, the system can then calculate blood flow and utilize spectral information to distinguish individual cell types. The system then extracts spatial and spectral features from the incoming images to identify each cell type and track each unique cell as it passes through the frame of the imaging system.



Figure 5: A process flow of the final system.

Spatial features of the images, while useful for tracking and differentiating individual cells, are not as helpful in identifying the cell types. As blood cells are very elastic objects, their morphology and size change from collisions, blood tonicity and genetic factors (e.g. sickle cell anemia) [25] [70]. The measurements and extracted data can then be used to calculate an estimated blood count based on the number of cells detected and the estimated flow volume through the blood vessels.

## 1.6.2 System Design

The initial concept to explore a new solution to this problem was to research current or newly developed systems that were well suited to this particular application. The most important requirement of this research was that the application of this system allowed for non-invasive blood cell counting. From this stipulation, many obvious choices were no longer applicable to the problem as systems used in similar research required the use of cell damaging staining processes or the extraction of a blood sample.

Ultimately, this system will need to perform its task at competitive level as a method to provide rapid on site blood cell count for medical professionals. Its target environment may vary from controlled hospital wards to on site triage for disaster victims. As such, this system will need to be modular and portable in order to perform functions in remote locations that may or may not have reliable power sources. Since the device will need to be portable, there will need to be a space and power requirement for most components in the system. From a human factor standpoint, the final device requirement specification should elicit a battery life and system weight comparable to that of a mobile tablet or laptop.

In order implement the required specifications the components within the system will need to meet a minimum set of requirements. Its hyperspectral imager needs to provide a useable image in sub optimal conditions (low illumination, remote location, limited network or infrastructure access) and yet capture enough information to fulfill its function as a blood cell detection and counting system.

27

The intended methods to be used in performing this task are directly dependent on the specifications of the imaging system. The camera system will need to possess an integration time (the time it takes to capture a frame) under 100µs, short enough to eliminate any motion blur that would occur from cells traveling through blood vessels [71]. There also needs to be a method implemented within the system to determine blood vessel diameter for flow estimation. As blood flow is a crucial metric in order to provide an estimated blood count based on the number of cells that pass through the system's imaging region over a given period. The spectral bandwidth of the imaging system should cover the region of interest (420-440nm) to collect enough information necessary to differentiate RBCs and WBCs.

Camera specifications aside, there are several other features this system must possess. In order to process all of the data captured by the camera, the device must contain hardware designed for image/video decoding and processing. Alternatively, the system could be linked to remote host computer where the data could be streamed from the device to the remote system for processing. The later method may prolong battery life of the device by offloading computationally expensive operations to a remote host. However, it introduces a dependency on a local network or stable infrastructure to provide power to the host system.

## 1.6.3 System Impact:

While the proposed design of such a system has not been developed, there are other systems that perform a similar function but are designed for different applications. These systems were developed to either provide a rapid blood counting method that can scale to a large number of patients. As well as other systems which were developed to count blood cells without the need for a dye or fluorescent label to be applied to the blood sample beforehand. Other systems are used to measure and monitor blood flow in vivo however they do not possess the spatial resolution to distinguish individual cells, let alone the type of said cell. **Table 3** enumerates all similar systems mentioned in (section **1.5 Similar Applications**).

System	Description	in vitro/ vivo	Manual/ Automated	Imaging Technology	Processing Time	Application	Commercial/ Academic
[4]	RABIT	in vitro	Automated	Visual: Label	~10s	Radiation Exposure Detection	Commercial
[28]	Photothermal Optical Coherence Tomography	in vivo	Automated	ОСТ	4-6s	Blood Oxygen Saturation	Academic
[33]	Coulter Counter	in vitro	Automated	Electrical Impedence	60s	Complete Blood Count	Commercial
[72]	Sysmex DI-60*	in vitro	Automated	Flow Cytometry	3-20 min	Complete Blood Count	Commercial
[24]	Abbott CELL- DYN Sapphire*	in vitro	Automated	Flow Cytometry: MultiAngle Polarized Scatter Separation	<60s	Complete Blood Count	Commercial

## Table 3: Comparison Table of Related Systems to Proposed System

\* commonly used systems

To aid examiners when performing blood cell counts, blood cells extracted from a blood sample are typically stained to help examiners detect and identify white and red blood cells [27]. This staining procedure involves chemical stains that can penetrate specific membranes of blood cells and bond to proteins within. The cells are then killed during the fixation process which involves an alcohol washing stage to fix the stain to the cells permanently [25].

While there have been several systems developed to perform and automate blood cell counting. There isn't one system capable of meeting the specifications required to solve this particular problem [24], [33], [72]. A new system would need to be developed; based on technologies used in the remote sensing field as an alternative method to imaging and identifying particular cells based on their chemical content. We develop experiments to determine the target spectral signatures (also referred to as target spectral endmembers) of blood cells and test if these endmembers are unique enough to easily distinguish red and white blood cells.

An alternative approach for imaging live cells involves the usage of hyperspectral imaging systems (HSI) [73]. HSIs capture spatial information (an image) in a wide variety of specific spectral bands. These bands are stored together in a data structure called a 'datacube'. A datacube refers to the three dimensional data structure in which the spatial information of each band (wavelength) is stacked on top of one another.

# **1.7 Thesis Contribution**

The overall design of this system is to provide an alternative method to provide a rapid blood cell count independent of other subject samples that can scale to large sample sizes by deploying multiple systems. In order to fulfil this ultimate design goal, the following sub-goals need to be met.

- 1.) A system capable of capturing images of blood cells as they pass through a blood vessel with little to no blurring (i.e. fast integration time or shutter speed)
  - a. no marker or label will need to be injected into the subject to aid cell detection.
- A high enough frame rate in order to enable cell tracking; thus providing a way to avoid counting the same cell twice.
- 3.) A spectral bandwidth in which the peak absorption wavelengths of the target endmembers (cells) is unique to each target to be classified.
- 4.) A self-contained system that is relatively lightweight or portable that can be deployed with minimal setup.
- 5.) A data processing and classification technique that may perform accurately under varying illumination intensities.

In order to construct such a system as the one proposed, there are many underlying components that have yet to be utilized in this application. In order to construct a system on a foundation of these systems, several tests and scenarios need to be explored in order to determine the viability of these technologies in this application. In order to utilize a hyperspectral imaging system to track and count blood cells, it first needs to be able to detect enough unique spectral information needed to differentiate them. The target application of this system will rely on whether a hyperspectral imaging system is capable of capturing and un-mixing spectral information from blood cells, inside blood vessels, under superficial layers of tissue cells. Therefore, the first methods to explore would be to determine if types of blood cells do in fact possess unique spectral signatures and if so can they be detected to differentiate these cells.

The research conducted to achieve this final goal covers studying the underlying mechanics needed to achieve a reliable blood counting system (**Figure 6 & 7**). The tasks covered are as follows:

- 1) <u>Data Collection</u>: Testing the capabilities of a snapshot HSI to image unstained blood cells.
  - a. Compare this capability to that of systems using RGB imaging systems
- <u>Data Processing</u>: Labeling the data such that each cell is masked in a specific RGB value which can then be used to generate labels for binary and multiclass classification algorithms.
- 3) <u>Cell Recognition</u>: The development of a methodology that can detect white blood cells based on their spectral signature, without the need for fluorescent markers or staining



**Figure 6:** Diagram illustrating process flow of the research conducted toward the final system. .

The first method presented is an attempt to image unstained blood on slides with a snapshot hyperspectral imager. As there is little to no datasets containing reference data to start from, blood samples must be prepared and imaged to build a robust dataset. The second method builds off of initial data collected to construct a segmentation and detection algorithm to accurately determine whether a cell is a leucocyte or erythrocyte.



**Figure 7**: Block Diagram Illustrating Design Problems Presented and addressed from the illustration above.

An imaging system was constructed using the Arrow HSI attached to the top of a Leica microscope to capture images of prepared raw blood samples. The objective is to collect enough data to construct a spectral endmember, a point of reference needed to begin basic feature extraction and classification. These endmembers are a simplistic reflectance spectra of a particular material, in this context a white or red blood cell.

With the endmember constructed the next step was to determine the best approach to gathering an accurate endmember via statistical averaging as well as non-supervised clustering. Using these endmembers, feature extraction methods were applied to the data to find features of most significance in differentiating white and red blood cells. Several dimensionality reduction techniques were applied to determine if these features best represent the data into a domain that can be interpreted and classified. Finally, these features were then classified via Naïve Bayes, Support Vector Machine

## **1.8 Thesis Organization**

Chapter 2 will describe techniques and theory behind the experiments conducted in this research. Chapter 3 lists in detail the processing methods developed to image, label and detect cells using the Rebellion Snapshot HSI. Chapter 4 describes the tools used in this research as well as the methods designed to collect and analyze data. Chapter 5 contains conclusions and lists a brief summary of a path for future work towards development of a blood counting system based on the desired specification described in (1.6 System Proposal).

# Chapter 2: Theory

#### 2.1 Blood Cells & the human Immune System

Blood is a tissue comprised of four components: plasma, red and white blood cells, and platelets. Its primary function is to distribute essential nutrients throughout an organism's circulatory system. In humans, three of the four components of blood are distinct cell types: red blood cells, white blood cells, and platelets. Red blood cells (RBC, Erythrocytes) contain a protein, hemoglobin, which binds very well with oxygen ( $0_2$ ) and distribute it to cells in the body [74]. Once the  $0_2$  has been transferred to bodily tissues, hemoglobin then binds with CO<sub>2</sub> and carries it away to the lungs where it is exchanged for O<sub>2</sub> and the cycle repeats. The shape of RBCs is unique compared to other blood cells, they possess a biconcave structure formed by hemoglobin proteins [49], [50]. This structure increases surface area to aid in diffusion of oxygen molecules across the cell membrane.

Platelets (Thrombocytes) are fragments of cellular material which are used to form clots when exposed to collagen [75]. This normally occurs where blood vessels have been damaged or ruptured, platelets will bind to collagen fibers in damaged blood vessel walls. This clot eventually seals off the damaged region and allows the blood vessel to heal with scar tissue.

The main function of leukocytes (WBCs) is to fight off foreign cells and materials. These cells make up the active part of the immune system. These blood cells possess a nucleus, unlike platelets and red blood cells which extrude their nucleus during cell creation. However, this is where the function and features of each type of white cell differ. White blood cells are divided into two families' myeloid leukocytes and lymphocytes. The myeloid family contains the majority of different blood cells including platelets (thrombocytes), RBCs (erythrocytes), Basophils, Neutrophils, Eosinophils, monocytes and macrophages (**Figure 8**). The lymphocytes consist of T Cells, B cells and Natural Killer Cells.

While the functions of most leukocytes are diverse, their overall structure and composition is relatively similar. They all possess a nucleus containing DNA and most contain granules, tiny

structures that generate substances needed by the cell to perform it's cell killing functions [76]. Leukocytes of all types make up the immune system however, not all of these cells end up in blood [25]. The lymphocytes for example, mostly reside in the lymph nodes and spleen. Macrophages, a type of granulocyte, flow through the blood stream for a short period before spending the rest of their life cycle residing in a layer of tissue cells. Due to their wide diversity of functions, only (3 or 4) types of Leukocyte are commonly found the bloodstream.



**Figure 8:** Leukocyte Family Tree, the most common of leukocytes (Basophil, Neutrophil, Eosinophil and Macrophages) all derive from Myeoblast cells. Lymphocytes are less common but vital contributors that work together and provide a mechanism that labels and remembers pathogens to increase the effective utility of the other WBCs [77].

The Neutrophils are by far the most common leukocyte found in the blood stream. These cells are roughly 10-12(um) in diameter and possess a multi-lobed nucleus and make up around 62 percent of all white blood cells in the blood stream. These cells are the first to attack invading microbes when an infection occurs. They respond by absorbing the invading microbes and digest them through phagocytosis. This process ultimately kills the Neutrophil due to its inability to synthesize

new lysosomes in the cytoplasm [63]. These dead Neutrophils are eventually broken down by Macrophages, another type of leukocyte, and form a viscous fluid commonly known as pus.

Monocytes are another type of leukocyte that share a similar function to Neutrophils by absorbing and digesting foreign pathogens. Other than their role of fighting invading microbes, Monocytes also carry dead segments of pathogens until they encounter T-cell lymphocytes. Once encountered, the T-cells can manufacture antibodies for that particular pathogen allowing the immune system to fully recognize and destroy the invading microbes. Sometimes, Monocytes will migrate into body tissues and change into macrophages, long lived pathogen absorbing "clean up" cells to fight off secondary infections.

#### 2.1.1 Optical Properties of Leukocytes

Reflectance is the capability or effectiveness of a material's surface to reflect radiant energy. This effectiveness varies depending on the amount of radiant energy that hits the material. This property is unique for different materials as the energy absorbed or reflected is directly affected by the atomic structure of the material [78]. However, some energy from photons is absorbed by a materials atoms or molecules causing outer shell electrons to momentarily jump up to a higher energy level. This energy level cannot be maintained for long however, and the electron drops back down into its stable energy level. When this happens, a photon of a lower energy than the one absorbed is released. This is what causes a material to fluoresce, an effect that causes a material to radiate electromagnetic energy most commonly in the visible spectrum. Often, this energy emission creates a peak reflectance at this spectral band when higher energy wavelengths are present in the scene (**Table 4**). This physical property of many materials is exploited in many applications including the field of remote sensing.

**Table 4:** A list of the wavelenghts where different components of blood tissue have a diverging absorption peak compared to surrounding cells

WBC local max absorption coefficient (vis)[79]	440NM
Whole blood local max absorption [50]	420NM, 540NM
WRC nuclei local auto fluorosconco [80]	366NM (ABSORB)
WDC nuclei local auto-nuol escence [ov]	436NM (EMISSION)

#### Whole Blood Optical Properties in UV-Visible Spectrum

Similarly, a material may completely absorb certain wavelengths of electromagnetic radiation and reflect others. While materials consisting of one element absorb or reflect EM radiation uniformly throughout, a compound material's molecules contain structures where the absorption or reflection of certain wavelengths is more dramatic. These structures, known as chromophores, are what give compounds their reflectance and absorption spectrum as well as their color when observed in the visible spectrum [81].

In the field of microscopy, labels, compounds that possess a strong fluorescent effect that also bind to specific molecules, are used to mark cells or tissues to help observe or identify structures. This method is different from traditional staining methods in that the cells being observed do not have to be destroyed to stain them. This imaging technique has wide applications in the biological, medical and other scientific fields; mainly used to observe living materials without impacting their natural functions or environment. Due to a natural occurring phenomenon found in DNA, white blood cells fluoresce at certain wavelengths of light when exposed to ultraviolet radiation(366nm & 436nm) [80]. This phenomenon, called auto-fluorescence, is a substance or material's natural ability to absorb energy from photons at certain wavelengths and then emit energy in the form of photons at a lower energy level. White blood cells fluoresce the most between 500-550nm when exposed to a 436nm source. This characteristic is useful for identifying white blood cells without the use of a stain or label. This natural reflectance presents an additional method to providing a means to amplify specific wavelength intensities that are not expressed in red blood cells. Exploiting these natural phenomena may help to segment a cell nucleus from cytoplasm.

However, blood cells as well as most biological samples, are susceptible to UV radiation which can cause photo-bleaching or accelerate cell death. Photo-bleaching occurs when the molecules containing chromophores or luminescent structures become decomposed/denatured by the higher amount of energy contained in UV radiation. Similarly, it is this denaturing of proteins in the cells which causes cell death.

# 2.2 Endmember Storage Database

Hyper-spectral data by nature is very large. One of the initial challenges when analyzing these images is how to manage so much information stored in each datacube. There are several different approaches to storing, pre-processing and analyzing. This section explains the merits found between each method.

The process of reading data stored in BSQ and image files is a very process heavy task as the data stored in these files needs to be transformed into a structure that software can access. To speed up this procedure, a database file is used. There exists a large diversity of data storage structures each with particular benefits in terms of portability, Input/output operations (I/O), data distribution as well as a few other factors.

A problem more commonly seen in big data applications is the issue of having more data than what can actually fit into a system's memory. This causes a problem when data access could be random, causing large chunks of data to be dumped from memory and then new chunks written into memory, a very resource intensive process. To overcome this issue, several database structures were designed to handle splitting up data into manageable sized blocks that can speed up throughput and not cause a bottleneck when sending data to the processor(s). In the context of this thesis research, an HDF5 database was used to store endmember data.

# 2.2.1 Hierarchical Data Format

The Hierarchical Data Format (HDF5) is an entire filesystem contained within a single file in memory. This data structure design allows for a large database to be divided into smaller 'chunks' while minimizing as much system memory overhead as possible. One major benefit to this is that more data is stored in memory at any given moment over each example in its own reserved piece of memory. These benefits equate to scalability across distributed systems and access from multiple distributed systems while maintaining data integrity.

#### 2.3 Blood Cell Segmentation Techniques

With the increase in available compute power over the past 20 years, newer and more accurate segmentation methods have been developed. Spatial information has been incorporated into segmentation algorithms to improve accuracy of detection models. Several of these composite kernels show improved performance over methods that only rely on spectral information [5]. The following sections briefly describe these algorithms and their effectiveness.

Blood cells are primarily water and proteins, making it difficult to differentiate the two without the aid of a stain or label. However, Leukocytes contain proteins and nuclear material not present in red blood cells. The purpose of this research is to prove that the spectral absorption of these materials is different enough to discern whether an object is a white or red blood cell. Originally, methods used to count the number of blood cells in an individual involved extracting a blood sample and manually count the number of white cells present per unit volume of blood. The time and resources needed to perform these measurements are too great to deploy in security checkpoint or triage type scenario. The challenges posed to speed up this process as well as reduce the amount of resources needed to do so require the application of devices that can perform this operation either very quickly or in a large sample, parallel operation.

# 2.4 Hyperspectral Imaging Systems:

Hyperspectral imaging (HSI) systems capture spatial and spectral information of an object or material (**Figure 9**). This is achieved in one of several methods, each of which possess advantages and flaws for particular applications. Ultimately, the best system for an application is determined by the requirements specified, the platform from which the system will operate, and the subject matter that is to be imaged.



**Figure 9:** Example image showing that there is more information available than what can be measured from a standard RGB camera.

#### 2.4.1 Line Scan Hyperspectral Imaging Systems

HSI line scan systems capture each spectral band along a single dimension as a scanned composite image of the object or area being viewed. Often referred to as 'push broom' hyperspectral imaging systems, these devices are deployed in applications from quality control in food production to remote surveying of geographical regions. The defining trait of these systems describes the narrow  $(1 \times N \text{ pixels})$  imaging field of view of the system (as seen in **Figure 10**). The system obtains its images using an optical slit to limit only a thin strip of the image to pass through a prism or grating that then projects the diffracted scene onto a sensor. By limiting the amount of scene (spatial) information into the system at any given instance, the majority of the imaging sensor's area can be utilized to capture spectral information. This reduction in spatial resolution allows for simultaneous capture of data at a higher spectral resolution.



**Figure 10:** An diagram illustrating the basic components used in a line-scan hyperspectral imaging system. A slit shaped aperature allows a small portion of the incomming light from a subject to be split into separate wavelenghts via a prism. An imaging sensor records information from each individual wavelength and stores the information into a datacube [64].

This data capture technique is a practical solution for applications where a scanning operation is possible. In particular, airborne mounted systems that image the ground surface as an aircraft flies overhead. Food quality inspection is another successful application of these particular systems as they can rapidly detect defective or unhealthy produce. In some food inspection applications these systems can be used to provide quality grading of each item as it travels down a processing line [17], [41], [82].

While this technique provides both high spatial and spectral resolution, line scan HSIs are very susceptible to changes in their target subject's morphology. Which means the system must be fixed to a steady structure as the subject passes through its linear field of view or, the subject is stationary as the system's imaging plane scans over the subject at a steady rate. Presenting a major disadvantage specific to *in vivo* blood cell counting. RBCs often collide with one another and pass over each other as they travel through blood vessels [6]. If this were to happen as they passed through a line scan HSIs imaging plane, distortions created by these movements may introduce confusion to detection systems. Additionally, WBCs tend to rotate and adhere to vessel walls as they travel through the target capture region [83], causing the image recorded by the system to become spatially distorted.

# 2.4.2 Spectral Scanning Hyperspectral Imaging Systems

Spectral scanning HIS systems, such as those employing an Acousto Optical Tunable Filter (AOTF), use piezo-optical devices (see **Figure 11**) that allow specific wavelengths of electromagnetic radiation pass through to a broadband camera sensor [12]. AOTFs are not the only filtering technology that can selectively allow specific wavelengths to pass through. Liquid Crystal Tunable Filters (LCTF) use the same technology found in Liquid Crystal Displays (LCDs) to adjust its spectral transmittance. This technology eliminates any vibrations that may be introduced by an oscillating piezoelectric crystal [84]. While the fundamental technology behind these tunable filters is different, their application achieves the same goal in a similar fashion by iteratively selecting the spectral bands of a subject.



**Figure 11:** Diagram illustrating the optomechanics behind a an Acousto Optical Tuneable Filter [85] The light from the subjet travels through a piezoelectric crystal that alters it's geometry slightly to alter the angle of incidence for each wavelength as it enters the crystal. The crystal is tuned to distort in such a way that only specific wavelengths exit along the 0-order plane toward a monochromatic imaging sensor [64].

Depending on the model of AOTF used, the integration time between the capture of each band can vary based upon the driving frequency of the piezo-optic sensor and the integration time of the camera. One limitation of scanning HSIs is that all of the bands in the datacube cannot be captured simultaneously [86]. During the time between each band capture, a blood cell could rotate or move out of frame causing artifacts between each band in a single datacube. These drawbacks can be overlooked in applications where spatial resolution would provide beneficial information. However, neither of these two types of hyperspectral imaging systems would be applicable for imaging blood flow accurately *in vivo*.

## 2.4.3 Snapshot Hyperspectral Imaging Systems

A HSI Snapshot, or Non-Scanning, system does not suffer from the distortion issues observed in both of the previously mentioned systems. Snapshot HSIs have the capability to capture each band for a single datacube simultaneously, eliminating spatial and temporal distortions by capturing every band on the same sensor [64], [87], [88]. One method of collecting this data simultaneously is to split the incoming image into discrete strips or lines. One method developed, as seen in [18], [87], uses a precisely ground mirror called an image mapper (**Figure 12**) that re-directs linear segments of the incoming image. The result is a collection of input "strips" or lines closely resembling the input for line scan systems (**Figure 13**). However, rather than implementing one prism to diffract the incoming image into individual bands, lens and prism arrays are used to diffract each slice of the incoming image simultaneously. The diffracted slices ultimately project onto a monochromatic sensor where the spatial and spectral information of the image is collected in the same frame grab (integration) of the camera [18], [87]. This design philosophy performs in applications where line and spectral scanning systems cannot (e.g. mobile subjects or unsteady platforms).



**Figure 12:** (left) An image mapper is a precision ground mirrored surface that directs segments of an image [18] (right) into a collecting lens and projects the segmented imnage into a prism array [87].

These systems do suffer from one disadvantage in that the simultaneous capture of both spectral and spatial information limits the spatial resolution of the system. Therefore, in applications that don't involve a subject that may change its shape or direction of motion sporadically, line scan and spectral scanning HSIs will outperform snapshot systems.



**Figure 13:** The large image plane on the right shows the elemental (raw) image data captured by a snapshot HSI. This data consists of both spectral and spatial information that must be recombined to form a datastructure that can be easily interpreted by a user or image processing algorithm. When the scene information is finally captured by the sensor, the data recombined into a datacube via a lookup table created when the system is calibrated.[87]

Other snaphshot HSIs approach this problem by capitalizing as much surface area of the sensor as possible in order to allow for an increase in either spectral or spatial resolution. Rather than using prisms and vibration sensitive optical components, these systems have sensors that have narrow bandwidth filters for each subpixel on the sensor (**Figure 14**) [89], [90]. These subpixels are arranged in a mosaic pattern similar to the Bayer pattern used in standard RGB camera sensors. By filtering and capturing spectral information on the sensor itself, these systems are significantly more compact and have dedicated decoding processors that allow for frame rates above 60Hz [89].



**Figure 14:** Microscope images of a snaphshot hyperspectral imaging sensor with individual band filters over each subpixel of the sensor. This design ensures that every pixel of the sensor is utilized for data capture, a feature that image mapping optical hardware cannot provide [90].

#### 2.4.4 Datacube

The term 'datacube' defines a three dimensional data structure used to store, for this research purpose, hyperspectral image data. Depending on the imaging system and how it records data off of the sensor, the overall sequence or encoding of the data can vary (**Figure 15**). There are three major encoding methods; Band Sequential (BSQ), Band Interleaved by Line (BIL), and Band Interleaved by Pixel (BIP) [91]. Consequentially, the amount of data that is captured and stored within each datacube scales with the number of bands the HSI can capture. Including other factors such as bit-depth and image data compression, hyperspectral datacubes can exceed tens of MB per frame of data captured.

With the emergence of general purpose graphical processing unit (GPGPU) computing, "data heavy" imaging techniques (e.g. Hyperspectral Imaging) which produce substantial amounts of data became practical. Techniques such as hyper-spectral and ultra-spectral imaging systems which can produce a substantial amount of data per frame captured. The advent of GPU powered parallel processing allowed for these imaging systems to capture data at practical integration times. In fact attempts have been made to construct deep neural networks in order to classify blood cells [92]. However, their research did encounter issues with feature extraction as well as cell movement. Unmixing the data based on spectra subtraction, a technique commonly used in GIS surveys, may not prove to increase performance.



Figure 15: Datacube construction per unit time for each HSI capture technique.

## 2.5 Spectral Mixing

When a scene is captured by a HSI, it is highly likely that the scene consists of many objects with different spectral properties. It is safe to assume that not every pixel in the image represents a single object but rather a combination or mixing of different target spectra. The complexity of this mixing varies with respect to the environment of the scene, the distance from the camera each object is and the spatial resolution of the system [93]. Hyperspectral Imagers with high enough spatial resolution relative to the number of target endmembers can capture enough data to calculate the percentage otherwise referred to as abundance ( $\alpha$ ) of each target endmember within a given pixel [94]. This problem presents a multitude of challenges in determining the abundance values of target endmembers within a scene [95]. Many algorithms which approach the problem with a

linear model assume that the abundances of every target endmember are non-negative and that their sum is equal to one [15]. This abundance value is defined by the following.

In a given  $(h \times w \times L)$  hyperspectral image datacube where L is the number of bands in the image and  $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_p$  are *p* known targets (unique objects of interest) present in the image. Let  $\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_p$  denote target signature endmembers and a linear spectral mixture **r** is a combination of  $\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_p$  with appropriate abundance fractions (1).

$$\alpha = \left(\alpha_1 \alpha_2 \dots \alpha_p\right)^T \tag{1}$$

**r** and **M** are represented as an  $L \times 1$  pixel vector and  $L \times p$  target spectral signature matrix denoted by  $[\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_p]$  respectively, where  $\mathbf{m}_j$  is a Lx1 pixel vector for  $1 \le j \le p$  and (1) is a  $p \times 1$ abundance vector associated with **r** and  $\alpha_j$  which denotes the j<sup>th</sup> target signature  $\mathbf{m}_j$  present in pixel vector **r**. The spectral signature of an image pixel **r** can be modeled in a linear regression form as (2), where **n** is noise which can represent sensor or instrument noise.

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{2}$$

This r vector can be used to represent either a pixel vector or spectral endmember as their structure is identical and can be modeled using the equation given in (2).

## 2.6 Hyperspectral Image Preprocessing

Immediately after hyperspectral image data is captured it is transformed by one or a combination of many image processing techniques. This processing step is essential in either compressing data, removing noise, normalizing data, etc. Before any processing techniques are applied, statistical information about the image is collected in a histogram of pixel values [17]. From this histogram, assumptions about the data can be made. For example, if the histogram shows a skewed distribution of pixels toward the max or minimum values, then the image is either over or under exposed.

# 2.6.1 Fast Fourier Transform Filtering



Figure 16: Example of FFT filtering on Lena [99].

The discrete fast Fourier transform (FFT), an optimized version of the Discrete Fourier Transform (DFT) (3), is a commonly used signal processing tool to project a signal from the time domain to the frequency domain. Signal patterns that continue infinitely in the time domain are transformed into finite patterns that can be easily observed and manipulated in frequency space (**Figure 16**). Likewise, signal patterns that occur in an image are often obfuscated by many other signals of different periodicity. A 2 dimensional FFT (4) can return these patterns as discrete structures in the frequency domain. Convolution, an operation which effectively multiplies two signals together in the time domain can be easily achieved with simple multiplication in the frequency domain.

$$X_{k} = \sum_{n=0}^{N-1} x_{n} e^{-\frac{i2\pi kn}{N}}$$
(3)

This equation can be expanded into an infinite number of dimensions which can become a very powerful tool when working with images or in the scope of this research, three dimensional hyperspectral data.

$$\boldsymbol{X}_{\boldsymbol{k}} = \sum_{\boldsymbol{n}=0}^{N-1} e^{-2\pi i \boldsymbol{k} (\frac{\boldsymbol{n}}{N})} \boldsymbol{x}_{\boldsymbol{n}}$$

$$\tag{4}$$

Projecting a filter into the frequency domain of an image, depending on its shape, can act as a low, high or band-pass filter in the time domain. Additionally, unique filters can be applied in the

frequency domain to eliminate specific features that exist in signal (image) in the time (spatial) domain [73], [74], [75].

Since the Snapshot Hyperspectral system captures all data simultaneously, there is no need to be concerned for temporal or spectral distortions to appear as artifacts in the frequency domain. Otherwise, the shifting spatial or spectral information would limit the usefulness of frequency domain operations. From this inherent property of the imaging system, frequency domain filters can be applied to the entire datacube to remove artifacts present in every band.

## 2.6.2 Endmember Construction

In order to properly identify a substance or material, reference spectral information is required. The reference spectra for a given substance is called an endmember. Endmembers are created by collecting purified samples of a known material and collecting the spectral data with precision instruments. Depending on the target spectral bandwidth, more than one sensor instrument may be required to construct an accurate endmember. Once collected, corrections are applied to counter parasitic absorption due to atmospheric or other environmental effects [100].

The need for an accurate endmember is essential when attempting to differentiate materials that are very similar in composition and appearance. In most cases, especially in geological remote sensing, most pixel vector values in a datacube are a mixture of several endmembers [101]. This mixing problem's complexity is also compounded by environmental effects. Segmenting out and identifying specific endmembers from such a complex problem using conventional RGB images is impossible or underdetermined.[102] In order to solve such a complex system, there needs to be more information in the form of measurements (specifically hyperspectral bands) in order to find solutions for specific endmembers.

Several techniques have been researched and developed over the past 20 years, one of the first and simplest is the Pixel Purity Index (PPI) [102]. This algorithm first reduces the dimensionality of

the datacube via the Maximum Noise Fraction (MNF) [103], a technique[104] developed to improve signal to noise ratio of individual bands. This technique gives better results rather than maximize variance in individual bands via principal component analysis (PCA). The MNF a linear transform employs two separate iterations of PCA and noise whitening.

# 2.7 Feature Extraction Techniques

Feature extraction is the process by which data is transformed into a feature space. This feature space has a dimensionality equal to the number of different features extracted from the data. However, as the number of unique features or dimensions increases so too does the required sample size. More features does not always equate to better performance, in fact the very opposite is more likely as the number of features increases. This phenomenon is known as the Hughes phenomenon or more commonly referred to as the "curse of dimensionality" [69]. Therefore, many feature extraction techniques focus to extract only features that provide the greatest statistical weight in feature detection or classification algorithms.

## 2.7.1 Spectral Angle Mapper

The Spectral Angle Mapper(SAM) is a transformation which returns an angle between a probe vector and a library vector (endmember) [105], [106]. The algorithm works by measuring the angle distance between two spectra in n dimensional space, where n is the number of spectral bands in the data. The smaller this angle, the more similar the two spectra are to one another. A powerful feature of this algorithm is that it is insensitive to illumination intensities as the overall "shape" of the spectra is what is compared rather than its actual value [107]. The computation to solve for the spectral angle can determined with equation (5).

$$\alpha = \cos^{-1}\left(\frac{\vec{x} \cdot \vec{r}}{\|\vec{x}\| \cdot \|\vec{r}\|}\right) = \cos^{-1}\left(\frac{\sum_{i=0}^{n-1} X_i \cdot r_i}{\sqrt{\left(\sum_{i=0}^{n-1} X_i^2\right) \cdot \left(\sum_{i=0}^{n-1} r_i^2\right)}}\right)$$
(5)

52

Where *n* is the number of spectral bands in the datacube;  $(\mathbf{X}_i)$  is a probe pixel vector from the datacube;  $(\mathbf{r}_i)$  is the reference endmember and  $\alpha$  is the spectral angle (radians) between the two vectors in *n*-dimensional space (6) (**Figure 17**). Once calculated, the similarity of the probe vector's spectral angle  $\alpha_{\vec{x}}$  and the angle of the reference end can be compared by using the Euclidean distance.

$$\|\alpha_{\vec{x}} - \alpha_{\vec{r}}\| = \sqrt{(\alpha_{\vec{x}} - \alpha_{\vec{r}})^2} \tag{6}$$



**Figure 17:** An example of spectral angle ( $\alpha$ ) and spectral distance (d) [107]

#### 2.7.2 Principal Component Analysis

Principal Component Analysis (PCA) is a commonly used feature extraction algorithm in many signal and image processing applications. PCA is a transformation operation that maps the data into a new feature space. This feature space is based on the variance in the data rather than based on the values of the data. The first dimension is established along an axis in the data with the greatest variance. From there an orthogonal dimension is established again along the axis of greatest variance. This process is repeated until the feature space has an equal number of dimensions as the original data. While the number of dimensions in this feature space is the same as the original data, the feature space is different in that its dimensions are ordered from greatest variance to least variance. Therefore, the data can be best represented by the first several dimensions in the feature space and the rest may be ignored. This pruning operation effectively

reduces the dimensionality in the data while retaining most of the information needed to differentiate classes in the data.

PCA can be performed via a many number of methods but the one implemented in thesis research is primarily eigenvalue decomposition of the data's covariance matrix. For the sake of brevity, this section covers PCA performed on two-dimensional data however, this operation can be expanded to three, four, or n dimensions easily. The covariance matrix of the data is constructed by finding the empirical mean vector  $\boldsymbol{u}[j]$  of every  $j^{\text{th}}$  column in the data (7).

$$\boldsymbol{u}[j] = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{X}[i, j]$$

$$for \, j = 1, \dots, p$$

$$(7)$$

Using the  $1 \times p$  vector  $\boldsymbol{u}[j]$ , the deviations from the mean can be calculated by subtracting the mean vector  $\boldsymbol{u}[j]$  from each row in the data. Which can then be used to calculate the covariance matrix **C**(8).

$$\boldsymbol{B} = \boldsymbol{X} - \boldsymbol{h}\boldsymbol{u}^{T}$$
(8)  
$$\boldsymbol{h}[i] = 1 \qquad \text{for } i = 1, ..., n$$

$$\boldsymbol{C} = \frac{1}{n-1} \boldsymbol{B}^* \cdot \boldsymbol{B} \tag{9}$$

where  $\mathbf{B}^*$  is the conjugate transpose (real transpose) of itself. Once the covariance matrix **C** has been calculated (9), the next step is to find the eigenvalues of the covariance matrix **V**.

$$\boldsymbol{D} = \boldsymbol{V}^{-1} \boldsymbol{C} \boldsymbol{V} \tag{10}$$

The diagonal matrix of eigenvalues **D** is a matrix of size  $p \times p$  and the corresponding eigenvectors in matrix **V** (10). From here, the ordered and paired eigenvalues and eigenvectors can be sorted in descending order based on eigenvalue. The greatest eigenvalue denotes the dimension of most variance followed by the second dimension of most variance and so on. At this point the transformation is complete and these features can be examined and classified [108]. However, it is important to note that while PCA minimized the amount of information lost when reducing dimensionality, there is still some information lost in the process.

## 2.8 Classifiers

#### 2.8.1 Naïve Bayes Classifier

The naïve Bayes classifier is a simple binary classifier that draws a boundary between two classes of data in the feature space of the two classes. This is done for each feature independently and does not rely on relationships between two or more features. This type of classification is a very quick test to determine if a dataset is linearly separable. In which case the need for a more advanced classifier is unnecessary.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
(11)

The target pixel *c* and the predictor vector *x* are evaluated between the probabilities of the predictor P(x) and the probability of the target P(c) and P(x|c), the probability of the predictor given class (11). These three probabilities are used to calculate the posterior probability P(c|x) or the probability the predictor *x* is a member of target class *c*.

#### 2.8.2 Gaussian Maximum Likelihood Classifier

In image classification, pixels in the image are segmented into a finite number to represent entities within the image belonging to classes based on certain rules. In nearly all hyperspectral images, the spectral signature of an image pixel is mixed with a number of target endmembers. This mixing is based on the abundance of a target material(s) in each pixel. This abundance is represented as a scalar fraction and can be estimated by *a priori* or *a posteriori* information.

A powerful extension of the Orthogonal Subspace Projection (OSP) classifier based on a relatively simple assumption that any given pixel vector in the image is likely surrounded by a normal distribution of like pixel vectors. This unconstrained classification algorithm employs an *a priori* approach to maximize the assumed Gaussian probability distribution of an unknown constant which specifies abundance fractions for each target. This estimation combined with both the variance and covariance of the data are used to classify an unknown image pixel [15]. Pixel vector classification is performed by calculating the probability of the target pixel belonging to any given class in the image. The pixel vector is then classified to the class with the highest probability [68].

This classifier, much like many other classifiers, makes several assumptions about the data to be classified. The Gaussian maximum likelihood (GML) classifier assumes that the noise in the data is structured as a Gaussian distribution with mean  $\mathbf{M}\alpha$  and variance  $\sigma^2 \mathbf{I}_{L \times L}$  (12).

$$p(r) = N(\mathbf{M}\alpha, \boldsymbol{\sigma}^2 \mathbf{I}_{LxL}) \tag{12}$$

In particular, the estimate of the  $p^{th}$  abundance value  $\alpha_{\mathbf{r}}$ ,  $\hat{\alpha}_{GML,p}$  is given by the following (13).

$$\hat{\alpha}_{GML,p} = (\mathbf{d}^T P_U^{\perp} \mathbf{d})^{-1} (\mathbf{d}^T P_U^{\perp}) \mathbf{r}$$

$$= (\mathbf{d}^T P_U^{\perp} \mathbf{d})^{-1} \delta_{OSP}(\mathbf{r})$$

$$= (\mathbf{d}^T P_U^{\perp} \mathbf{d})^{-1} \mathbf{d}^T P_U^{\perp} (\mathbf{d}\alpha_p + \mathbf{n})$$

$$= \alpha_p + (\mathbf{d}^T P_U^{\perp} \mathbf{d})^{-1} \mathbf{d}^T P_U^{\perp} \mathbf{n}$$
(13)

Giving an associated estimation error as (14)

$$\varepsilon_{GML,p} = \alpha_{GML,p} - \alpha_p$$

$$= (d^T P_U^{\perp} d)^{-1} \mathbf{d}^T P_U^{\perp} \mathbf{n}$$
(14)

From (13) the GML classifier,  $\delta_{GML}(\mathbf{r})$  can be defined as

$$\delta_{GML}(\mathbf{r}) = (\mathbf{d}^T P_U^{\perp} \mathbf{d})^{-1} \mathbf{d}^T P_U^{\perp} \mathbf{r} = \hat{\alpha}_{GML,p}$$
(15)

56

which gives an unconstrained Gaussian maximum likelihood estimate of an abundance fraction  $\alpha_p$  for a desired target endmember **d** from an image vector **r** [15].

#### 2.8.3 Support Vector Machine

Traditionally, a Support Vector Machine (SVM) is a linear supervised binary classifier which attempts to find the optimal boundary, referred to as a hyperplane, between two classes [109], [110]. Given a dataset of *n* points  $(\vec{x}_1, y_1) \dots, (\vec{x}_n, y_n)$  where  $y_i$  is either -1 or 1, indicating which class  $\vec{x}_i$ belongs to  $(\vec{x}_i \text{ is } p\text{-dimensional})$ . The objective is to determine a hyperplane that will divide classes of points  $\vec{x}_i$  for which  $y_i = 1$  from points  $\vec{x}_i$  for which  $y_i = -1$ . This hyperplane should divide these classes such that the distance from the nearest points  $\vec{x}_i$  of each class is maximized. The hyperplane is defined as (16) where  $\vec{w}$  is the normal to the hyperplane.

$$\vec{w} \cdot \vec{x} + b = 0 \tag{16}$$

The underlying basis of a SVM is the assumption that the classes of  $\vec{x}_i$  (for  $y_i = 1$  and  $y_i = -1$ ) are linearly separable. With this assumption, two parallel hyperplanes (17),(18) can be drawn to separate the data points  $\vec{x}_i$  as much as possible.

$$\vec{\boldsymbol{w}}\cdot\vec{\boldsymbol{x}}+b=1\tag{17}$$

$$\vec{w} \cdot \vec{x} + b = -1 \tag{18}$$

The distance between these two hyperplanes (17),(18)can be defined as  $\frac{2}{\|\vec{w}\|}$ . This distance can be increased if  $\|\vec{w}\|$  is minimized. However, conditions must be applied to these equations such that no point  $\vec{x}_i$  from either class falls into this margin (19).

or,

$$\vec{\boldsymbol{w}} \cdot \vec{\boldsymbol{x}}_i + b \ge 1 \qquad \text{if } y_i = 1$$

$$\vec{\boldsymbol{w}} \cdot \vec{\boldsymbol{x}}_i + b \le -1 \qquad \text{if } y_i = -1$$
(19)

Which can be generalized to (20) for all  $1 \le i \le n$ . The result of this optimization gives the optimization to minimize  $\|\vec{w}\|$  to (20) for i = 1, ..., n, and solve for  $\vec{w} \& b$ .

$$y_i(\vec{w} \cdot \vec{x}_i + b) \ge 1 \tag{20}$$

Additionally, the only points  $\vec{x}_i$  that determine this optimal boundary are the  $\vec{x}_i$  closest to it. These points are referred to as support vecctors. Thus, this classifier requires at least three points (2 from one class and at least 1 from the other) of data from which it draws a line evenly separating the points. The classifier then draws a region extending from the boundary in each direction, with the width of this margin determined by the closest point to the boundary [111]. Any points touching or relatively close to this margin are referred to as support vectors.

This algorithm works off of the assumption that any data not near the boundary between two classes is irrelevant to determining its final position. In fact, it is only the support vectors themselves which encode the information needed to determine an optimal boundary. However, since the support vectors of one class are calculated based on their distance from those of the other class, SVM's are prone to overgeneralization if there is an insufficient number of samples near the boundaries of the two classes. The only way to ensure a SVM does not overgeneralize the training set is to have a substantial amount of useful data. In this context, useful data is defined as accurate data closest to the boundary of the two classes.

### 2.9 Research Tools

The following sections describe tools used in the research conducted in this paper. Other than the Rebellion Photonics Crystal software, these programs are available at no cost or open source. These

programs all have a strong user base and active communities to answer questions and help debug programs.

#### 2.9.1 Rebellion Photonics Crystal

Software provided by Rebellion Photonics which controls, decodes and processes data coming in off of the Arrow Snapshot HSI. This software contains look up tables referencing the exact points on the sensor where each band of the image is located. The software then reconstructs the input back into a three dimensional datacube. Crystal also possesses a flat field correction tool which can be used to normalize the background of the image. This tool is required in order to correctly export the incoming image from the sensor as a band sequential (BSQ) or TIF file as the look up tables for the sensor are proprietary to Rebellion Photonics.

# 2.9.2 Python Packages

Python is a powerful interpreted programming language which is frequently used in the scientific community. Built upon the C programming language, Python possesses little overhead in terms of processing power. It has been largely expanded upon with open source library packages designed for many applications some of which were used in this experiment. A small summary of these packages used is listed below.

#### 2.9.2.1 Numerical Python

Numerical python (NumPy) is a package designed for the python language which completely overhauls the built in mathematic functions. NumPy allows for a wide range of data types capable of being stored in n-dimensional array data structures. This package also includes many matrix operations and naming schemes from other popular scientific analysis software [112]. Installation of this package on Windows machines can be rather difficult when compiling the package from source due to its many dependencies on other open source software libraries. However, a repository of precompiled python libraries exists for windows users to use at their leisure [113] [114].

## 2.9.2.2 Scientific Python

Scientific Python (SciPy) is a package collection that builds off of the NumPy framework that includes optimized functions for scientific computing [115]. SciPy incorporates some other packages that are essential in image processing, rapid prototyping and generating plots for displaying results. While Linux is the preferred operating system for these libraries, it is still possible to deploy them in Windows with slight modifications (Appendix D).

#### 2.9.2.3 SciKit

Sci-kit (SciPy Toolkits) are open source python packages that contain well tested and maintained libraries for scientific computing. The toolkits available range in applications from image processing to cheminformatics [115]. These tools are publicly available and licensed; their use eliminates provides a controlled testing environment so that any variations in the data would be easier to locate and document. The two SciKit modules used for this experiment were SciKit-Image and SciKit-Learn, a powerful image processing toolkit with optimized filtering algorithms and a machine learning toolkit possessing powerful, streamlined learning algorithms that are easy to implement.

#### 2.9.2.4 Spectral Python

Spectral Python (Spectral) is a package designed and developed to effectively process and manage hyperspectral data [116]. This library includes tools that can implement well used classifiers and visualization tools capable of displaying three dimensional HSI datacubes in an easy to understand format. Additionally, this package can perform feature extraction operations on the data to help visualize the 'separability' of data classes in feature space. The tools provided in this package are used to perform data structure manipulations that would otherwise require a large amount of helper functions to make sure data order is preserved.
# 2.9.3 JetBrains PyCharm

PyCharm is a python oriented integrated development environment (IDE) that possesses a powerful python interpreter debugging tool [117]. This debugging tool is essential in tracing errors to their source. It also incorporates helpful package management as well as virtual environment management. This IDE was primarily used to write, test and run the experiments in this thesis work.

# Chapter 3: Imaging System Configuration and Data Capture

## 3.1 Research Objective



**Figure 18:** This flowchart depicts the overall process flow beginning from, configuration and adjustment of the snapshot HSI prior to data capture, the data capture process, followed by inserting data & metadata into a database, and finally hyperspectral image analysis involving feature extraction and blood cell detection.

After analyzing hyperspectral imagery as a potential solution to many of the problems faced in this research, several other options had to then be considered. For instance, while hyperspectral imagery is the process of capturing an image at many continuous spectral bands, the method at which those bands are captured can vary from one device to another. Several imaging systems were considered and compared to see which of them meet the requirements of the research goal.

The next steps involved planning out phases of research to be conducted in order to achieve the target goal (**Figure 18**). One issue of concern is that some microscope objectives have a very narrow depth of field and thus prone to chromatic aberration when imaging subjects in multiple spectral bands. In normal viewing applications this aberration is very difficult to detect with the human eye but becomes blatantly clear in bands toward the edge of the HSI's working bandwidth.

To conform to initial planning phases, several experiments were designed to best simulate conditions seen in real world applications. Experiments began with testing the possibility of segmenting white blood cells in a live blood sample. These experiments were designed to control as many environmental factors as possible to determine which factors impact performance the most. With the most influential factors identified, they can then be given a higher priority in terms of minimizing error.

The initial design of the first experiments was to collect data from an unstained blood slide with as many imaging factors controlled so that a spectral library could be constructed. Roughly 120 samples of blood were prepared at several different cell concentrations and dilution levels. These samples consisted of whole blood and buffy coat smear; at whole, 1:10 and 1:100 blood to Additive Solution (AS) Acid Citrate Dextrose a common solution used to preserve blood tissue dilutions (Hematocrit) [25].

With spectral endmembers of WBC, RBC, background and unknown materials (artifacts from debris or dead tissue) properly constructed, they were then used as target vectors for linear classifiers. A limited amount of preprocessing was done to these endmembers in regards to background removal and illumination normalization. This was intentional because illumination intensity, in real world applications, will constantly be changing and if these methods cannot tolerate variance in illumination intensity then they would not be worth exploring.

First testing endmembers using a naïve Bayes classifier utilizing the Euclidean distance between a probe vector and endmember. With an error rate of (refer to result A) it was determined that this problem was non-trivial and that more advanced segmentation and classification algorithms would be needed. Some further analysis of the data was needed in order to understand which features, if any can be used to reach this research goal.

Analyzing and understanding the data played a critical role in deciding which segmentation and classification techniques to use. With an analysis of the data completed, some feature extraction methods could be implemented. However, feature extraction with hyperspectral data differs from standard color images. In color image processing, color information is usually less important than the overall greyscale intensities of objects in the image. The feature extraction methods performed for this data focus on color (spectral) data over pixel intensities or gradients.

The following sections describe in detail, the data collection process as well as any operations or experiments performed on the data collected. Each section will list descriptions of tools or operations relevant to the corresponding experiment section. A brief analysis of the results will follow reported results and fully summarized in the conclusion section.

# 3.2 Data Capture and Processing

Unlike standard visible imaging sensors, hyperspectral imagery requires the use of filters or prisms to separate narrow bands of light from the input lens and direct these bands onto a sensor. This separation has to occur in such a way that the individual bands do not 'mix' when absorbed by the sensor, otherwise the spectral information of both bands will be sampled into a single value. There is no way to 'undo' this mixing (**Figure 19**) which is why standard RGB images cannot be sliced into more than 3 channels (Red 630-700nm, Green 520-600nm, and Blue 450-520nm).



**Figure 19:** Spectral mixing occurs when the spectral information of two different materials (RBCs and WBCs as seen above) populates the same pixel on the sensor of the HSI. The resulting pixel spectrum is a convolution of the two endmember spectra for each material. Spectral unmixing algorithms find probability distributions that best represent the presence of each target endmember in the pixel [94].

Hyperspectral imagery presents a solution to this problem, if enough spectral bands are collected of known targets (i.e. *a priori* target endmember information) then it is possible to estimate the abundance of each target endmember in a given pixel [17], [102]. These abundance values are probabilities of the presence of each target endmember in a pixel and are essential for multiclass detection algorithms. However, if enough bands in the image are oversaturated (the intensity value for a given subpixel is its maximum value) or under-saturated (the noise in the image overpowers the signal) it is best to discard those bands when calculating abundance values. If there aren't enough useable bands left after this "pruning" operation, the abundance values cannot be determined for the exact same reason abundances cannot be determined in a RGB image [17].

# 3.2.1 HSI System Setup:



**Figure 20:** The Arrow Snapshot Hyperspectral Imaging system developed and manufactured by Rebellion Photonics [18], [118].

The camera system used in this work is the 'Arrow' Snapshot Hyper-spectral system from Rebellion Photonics (**Figure 20**). It was mounted to the camera port of a Leica microscope, supported by an aluminum rig mounted to an optical breadboard table. The Arrow snapshot hyper-spectral imaging system, mounted to the top of a Leica semiconductor inspection microscope is equipped with a Screamin' Eagle motorized stage. This motorized stage is utilized as a means to provide translation of cells through the frame of the snapshot HSI in a simple attempt to replicate blood cells moving through a vessel. The mount structure constructed to support the snapshot hyper-spectral system from t-slot aluminum framing from 80/20. In order to prevent as much motion blur as possible, the entire system was mounted on top of an air-bag suspended optical breadboard table.

Band	Prism 1	Prism 2	Band	Prism 1	Prism 2
1	641.4	-	24	486.9	486.9
2	630.5	-	25	483.0	483.0
3	620.3	-	26	479.3	479.3
4	610.8	-	27	475.5	475.5
5	601.8	-	28	471.9	471.9
6	593.3	-	29	468.3	468.3
7	585.2	-	30	464.8	464.8
8	577.5	-	31	461.4	461.4
9	570.0	-	32	-	458.0
10	562.8	-	33	-	454.7
11	555.9	-	34	-	451.5
12	549.2	-	35	-	448.3
13	542.7	-	36	-	445.2
14	536.3	-	37	-	442.2
15	530.2	-	38	-	439.3
16	524.1	-	39	-	436.5
17	518.4	-	40	-	433.8
18	513.0	-	41	-	431.1
19	508.0	-	42	-	428.5
20	503.3	-	43	-	426.0
21	498.9	-	44	-	423.6
22	494.7	494.7	45	-	421.3
23	490.8	490.8	46	-	419.1

Table 5: The spectral bands for each prism in the Rebellion Photonics Arrow HSI

This camera is controlled via Rebellion Photonics' Crystal software communicating through an Imperx frame grabber card attached to a PC. This snapshot HSI is capable of imaging two bandwidths, with a spatial resolution of 443 x 313px, due to the two separate prisms contained within the device. The first prism is capable of capturing 32 bands ranging from 461nm to 641nm, the other prism captures 25 spectral bands from 419nm to 494nm (**Table 5**). It is attached to a microscope with a motorized stage (**Figure 21**). The Arrow system can capture datacubes at a rate of ~4 cubes/second and an integration time between 30-200µs. The microscope objectives used, magnified at 20x and 50x however, the imaging area of the HSI crops out only a small area, ~20% of the actual field of view from the eyepiece.



**Figure 21:** (A) a snapshot hyperspectral imaging system mounted above a microscope. (B) A diagram depicting the setup used to collect data.

#### 3.2.1.1 Alternative System Considerations and Justifications

An alternative Hyperspectral Imaging system considered was an Acousto-Optical Tunable Filter (AOTF) design distributed by Brimrose Corporation. The Brimrose AOTF spectral scanning imager captures bands by utilizing a piezoelectric tunable band-pass filter [119]. The filter oscillates at the same frequency as the integration rate of the imaging sensor behind it, after a full cycle of bands, a data-cube is constructed and the process repeats [119]. However, one drawback of this type of system is temporal distortions caused by the need for the filter to "sweep" through each band over a certain time interval.

Due to possible issues with distortions caused by integrating the bands over a period of time, The Arrow, a newly developed snapshot hyperspectral system was considered. This system designed and developed by Rebellion Photonics utilizes a multi-faceted mirror and prism[87] to allow the entire data-cube to be captured on the sensor of a camera simultaneously. However, because all the spectral information is captured on the sensor simultaneously, the amount of sensor area is inversely proportional to the number of spectral bands. The limitation from the lack of spatial resolution is diminished by the fact that temporal and spatial distortions no longer needed to be considered a challenge to overcome when imaging blood cells. Consideration of these systems

limitations within the scope of this research led to the decision that a snapshot hyper-spectral system would be the best solution available.

### 3.2.2 HSI calibration and target size estimation

During the installation and set-up of the imaging system, several tests were run to calibrate the imaging software so that the image bands detected on the sensor was accurate. Several precision filters were placed between the illumination source of the microscope and the objective lens. The transmitted light entering the camera was measured and compared to the filter's cutoff wavelength. Settings for the device were adjusted in this manner until the bands observed by the camera were correctly representing the light permitted to enter the objective by the sensor.

With calibration of the device completed, the next experimental variables, magnification and distortion, were measured. It was hypothesized, distortions caused by the microscope objective as well as the crop factor of the imaging system may have an impact on image quality. To test this, a metered glass slide was placed on the microscope stage and to determine if there is any distortion as well as determine the relative physical dimensions of each pixel (**Table 6**) in the image (**Figure 22**). A metered slide with a two dimensional scale (**Figure 23**) was used for measuring pixel dimensions and determine if any barrel distortion occurred [78].



Figure 22: Pixel resolution for x20 (left) and x50 (right) objective lenses calculated using a metered slide



Figure 23: Metered Scale used to detect distortions as well as measure the exact dimensions of the frame

-

1 50

Table o: Pixel	dimensions	of images	from 20x and	1 SOX Objectives

1. C. D.

1 1.

<b>OBJECTIVE LENS</b>	WIDTH	HEIGHT	PIXEL DIMENSIONS
20X	.225 mm	.162 mm	.5um/pixel
50X	.090 mm	.065 mm	.2um/pixel

After several test images were taken, it was determined that there is little to no distortion caused by the optical components of the device. The lines of the scale are known to be respectively parallel so any distortion could be easily identified if the lines were no longer parallel in specific regions of the frame. (**Figure 22**) shows that these lines are parallel to each other across the frame in both dimensions proving there is little to no distortion present in the image. Knowing that the perspective in the image is nearly constant throughout, rough cell dimensions can be determined by using the values calculated in (**Table 6**) above. The importance of this is the spatial information of the cells in the image will not change drastically as they move in and out of frame.

Additionally, there needed to be some way to measure the velocity of the blood cells as they move through the frame. This information can be used to determine the width of a blood vessel as well as the flow volume of the blood within. Values that are necessary to calculate the number of blood cells per unit volume.

The system was also tested to see if the integration time of each frame was fast enough to capture useable data of cells moving through the camera field of view (FOV). This integration time needs to be as short in duration as possible to reduce motion blur. Otherwise, a slow integration time could mean that some cells may also enter and leave the frame without ever being captured by the HSI. To test this limitation, image sequences were captured of the metered slide used to find pixel dimensions translating across the frame at discrete speed increments via an analog controlled motorized stage. An operator then incremented the translation speed as uniformly as possible until the markings on the slide were illegible. Using the known dimensions of the slide's meter, an estimated maximum translation speed

To build a point of reference to compare results of future experiments involving raw blood, several pre-fixed blood slides were procured and imaged. These reference slides contained H&E stain labels to distinguish RBCs and WBCs. The blood slides used were created using blood from an anonymous donor and stained with a standard Hematoxalyn and Eosin (H.E.) stain. The data captured from these slides was used to create datasets that could then be used as a prototype to test classification algorithms.

## **3.2.3 Blood Sample Preparation**

Blood samples, in the form of fixed and sealed histology slides, were obtained from two individuals for the imaging and classification studies. The extracted blood samples in the slide consists of cells collected from the buffy coat of a spun-down blood sample (**Figure 24**). This buffy coat refers to

the narrow band of white blood cells that forms between the blood plasma and red blood cells after spinning a sample in a centrifuge [120].



**Figure 24:** The buffy coat is a thin layer of white blood cells which make up less than 1 percent of a blood sample's total volume [121]

To prepare the sample, an isotonic saline solution was added to delay cell degradation caused by illumination conditions which dry out the sample. Several samples were extracted from the buffy coat [25], [120] and placed onto a glass slide and a cover slip placed on top to spread the sample as thin as possible. This step was implemented to better simulate conditions seen in narrow blood vessels as seen in the sclera of the eye or the inside of the lip. Once several sample slides were prepared, they could then be imaged using the snapshot HSI.

Several sequences of datacubes were captured from different regions of each sample. A prudent effort was made to capture as many different regions within each sample slide to improve variance within the data. While the cells on the slide come from the same individual, each cell has slight differences in terms of overall shape and structure. Some cells may be stacked on top of one another or deformed by groupings of neighboring cells. This variance will help train the system to detect cells on generalized features rather than a feature specific to an example. Once the data collection procedures were completed, post processing of the data needed to be performed in order to easily read data into feature extraction algorithms.

#### 3.2.4 Blood Sample Hyperspectral Data Capture:

To ensure the data capture between different imaging sessions remains as consistent as possible, a data capture process was developed (**Figure 18**). Initially, testing data was captured with different camera sensor settings to minimize noise in the image. The greater the amount of light hitting the sensor per integration period is directly proportional to the intensity values in each band. Testing began by varying one setting while the other remained fixed to see what impacts the changes had on image quality. Finding optimal settings manually was necessary as the Rebellion Photonics Crystal software did not possess an automated exposure control, the integration period and sensor gain were adjusted manually. These tests were to explore which settings had the greatest impact on the signal to noise ratio (SNR) between the target endmembers. The SNR of the image is a ratio of a signal (e.g. pixel vector intensity) compared to the intensity of the noise in the background of the image. The settings that most influenced SNR were the integration time (shutter speed) and the analog gain (sensitivity) of the sensor. These two settings of the Arrow HSI were adjusted when capturing data to ensure SNR was maximized.

The settings that gave the greatest divergence between the two targets (RBC and WBC endmembers, respectively) while keeping less than 25% of the lower energy bands oversaturated. This operation involved selecting target endmember regions of the image using the Crystal software. Once selected, the spectra of the target pixels would be displayed as a two-dimensional plot of the pixel intensity for each band. Next, the integration period of the sensor (i.e. shutter speed) was increased incrementally from 25µs to 200µs until the intensity differential between the two spectra was as great as possible before more than 25% of the bands became oversaturated. Due to the output spectrum of the illumination source, oversaturation always occurred in the lower energy bands (580nm-600nm).

A higher integration time of the sensor in the HSI introduces motion blur into the spatial domain of the datacube. A distortion that is much more challenging to compensate for without the implementation of a coded aperture or de-convolutional algorithm [64]. While increasing the sensor gain introduces noise into the image data, it proved to increase SNR in bands at the higher energy levels (420nm - 450nm). While this change did oversaturate bands in lower energy bands, initial research behind the absorption spectra of blood cells showed local maximum absorption in the higher energy bands. Therefore, higher priority was placed to maximize the intensity differential between the two target endmembers in the regions listed in **Table 4**.



**Figure 25:** Rebellion Photonics Crystal interface. Bands can be selected on the left with a 2d spatial display in the middle. The graph on the top right correlates to the pixel vector selected by the crosshairs in the bottom right image

The data capture process was then designed around increasing SNR. By adjusting the integration time and analog gain of the camera's sensor until SNR couldn't be increased further. However, one drawback of this approach is the settings used to capture data for each sample manually tuned

to capture a useable image. These operations would be equivalent to manual operation of a camera's shutter speed and sensor sensitivity. SNR is then further maximized by reducing background noise using the automated flat field correction (**Figure 26**) tool built into Rebellion Photonics' Crystal software (**Figure 25**).

More specifically, the flat field correction tool (**Figure 26**) detects imperfections in the background of each image by capturing datacubes at different integration times (shutter speeds) and compensates pixel sensitivity accordingly to produce a flat, normalized background image in each spectral band. If any part of the background in the target region changes, artifacts will appear in the datacube. Therefore, initial datacubes were captured with and without flat field calibration in order to determine if these artifacts could cause detrimental effects to segmentation performance. After the first few sessions of data capture,



**Figure 26:** A sample band from a datacube with no flat field correction applied (left). The same sample band in the adjacent image after flat field correction is applied (right).

With the best camera settings to select decided, the Imprex<sup>®</sup> camera integrated into the Arrow HSI was configured to record at the highest capable bit depth of 12 bits per subpixel datacubes. Giving

each subpixel in the datacube a maximum value of 4095. Since data is allocated in bytes, each datacube is given a 16 bit per pixel bit depth but the four greatest significant bits are padded with zeroes. This padding needs to be trimmed via a normalization operation or ignored in analysis software. With bit depth selected, the cubes are stored as a custom binary sequential files (BSQ) with an ISO 8601 [122] timestamp as the filename. The script used to decode these files can be found in **Appendix A**.

A dataset of stained blood cells was captured in 50 individual sequences of datacubes. Each sequence was captured from different regions of 2 unstained blood slides over a duration of 5 seconds. Two complimentary datasets (**Figure 27**) were created one used top down while the other used bottom up illumination. Top down illumination better represents the target application. However, bottom up illumination resulted in images with a higher SNR. The illumination source used is a Schott ACE-1 fiber bundle coupled illumination source [123].



**Figure 27:** Bands from a datacube in stained blood slide dataset. The three images above represent bands that closest resemble the red (left: 471) green (middle: 555) and blue (right: 601) channels of an RGB image:

#### **3.2.5 Dataset Construction and Preparation**

For the data collected for this study, spectral libraries for white blood cells, red blood cells, unknown artifacts, and background were generated by manually masking each class in a sample set of 50 datacubes (**Figure 28**). Cells were masked based on examples provided from hematology related sources as well as the reference slides (3.2.4 Blood Sample Hyperspectral Data Capture:). White cells were labeled green, red cells red, and background was left unlabeled as those locations were interpreted as any unlabeled object remaining in the image. Objects labeled in yellow consisted of glass shards from the microscope slide, dust particles, or other foreign matter that could not be identified.

Bands present in a datacube that are oversaturated (overexposed) have a detrimental impact to determining abundance values for target endmembers [17]. Examples that contained over 75% of useable bands were manually labeled, a python script was written to read in the data and labels; they were then stored into a HDF5 database as float32 values. This datatype is necessary for future steps where this data and its labels will be read back into memory for analysis. Label images were read and stored without alteration however, a second matrix was generated to represent each class as a one-hot binary value.



**Figure 28:** An example of a spectral band used to manually mask data for spectral library construction with one white blood cell labeled in green (left); The same image masked with different RGB values to signify which class the objects belong to (right)

One-hot labels are a fast way to interpret each class as a binary number where each digit represents a class. For example, if an image contained four classes then a four-digit binary number was generated for each pixel. If a pixel was classified to be class A then a one hot representation would be transcribed as '0001', class B '0010', class C '0100' and so on. Storing these one-hot labels at time of dataset creation would reduce overhead when data was read back to perform classification. One-hot encoding is a simple way to label each material as well as incorporate confusion. If the ground truth operator is unsure of a material, they could mark the respective material's bits '1'. For example, regions bordering cell membranes and nuclear membranes could be labeled as A or B '0011' for example. This reduces operator error in masking these cell regions while still providing some useable information for the target pixel vector.

**Table 7:** Statistics Regarding the no. of Samples in Each Data Set

Prism 🔽	Train 🔽 Test	-
Prism 1	138	12
Prism 2	380	30

Before storing the data into an HDF dataset, the training and testing datasets were separated manually (**Table 7**). This was to produce a testing dataset that would consist of data that will never be seen by the classification algorithms during training. Since many of the example images are from very similar imaging conditions, a random sampling of these examples to construct each dataset would likely contain very similar examples. In this scenario a manually generated dataset of mutually exclusive examples would not produce a skewed accuracy score. The objective is to replicate a scenario of how this system can perform on unfamiliar data giving a performance value that is better reflected in real world applications [124].

# 3.2.6 Data Preprocessing

A different approach to labeling data was taken in the construction of this dataset. Instead of labeling white blood cells entirely as one class, they were split into a WBC class and a separate

class for the nucleus in the cells. The colors chosen to represent these classes have changed slightly as well. In the prior dataset, green was used to represent WBCs where as in this dataset, WBCs are labeled in red and RBCs are labeled in green (**Figure 29**).



**Figure 29:** An example showing an extracted band image (507nm) and its manually segmented version used for training and testing.

Once training and testing datasets were selected, they were then read and stored into separate HDF5 files. These datasets were intentionally kept separate so the testing set could be kept on smaller capacity media devices. This would also eliminate any possibility for a training prototype from training on the test data. To improve data access performance, the data was stacked along the spectral axis to create a  $443 \times 313 \times (\lambda N)$  datacube (**Figure 30**). Where b is the number of bands per cube and N is the number of samples in the dataset. This allows for simple loading of the data in 'chunks' into a numpy (See section 2.9.2.1 Numerical Python) array without any kind of data conversion needed.



**Figure 30:** Diagram illustrating the transformations applied to the data. Each cube in the set was rotated and stacked along the spectral axis  $\lambda$  to enable efficient data retrieval during processing and analysis.

Additionally, a third dataset was constructed using the second prism from the Arrow HSI. This dataset consists of data that was collected alongside the prior dataset and constructed in the same fashion. Both of these datasets were trained and tested independently from one another as the bands in each data-cube do not cover the same visible spectrum.



**Figure 31:** Mean pixel vector values for white blood cells and red blood cells with respective standard deviations for both. The black box shows the ROI where a detectable differentiation between cell types is seen.

After collecting and constructing these datasets, the bands residing in the spectrum of interest (Bands 420 - 460 nm in **Figure 31**) as reported in [51], were poorly illuminated. This illumination issue arises from the halogen filament illumination source used [123], its output spectrum is non-linear in the spectral range of the Arrow HIS (**Figure 32**). To capture images that have the best chance to resolve WBCs, it is best to either record the output spectrum of the illuminator or to utilize an illuminator with a normalized output spectrum. Recording the output spectrum of the illumination source allows for background adjusted normalization of the data which can be used to normalize data across all imaging sessions. Alternatively, attenuating the lower energy wavelengths (570 - 750 nm) would be beneficial as the sensitivity of the sensor could be increased to increase intensity in higher energy bands. However, this attenuation filter would need to be incorporated into the background spectrum since this attenuation changes the overall output

spectrum of the illumination source. A reduction of sensor oversaturation in bands with higher light intensities can increase the amount of useful spectral information in the data.



**Figure 32:** Output spectrum of halogen lamp used in Leica Microscope, the red box outlines the output spectrum of the illumination source within the target region of the HSI (inside). The outlined region is further expanded to show the drastic variance in illumination intensity throughout the target spectrum (outside).

From observations of the data, the band at 428nm shows the greatest amount of the natural reflectance from WBCs. **Figure 32** shows the output spectrum of the illumination source used to image the blood slides. The range outlined in the figure directly corresponds to results seen in several medical studies on the optical properties of blood cells [30], [49], [50]. It also shows that the peak intensity does not lie within the active region of the snapshot HSI outlined by the red bounded region. The larger plot in **Figure 32** shows an expanded view of the bounded region from the inside plot. This expanded view shows that even in the active region, the bands of most interest (420 - 440nm) have ~5% of the maximum intensity of the illumination source. To counter this, the

gain of the camera sensor in the HSI was increased. The increased gain caused oversaturation of the imaging sensor in the lower energy bands (475-498nm). This gain increase also introduces more noise into the datacubes, however, despite the increased noise the signal to noise ratio of the data improved. Since the SNR in the ROI bands was increased, it was considered acceptable to leave the lower energy bands oversaturated as the spectral profiles for RBCs and WBCs is too similar [50]. The output spectra of RBCs and WBCs as seen in (**Figure 31**) show that despite the introduced noise, the differentiation between the two cell types is still clearly observable. Although, without using a filter to attenuate the lower energy wavelengths (570-750nm) the sensor becomes oversaturated and spectral information in those bands is lost (**Figure 31**).

#### 3.2.8 Endmember Un-mixing and Construction

Initial methods for analyzing the training data were to separate each pixel vector belonging to each class into a separate matrix. From these matrices, an average pixel vector can be generated as a reference when calculating the spectral angle between it and any given test vector (Eq. 1). Then, a large number of random pixel vectors are selected through the data set. The position of every element in each pixel vector or "skewer" as well as its value is recorded and data points corresponding to extreme values are selected and placed in a list. A counter for each pixel vector element is incremented for each time that the same element appears in the list. Pixels with the highest counter values are considered to be the purest of endmembers found in the datacube. Newer methods build off of this algorithm by artificial selection of skewers to improve efficiency.

# **Chapter 4:** Image Processing & Results

# 4.1 Naïve Bayes Classifier

To begin experimentation, a linear naïve Bayes probabilistic classifier was implemented (see section **2.8.1 Naïve Bayes Classifier**). A naïve Bayes classifier simply treats each feature independently and attempts to separate two classes with a linear boundary in feature space. The target endmember (i.e. boundary) used in this experiment was generated from the manually classified WBC pixel vectors in the training data. As the naïve Bayes classifier is a binary classifier, all other pixel vectors were classified as Non-WBC. The features extracted from the data were the Euclidian distance [125] of each pixel vector to the mean of the target endmembers present in the training data. The other feature was the spectral angle (see section **2.7.1 Spectral Angle Mapper**) distance of each pixel vector from the mean target endmember.

With the average reference vector, two feature extraction methods were applied to a linear segmentation algorithm. To determine whether the Arrow snapshot HSI is capable of capturing enough spectral information needed to detect white blood cells (**Figure 33**).



**Figure 33:** RGB visible image (captured with a microscope mounted Canon 5D mk II) of a region containing three WBCs among a background of RBCs. Note that there is very little contrast between the two types of cells due to spectral mixing of background RBCs between the WBCs and illumination source.

With Euclidean distance and spectral angle features extracted, a linear segmentation algorithm were implemented to test if these features are unique enough to reject all Non-WBC pixel vectors. The segmentation algorithm implemented a naïve Bayes classifier. The boundary condition imposed was if the Euclidean distance of a given pixel vector that fell within 2 standard deviations ( $\sigma$ ) of the target (WBC) pixel vector distributions collected to compute the mean reference vector. If so, the pixel vector was classified WBC (logic 1) as seen in **Figure 35** and **Figure 36**. The same boundary condition was imposed on the SAM feature based classifier except using the spectral angle distribution of the target pixel vectors. Any pixel vector with a spectral angle error within  $2\sigma$  of the mean spectral angle was classified as WBC (**Figure 35** and **Figure 36**).



**Figure 34:** The SAM error values (higher is better) between the pixel vectors of the datacube and the reference vector calculated from the spectral library. The error values used in the calculation from the same example in **Figure 35** on the left, error values from the same example in **Figure 36** are on the right.

Any pixel vector that fell outside that range was classified as non-WBCs (logic 0). When tested against testing datacubes, the Euclidean feature based Bayesian classifier is unable to detect WBCs as seen in (**Figure 36**). The spectral angle of every pixel vector in the data was calculated based on the reference vector. The result was an error value containing the difference angle in n-dimensional space between the two vectors as seen in **Figure 34**.



**Figure 35:** Classification results of image seen in figure 29 by naïve linear classifier classifier (left) and naïve linear classifier applied to SAM features (right)

Once the reference (target endmember) vector was determined from the training data, the classifiers were then tested against datacubes not in the training set (**Figure 34**). Based on results observed in **Figure 35** and **Figure 36**, the spectral angle mapper performs better than the Bayesian classifier due to it being insensitive to overall intensity induced by changes in integration time, sensor gain, or both.



Figure 36: An example of poor performance from raw spectral information compared to SAM features extracted from the spectral information.

While the Bayesian classifier performed well with datacubes from which the reference vector was calculated (**Figure 35**), performance in non-training data drops significantly (**Figure 36**). The performance (**Table 8**) of each classifier was measured by false positive (FPR) and true positive (TPR) rates [126]. The SAM method, with a high TPR of 66.11% and a low FPR of 0.37%, indicates a better performing classifier than the Bayesian classifier with a TPR of 13.74% and a FPR of 0.02%.

No. of Training Samples	Accuracy	Time (s)
120	67.2%	233
100	61.2%	207

**Table 8**: Results from linear Naive Bayes Classifier on datasets manually split into training and testing sets.

#### 4.2 Gaussian Maximum Likelihood Classifier

While the SAM features improved classification performance over Euclidian distance features, there is much room for improvement. Especially since the goal of the proposed system is to provide a blood count, part of which requires the calculation of a ratio of WBCs to RBCs. Thus both types of classes must be classified apart from background or other anomalous structures, which necessitates a multi-class decision boundary. A multiclass approach commonly used in remote sensing called the Gaussian Maximum Likelihood Classifier (GMLC), may perform better than using SAM features (see section **2.8.2 Gaussian Maximum Likelihood Classifier**). To perform the calculations needed for training, a python package called spectral python [116] (**2.9.2.4 Spectral Python**) was used. It contained an efficient GMLC and unit tests to confirm its proper implementation.

#### 4.2.1 Fast Fourier Transform Filtering, PCA, & Gaussian Maximum Likelihood Classifier

With such a high amount of noise and striations present in the data causing performance degradation (**Figure 37**), a de-noising algorithm was developed to remove some of the high frequency, linear artifacts created by the image mapper in the Arrow HSI (see **Figure 12** and section 2.4.3 Snapshot Hyperspectral Imaging Systems). This de-noising operation applied a two dimensional Fast Discrete Fourier Transform (FFT) filtering technique to remove these linear artifacts [87]. This technique applies a filter designed to remove high frequency signals from the image which make up the linear 'striations' in the spatial information of the datacubes.



**Figure 37:** Early tests of GMLC output for an image with background masked to be ignored (top left, top right). Pixel vectors incorrectly classified (bottom)



**Figure 38:** The absolute value or real component of the 2D FFT from a band in the hyperspectral data. The striations seen in the data are represented here as the central vertical line in the image. The mask (right) removes this feature from the frequency information of the FFT and the image is then reconstructed as seen in **Figure 39**.

The 2D FFT of every band in the dataset revealed that there is a repeating pattern of linear artifacts in the image. The linear artifacts are specifically described as the vertical line passing through the center of the image in **Figure 38**. In the Fourier space, linear features are perpendicular to their spatial component [127]. While a Gaussian smoothing operator could be applied to the data, it may remove some useful features in the process. Instead, since the artifacts from the sensor were consistent throughout the datasets, a custom binary mask kernel was applied (mapped) to the data.

To begin the process, hyperspectral datacubes are normalized to a [0,1] range based on the minimum and maximum values in each individual datacube. This normalization, while not necessary for a discrete Fourier transform, helps to take advantage of the added resolution in float64 datatypes as the data is originally stored as float32. The data type conversion is necessary as the FFT algorithm used implements float64 datatypes for the frequency and phase components of the

complex Fourier space result. Once, the FFT is complete, the frequency information is then masked with a generated mask designed to remove high frequency artifacts or 'striations' in the data.

Once the 2D Fourier mask was applied, the inverse 2D FFT was applied, the dataset was reconstructed and stored back into its database. The results presented in **Figure 39**: Un-processed hyperspectral data prior to filtering horizontal striations from the data (left). The same hyperspectral data after FFT filtering (right).**Figure 39** show how well the FFT filtering operation removed the artifacts inherited from the HSIs image mapper.



**Figure 39:** Un-processed hyperspectral data prior to filtering horizontal striations from the data (left). The same hyperspectral data after FFT filtering (right).

While there appeared to be visible improvements in image quality as these uniform striations may introduce an increased error rate in segmentation and ultimately classification performance, the filtering operation could have removed statistically significant features in the spectral data. The only way to test if this operation is beneficial is to apply the FFT filtering operation before the feature extraction of the GMLC is applied. To explore if there are any superfluous bands in the data such as bands that are oversaturated or spectral features of the classes present are identical in two or more bands. Much like the manual band rejection applied to the linear classifiers in the previous section. A principal component analysis (PCA) was applied to the training set (see **Figure 40** section **2.7.2 Principal Component Analysis**).



**Figure 40:** The covariance matrix after applying PCA to the training dataset used in the experiment illustrated in **Figure 37** excluding background (Non-RBC & Non-WBC) features.

A visualization tool seen in **Figure 41**, shows the components as a labeled point cloud in a 3 dimensional scatter plot. These plots are difficult to interpret from the figures provided (as they are a 2D projection of a 3D image). However, the contribution they provide is for the analyst to determine if there is indeed separability between the classes present in the data.



Figure 41: PCA components of raw (left) and FFT processed data (right). Classes are represented by their respective colors shown in Figure 42-45

The first feature set used was only the normalized datacubes with no filtering applied, this dataset was passed into the GMLC before (**Figure 42**) and after passing only the most significant PCA components (top 40%, or ~10-12 'bands') to the classifier. The results showed a reduction in sparse misclassifications and decreased the number of errors along the boundaries of cell membranes (**Figure 43**). However, some regions of background had a higher distribution of misclassification than others giving only a 2-3% increase in accuracy. As the data tested possessed a low SNR, it is unclear whether the PCA components provide any significant improvement to multiclass performance without training on a higher quality dataset. Despite this result, the Gaussian classifier performed very well compared to prior linear classification experiments. With mean percentage accuracy ranging from 65-89% based on background exclusion and dimensionality reductions performed prior to training as well as mathematical morphology operations [128] performed after testing to reduce overall misclassification error as much as possible.



Figure 42: Results of Gaussian Classifier on raw, unprocessed hyperspectral data.



Figure 43: Results of Gaussian Classifer on PCA features of raw, unprocessed hyperspectral data



Figure 44: Results of Gaussian classifer on FFT filtered hyperspectral data.



Figure 45: Results of Gaussian Classifer on PCA features of FFT filtered data


**Figure 46:** Observable improvements in classification performance of using PCA selected spectral features from the raw hyperspectral data (left) and the FFT filtered hyperspectral data (right)



**Figure 47:** Application of a erosion (a mathematical morphology algorithm) to the results to clean up sparse errors in the data (left). Compared to the manually generated ground truth (right) as well as results seen in **Figure 46**.

The FFT filtering technique was then applied to the data and the prior two experiments were repeated (**Figure 44**). Again, PCA components appeared to have reduced the number of sparse misclassifications and increased accuracy roughly 2-3% (**Figure 45**). However, the smoothing operation from the FFT filtering reduced a large number of linear regions of misclassification caused by artifacts from the Arrow HSI's image mapper. The error (see **Table 9**) decreased noticeably around the boundaries of the cell membranes of blood cells which may allow for further increased accuracy performance by removing sparse misclassifications around blood cells.

One final operation that may be applied to the results is an image manipulation technique called erosion (**Figure 47**). The erosion algorithm looks for collections of neighboring pixels that have the same value (label) and removes the outermost layer of pixels from the image. The result is a removal of sparsely distributed pixels which have a high likelihood of being erroneous, while keeping the larger neighborhoods of pixels relatively untouched. **Figure 46** depicts PCA and PCA & FFT GMLC classification results for a dataset that was generated using a second dataset where datacubes with a poor SNR (below 10dB) were excluded from their respective training and testing sets. This lower SNR threshold limit increased accuracy percentages by ~10% which demonstrates how proper illumination and image capture settings can give a significant reduction in classification errors. **Figure 47** shows an erosion operation applied to the PCA & FFT GMLC result, removing a majority of sparsely misclassified pixel vectors and increasing average total accuracy to 89%

Accuracy % of GMLC and Data trans	formations
GMLC Raw	70%-74%
GMLC w/ PCA	70%-72%
GMLC w/ FFT	71%-77%
GMLC w/ FFT & PCA	75%-81%
Cleaned GMLC w/ FFT & PCA	84%-89%

**Table 9:** Error rates for each data transformation applied during training for blood cell datasets

# **Chapter 5:** Conclusions & Future Work

## 5.1 Results Summary and Conclusions

The contributions from this thesis work include:

- 1. Constructed an HSI dataset of unstained blood samples on sealed blood slides.
- From experimental data, found a small region of spectral bands in the visible spectrum (420-440nm) in which WBCs and RBCs show a significant difference in absorptivity.
  - a. This region is consistent with results from other works regarding the optical properties of blood cells [30], [49], [50].
- Demonstrated that a snapshot HSI is capable of capturing spectral information RBCs and WBCs in this narrow region.
  - a. Implemented methods used in GIS and other HSI applications to show this region is significant enough to differentiate RBCs and WBCs.
- 4. Illustrate the potential of HSI systems to detect unstained blood cells compared to standard RGB methods which rely on the use of stain to improve contrast.

The use of spectral features of blood and other tissues is steadily becoming an applicable technology for medical professionals as a diagnostic tool for tissue damage and cancer detection. While a few technologies have used some of these spectral features to monitor blood oxygenation and blood flow but, no current systems have used this property for *in vivo* blood cell counting. Blood cells have shown to have a unique spectral response in the low energy ultraviolet spectrum  $(320 \pm 20 \text{nm})$  and the high energy visible spectrum  $(420 \pm 15 \text{nm})$ . A spectral response that directly correlates to findings in medical studies on the optical properties of blood [30], [49], [50]. With these properties, the goal of this thesis research was to exploit one or both of these properties to detect blood cells by type (e.g. WBC RBC).

With a collection of 150 unique examples captured from unstained blood slides of two subjects, sealed on a glass slide using in three diluted concentrations (1:1, 1:10 and 1:100 hematocrit). A training subset of 90 examples was used to extract spectral features and classify the remaining

examples outside the training subset. An image processing workflow has been established using mostly open source software, for example, python, numpy, scipy, scikit, spectral python, other python packages, HDF5, and PyCharm Community Edition.

The experiments in this paper involved the steps developed and explored exploiting a natural phenomenon (e.g. spectral absorption) of blood to detect WBCs. Collection experiments were developed to capture sealed and unstained blood samples in several different concentrations and illumination scenarios. A novel imaging approach using a snapshot HSI was deployed to collect and gather data of these blood on slides to test this hypothesis. Hyperspectral image processing and analysis methods were applied to the data in order to determine whether the spectral and spatial features of each cell type could be extracted and used to differentiate blood cell types.

The results of these experiments showed blood cell classification accuracy ranging 65-92% (see **Table 10-15**) based on the types of image processing techniques applied to the data before and after classification. With the FFT filtering method increasing classification accuracy to 78% (**Figure 48**) and applying morphological erosion (**Figure 47**) to the results after classification can also increase performance to 90% (**Figure 49**). This performance boost is likely caused by removing noise caused by the HSI's image mapper and spectral mixing influencing the GMLC to misclassify sparse groupings of pixels. Future experiments should test these techniques on a deterministic classifier rather than a stochastic classifier like the GMLC. A deterministic model may help reduce classifier error as the intensities of target endmembers in the image may not vary enough for a probabilistic model to predict.

Raw GMLC Performance					
Confusion	Matrix:				
3102	0	0	0	0	
0	1902	173	571	704	
0	465	13461	1547	2529	
0	599	175	1755	791	
0	9108	11973	14654	75150	
Classificat	ion Report				
	precision	recall	f1-score	support	
null	1.00	1.00	1.00	3102	
WBC	0.16	0.57	0.25	3350	
RBC	0.52	0.75	0.61	18002	
NUC	0.09	0.53	0.16	3320	
BKGND	0.95	0.68	0.79	110855	
avg / total	0.86	0.69	0.74	138659	
Classificat	ion accura	c <b>y:</b>		0.677802	

Table 10: Pixel clasification results using raw HSI data

Table 11: Pixel classification results after applying FFT filter to raw HSI data

<b>FFT Filt</b>	ering G	MLC Pe	rforma	nce
Confusion	Matrix:			
3102	0	0	0	0
0	2044	134	533	619
0	354	14357	1480	1811
0	570	132	1969	649
0	5605	8737	9581	86962
Classificat	ion Report			
	•			
	precision	recall	f1-score	support
null	precision 1.00	recall 1.00	f1-score 1.00	support 3102
null WBC	precision 1.00 0.24	recall 1.00 0.61	f1-score 1.00 0.34	support 3102 3350
null WBC RBC	precision 1.00 0.24 0.61	recall 1.00 0.61 0.80	f1-score 1.00 0.34 0.69	support 3102 3350 18002
null WBC RBC NUC	precision 1.00 0.24 0.61 0.14	recall 1.00 0.61 0.80 0.59	f1-score 1.00 0.34 0.69 0.23	support 3102 3350 18002 3320
null WBC RBC NUC BKGND	precision 1.00 0.24 0.61 0.14 0.97	recall 1.00 0.61 0.80 0.59 0.78	f1-score 1.00 0.34 0.69 0.23 0.87	support 3102 3350 18002 3320 110885
null WBC RBC NUC BKGND avg / total	precision 1.00 0.24 0.61 0.14 0.97 0.88	recall 1.00 0.61 0.80 0.59 0.78 0.78	f1-score 1.00 0.34 0.69 0.23 0.87 0.82	support 3102 3350 18002 3320 110885 138659
null WBC RBC NUC BKGND avg / total	precision 1.00 0.24 0.61 0.14 0.97 0.88	recall 1.00 0.61 0.80 0.59 0.78 0.78	f1-score 1.00 0.34 0.69 0.23 0.87 0.82	support 3102 3350 18002 3320 110885 138659

PCA prior to GMLC					
Confusion	Matrix:				
3102	0	0	0	0	
0	1904	176	550	720	
0	475	13648	1500	2379	
0	602	176	1719	823	
0	9122	12224	14310	75229	
Classificat	ion Report	•			
	-				
	precision	recall	f1-score	support	
null	precision 1.00	recall 1.00	f1-score 1.00	support 3102	
null WBC	precision 1.00 0.16	recall 1.00 0.57	f1-score 1.00 0.25	support 3102 3350	
null WBC RBC	precision 1.00 0.16 0.52	recall 1.00 0.57 0.76	f1-score 1.00 0.25 0.62	support 3102 3350 18002	
null WBC RBC NUC	precision 1.00 0.16 0.52 0.10	recall 1.00 0.57 0.76 0.52	f1-score 1.00 0.25 0.62 0.16	support 3102 3350 18002 3320	
null WBC RBC NUC BKGND	precision 1.00 0.16 0.52 0.10 0.95	recall 1.00 0.57 0.76 0.52 0.68	f1-score 1.00 0.25 0.62 0.16 0.79	support 3102 3350 18002 3320 110885	
null WBC RBC NUC BKGND avg / total	precision 1.00 0.16 0.52 0.10 0.95 0.86	recall 1.00 0.57 0.76 0.52 0.68 0.69	f1-score 1.00 0.25 0.62 0.16 0.79 0.75	support 3102 3350 18002 3320 110885 138659	
null WBC RBC NUC BKGND avg / total	precision 1.00 0.16 0.52 0.10 0.95 0.86	recall 1.00 0.57 0.76 0.52 0.68 0.69	f1-score 1.00 0.25 0.62 0.16 0.79 0.75	support 3102 3350 18002 3320 110885 138659	

**Table 12**: Pixel classification results after applying principal component analysis to raw HSI data

**Table 13**: Pixel classification results after applying principal component analysis

 and FFT filtering to raw HSI data

PCA &	FFT filte	er prior	to GML	С
Confusion	matrix:			
3102	0	0	0	0
0	2048	136	552	614
0	341	14458	1482	1721
0	567	148	1934	671
0	5564	8724	9369	87228
Classificat	ion Report	•		
	precision	recall	f1-score	support
null	1.00	1.00	1.00	3102
null WBC	1.00 0.24	1.00 0.61	1.00 0.35	3102 3350
null WBC RBC	1.00 0.24 0.62	1.00 0.61 0.80	1.00 0.35 0.70	3102 3350 18002
null WBC RBC NUC	1.00 0.24 0.62 0.15	1.00 0.61 0.80 0.58	1.00 0.35 0.70 0.23	3102 3350 18002 3320
null WBC RBC NUC BKGND	1.00 0.24 0.62 0.15 0.97	1.00 0.61 0.80 0.58 0.79	1.00 0.35 0.70 0.23 0.87	3102 3350 18002 3320 110885
null WBC RBC NUC BKGND avg / total	1.00 0.24 0.62 0.15 0.97 0.88	1.00 0.61 0.80 0.58 0.79 0.78	1.00 0.35 0.70 0.23 0.87 0.82	3102 3350 18002 3320 110885 138659
null WBC RBC NUC BKGND avg / total	1.00 0.24 0.62 0.15 0.97 0.88	1.00 0.61 0.80 0.58 0.79 0.78	1.00 0.35 0.70 0.23 0.87 0.82	3102 3350 18002 3320 110885 138659

PCA &	FFT GM	LC aftte	er erosio	on
Confusion	matrix:			
3102	0	0	0	0
815	1652	35	234	614
2080	3	13501	697	1721
989	282	5	1373	671
17242	203	4177	2035	87228
Classificat	ion Report	•		
	-			
	precision	recall	f1-score	support
null	precision 0.13	recall 1.00	f1-score 0.23	support 3102
null WBC	precision 0.13 0.77	recall 1.00 0.49	f1-score 0.23 0.60	support 3102 3350
null WBC RBC	precision 0.13 0.77 0.76	recall 1.00 0.49 0.75	f1-score 0.23 0.60 0.76	support 3102 3350 18002
null WBC RBC NUC	precision 0.13 0.77 0.76 0.32	recall 1.00 0.49 0.75 0.41	f1-score 0.23 0.60 0.76 0.36	support 3102 3350 18002 3320
null WBC RBC NUC BKGND	precision 0.13 0.77 0.76 0.32 0.97	recall 1.00 0.49 0.75 0.41 0.79	f1-score 0.23 0.60 0.76 0.36 0.87	support 3102 3350 18002 3320 110885
null WBC RBC NUC BKGND avg / total	precision 0.13 0.77 0.76 0.32 0.97 0.90	recall 1.00 0.49 0.75 0.41 0.79 0.77	f1-score 0.23 0.60 0.76 0.36 0.87 0.82	support 3102 3350 18002 3320 110885 138659
null WBC RBC NUC BKGND avg / total	precision 0.13 0.77 0.76 0.32 0.97 0.90	recall 1.00 0.49 0.75 0.41 0.79 0.77	f1-score 0.23 0.60 0.76 0.36 0.87 0.82	support 3102 3350 18002 3320 110885 138659

**Table 14**: Pixel classification results after applying erosion technique to resultsfrom Table 13

**Table 15**: Pixel classification results after ignoring removed pixels from resultsin Table 14

PCA & FF	T GMLC Ei	rosion aft	er ignorin	g remove
Confusion	matrix:			
24228	0	0	0	0
0	1652	35	234	614
0	3	13501	697	1721
0	282	5	1373	671
0	203	4177	2035	87228
Classificat	ion Report	• • • •		
	precision	recall	f1-score	support
null	precision 1.00	recall 1.00	f1-score 1.00	support 24288
null WBC	precision 1.00 0.77	recall 1.00 0.65	f1-score 1.00 0.71	support 24288 2535
null WBC RBC	precision 1.00 0.77 0.76	recall 1.00 0.65 0.85	f1-score 1.00 0.71 0.80	support 24288 2535 15922
null WBC RBC NUC	precision 1.00 0.77 0.76 0.32	recall 1.00 0.65 0.85 0.59	f1-score 1.00 0.71 0.80 0.41	support 24288 2535 15922 2331
null WBC RBC NUC BKGND	precision 1.00 0.77 0.76 0.32 0.97	recall 1.00 0.65 0.85 0.59 0.93	f1-score 1.00 0.71 0.80 0.41 0.95	support 24288 2535 15922 2331 93643
null WBC RBC NUC BKGND avg / total	precision 1.00 0.77 0.76 0.32 0.97 0.93	recall 1.00 0.65 0.85 0.59 0.93 0.92	f1-score 1.00 0.71 0.80 0.41 0.95 0.93	support 24288 2535 15922 2331 93643 138659
null WBC RBC NUC BKGND avg / total	precision 1.00 0.77 0.76 0.32 0.97 0.93	recall 1.00 0.65 0.85 0.59 0.93 0.92	f1-score 1.00 0.71 0.80 0.41 0.95 0.93	support 24288 2535 15922 2331 93643 138659



**Figure 48:** Accuracy metrics of the Gaussian Maximum Likelihood classifier based on the several pre-processing techniques applied to the data prior to classification.



**Figure 49:** Accuracy metrics of Gaussian Maximum Likelihood Classifier after the erosion post processing techniques were applied to the classification results.

It should be noted that there is an inherent bias in the data due to a limited number of subjects (2) from which blood samples were collected. This bias, while decreasing the statistical significance of these results, allows these results to demonstrate a HSI's promising capability to detect and segment blood cell types. The creation of a robust dataset of unstained blood samples is required in order to validate these results

This thesis research demonstrated that it is possible to differentiate blood cells based on their absorptivity without the need for a label or stain. The methods and experiments designed and implemented in this research have helped to lay out a foundation to further develop a system that may perform this task *in vivo*. These tests were conducted on datasets captured using two different spectral bandwidths that the Arrow HSI is capable of capturing within. This helped demonstrate how important the bands within the range of 420 – 440nm are to differentiating between RBCs and WBCs. Within this range, the difference between WBCs and RBCs is at its greatest and it would be prudent to design a HSI that capitalizes on this spectrum when moving forward in this research.

# 5.2 Future Work

### 5.1.1 Deep Learning Neural Network

One downside of previous experiments was that the features extracted were based on a heuristic approach. One added benefit of artificial neural network (NN) learners is that they extract features based on experiences from the training data thus a non-heuristic approach is taken to extract optimal features for the data observed (**Figure 50**).



**Figure 50:** Classification results from a Fully Connected Neural Network trained from roughly 30 unique sample data-cubes. Of which multiple data-cubes were captured over a period of 20 seconds. This result is from a single hidden layer fully connected neural network. A more advanced network, improved preprocessing and most imporantly a large dataset may surpass performance reported in this work.

The purpose of utilizing this neural network is to find any performance gain from using nearly all the examples in the data set rather than an extremely small percentage. Neural networks require a large amount of data and processing power to compute. With the samples collected in preliminary testing of the system there are not enough examples to construct a viable network that exceeds the performance of other methods tested. With a more robust data collection from many subjects, Artificial Neural Networks are worth exploring.

# 5.1.2 Improvements in snapshot HSI technology

Based on the discoveries made in this thesis research, a system capable of detecting and counting white blood cells *in vivo* is possible. Although given the current equipment in this research that goal is very challenging. To improve the performance and viability of this system, a snapshot HSI that is capable of capturing data-cubes at a higher frame rate. As well as a system with a smaller footprint so that these systems could be made semi-portable [10], [90].

Bay-Spec Corporation has recently developed a snapshot hyperspectral imager that has a small footprint and high frame rate. This device vastly surpasses the capabilities of the [11] Arrow system and it could be used for further study as well as tracking cells in high flow scenarios. Another system designed and manufactured by Ximea Inc, is a system on chip (SoC) solution that can filter up to 25 bands on the sensor itself, eliminating any need for expensive and vibration sensitive optical hardware [90].

Other improvements would be to gather and image blood samples from a small population of individuals. This will test if these optical features are common to all cells in different subjects. Also, one future problem to overcome is commonly referred to as the 3D problem. It refers to the depth at which blood cells travel within the tissue as the surrounding blood vessels and tissue layers will affect the spectral information recorded by the camera.

Alternative sites on the human body can also be used to image blood cells as they move through blood vessels. One site in particular is the vascular tissue on the inside of the lip otherwise known as the labial (oral) mucosa (**Figure 51**). These vessels in this tissue are very superficial and would provide a valuable alternative location if perhaps imaging scleral vessels in some subjects would be too difficult (damaged or missing eye).



**Figure 51:** Illustration of the Buccal (labial) mucosa, the regioun behind th elip which is very vascular and is another potential imaging site.

This effect also refers to the nature of blood cells as they travel through a vessel, as white cells tend to stick to the sides of vessels and red cells flow over one another. Spectral un-mixing of these tissues will pose a challenge as they all contain nuclear material that may reduce the view-ability of the WBCs.



**Figure 52:** Illustration Showing Limitations of non-specialized Microscope Objectives, note how the Depth of Field (DoF) only occupies a small volume of the capillary tube. Also not the focal plane of the objective widens as the distance from the focal point or working distance.

There is also an issue known as the 3D problem in which the object or in this case a blood vessel, does not have a depth of field deep enough to capture all objects in focus (**Figure 52**). Chromatic aberration of the microscope objectives also plays a role in distorting spectral data. To counter this, achromatic objectives can be used since they increase the bandwidth of wavelengths that will be focused correctly on the sensor of the system.

The design and construction of an endoscope system to attach a lightweight adjustable objective to the snapshot HSI is the next step before live subject testing could be conducted. This endoscope would allow for relatively simple image collection which would otherwise be impractical due to the large size and weight of the Arrow HSI (**Figure 53**). Alternatively, a newer model snapshot HSI could replace the Arrow system [46].



**Figure 53:** Example diagram demonstrating the desired system setup for performing on site complete blood counts.

With an endoscope system developed and tested with no detriment to system performance, the path toward live subject data collection and testing would be the next logical step. To build a system that can perform rapid label-less blood counts for any individual would require a significant amount of data. Data collected from a diverse group of subjects as well as complete blood counts performed on samples extracted prior to imaging as a control. The data and information collected should be

enough to build a spectral endmember library of all blood cells in the body. With a next generation snapshot HSI sensor, a methodical and organized sample collection. This system will have a potential to provide a simple, automated in vivo blood counting system that may protect lives from danger or save lives from disease.

# References

- [1] R. L. Cecil, L. Goldman, and A. I. Schafer, Eds., *Goldman's Cecil medicine*, 24th ed. Philadelphia: Elsevier/Saunders, 2012.
- [2] R. A. McPherson and M. R. Pincus, *Henry's Clinical Diagnosis and Management by Laboratory Methods*, 22nd ed. Elsevier Health Sciences, 2011.
- [3] B. M. Singal, "Point-of-care Blood Testing and Cost-effective Emergency Medicine," *Acad. Emerg. Med.*, vol. 2, no. 3, pp. 163–164, 1995.
- [4] G. Garty et al., "THE RABIT: A RAPID AUTOMATED BIODOSIMETRY TOOL FOR RADIOLOGICAL TRIAGE," Health Phys., vol. 98, no. 2, pp. 209–217, Feb. 2010.
- [5] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, Dec. 2015.
- [6] Y. Yang, Z. Zhang, X. Yang, J. H. Yeo, L. Jiang, and D. Jiang, "Blood cell counting and classification by nonflowing laser light scattering method," *J. Biomed. Opt.*, vol. 9, no. 5, p. 995, 2004.
- [7] F. Sadeghian, Z. Seman, A. R. Ramli, B. H. A. Kahar, and M.-I. Saripan, "A Framework for White Blood Cell Segmentation in Microscopic Blood Images Using Digital Image Processing," *Biol. Proced. Online*, vol. 11, no. 1, p. 196, Jun. 2009.
- [8] International Atomic Energy Agency, Ed., *The Radiological accident in Goiânia*. Vienna : [Lanham, MD: International Atomic Energy Agency; UNIPUB, distributor], 1988.
- [9] Q. Li, C. Dai, H. Liu, and J. Liu, "Leukemic cells segmentation algorithm based on molecular spectral imaging technology</title&gt;," 2009, p. 73830V-73830V-7.
- [10] BaySpec Corporation, "OCI 2000 Handheld Snapshot Hyperspectral Imager." .
- [11]C. Dai, Q. Li, and J. Liu, "Blood Cells Classification Using Hyperspectral Imaging Technique." [Online]. Available: http://www.academicpub.org/jbbe/paperInfo.aspx?paperid=15161. [Accessed: 08-Oct-2015].
- [12]Q. Li et al., "Leukocyte cells identification and quantitative morphometry based on molecular hyperspectral imaging technology," Comput. Med. Imaging Graph., vol. 38, no. 3, pp. 171– 178, Apr. 2014.
- [13]S. Crihalmeanu and A. Ross, "Multispectral scleral patterns for ocular biometric recognition," *Pattern Recognit. Lett.*, vol. 33, no. 14, pp. 1860–1869, Oct. 2012.
- [14]I. Grulkowski, J. J. Liu, B. Baumann, B. Potsaid, C. Lu, and J. G. Fujimoto, "Imaging limbal and scleral vasculature using Swept Source Optical Coherence Tomography," *Photonics Lett. Pol.*, vol. 3, no. 4, Dec. 2011.
- [15]C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*. New York: Kluwer Academic/Plenum Publishers, 2003.
- [16]F. Gruber, P. Wollmann, B. Schumm, W. Grählert, and S. Kaskel, "Quality Control of Slot-Die Coated Aluminum Oxide Layers for Battery Applications Using Hyperspectral Imaging," *J. Imaging*, vol. 2, no. 2, p. 12, Apr. 2016.
- [17]B. Park and R. Lu, Eds., *Hyperspectral Imaging Technology in Food and Agriculture*. New York, NY: Springer New York, 2015.
- [18] R. T. Kester, N. Bedard, L. Gao, and T. S. Tkaczyk, "Real-time snapshot hyperspectral imaging endoscope," J. Biomed. Opt., vol. 16, no. 5, pp. 56005–56005, 2011.
- [19]BD Diagnostics, "Best Sites for Venipuncture." Becton, Dickinson and Company.
- [20] R. D. (RN.), Delmar's Manual of Laboratory and Diagnostic Tests. Cengage Learning, 2003.
- [21] A Review of Human Carcinogens: Chemical Agents and Related Occupations, vol. Volume 100F. 2012.

- [22]F. C. Dimeo, R.-D. Stieglitz, U. Novelli-Fischer, S. Fetscher, and J. Keul, "Effects of physical activity on the fatigue and psychologic status of cancer patients during chemotherapy," *Cancer*, vol. 85, no. 10, pp. 2273–2277, 1999.
- [23] R. Bologna-Molina, P. Damián-Matsumura, and N. Molina-Frechero, "An easy cell counting method for immunohistochemistry that does not use an image analysis program: Correspondence," *Histopathology*, vol. 59, no. 4, pp. 801–803, Oct. 2011.
- [24] Abbott Laboratories, "Abbott CELL-DYN Sapphire Hematology Datasheet." Abbott Laborotories.
- [25]R. I. Handin, S. E. Lux, and T. P. Stossel, Eds., *Blood: principles and practice of hematology*, 2nd ed. Philadelphia, Pa: Lippincott Williams & Wilkins, 2003.
- [26] V. P. Zharov, E. I. Galanzha, and V. V. Tuchin, "Photothermal image flow cytometry in vivo," Opt. Lett., vol. 30, no. 6, p. 628, Mar. 2005.
- [27]K. Larson, H. H. Ho, P. L. Anumolu, and T. M. Chen, "Hematoxylin and eosin tissue stain in Mohs micrographic surgery: a review," *Dermatol. Surg. Off. Publ. Am. Soc. Dermatol. Surg. Al*, vol. 37, no. 8, pp. 1089–1099, Aug. 2011.
- [28] V. P. Zharov, E. I. Galanzha, and V. V. Tuchin, "In vivo photothermal flow cytometry: imaging and detection of individual cells in blood and lymph flow," *J. Cell. Biochem.*, vol. 97, no. 5, pp. 916–932, Apr. 2006.
- [29]R. V. Kuranov *et al.*, "Depth-resolved blood oxygen saturation measurement by dualwavelength photothermal (DWP) optical coherence tomography," *Biomed. Opt. Express*, vol. 2, no. 3, p. 491, Mar. 2011.
- [30]N. Bosschaart, G. J. Edelman, M. C. G. Aalders, T. G. van Leeuwen, and D. J. Faber, "A literature review and novel theoretical approach on the optical properties of whole blood," *Lasers Med. Sci.*, vol. 29, no. 2, pp. 453–479, Mar. 2014.
- [31]D. Izhaky, D. A. Nelson, Z. Burgansky-Eliash, and A. Grinvald, "Functional imaging using the retinal function imager: Direct imaging of blood velocity, achieving fluorescein angiography-like images without any contrast agent, qualitative oximetry, and functional metabolic signals," *Jpn. J. Ophthalmol.*, vol. 53, no. 4, pp. 345–351, Jul. 2009.
- [32]R. W. DeBlois, "Counting and Sizing of Submicron Particles by the Resistive Pulse Technique," *Rev. Sci. Instrum.*, vol. 41, no. 7, p. 909, 1970.
- [33] Beckman Coulter, "Coulter Principle Short Course." Beckman Coulter.
- [34]D. Heikali and D. Di Carlo, "A Niche for Microfluidics in Portable Hematology Analyzers," J. Assoc. Lab. Autom., vol. 15, no. 4, pp. 319–328, Aug. 2010.
- [35]P. Liu et al., "Parallel processing of massive remote sensing images in a GPU architecture," *Comput. Inform.*, vol. 33, no. 1, pp. 197–217, 2014.
- [36] J. R. Eastman, *IDRISI Kilimanjaro: guide to GIS and image processing*. Clark Labs, Clark University Worcester, 2003.
- [37]R. Trobec and G. Kosec, *Parallel Scientific Computing*. Cham: Springer International Publishing, 2015.
- [38]P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A Survey on Spectral-Spatial Classification Techniques Based on Attribute Profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.
- [39]Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral Image Classification via Kernel Sparse Representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.
- [40]J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral Image Segmentation Using a New Bayesian Approach With Active Learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.
- [41]A. Dacal-Nieto, A. Formella, P. Carrión, E. Vazquez-Fernandez, and M. Fernández-Delgado, "Common scab detection on potatoes using an infrared hyperspectral imaging system," in *International Conference on Image Analysis and Processing*, 2011, pp. 303–312.

- [42]G. Maino and G. L. Foresti, Eds., *Image Analysis and Processing ICIAP 2011*, vol. 6979. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [43]R. Gosselin, D. Rodrigue, R. González-Núñez, and C. Duchesne, "Potential of hyperspectral imaging for quality control of polymer blend films," *Ind. Eng. Chem. Res.*, vol. 48, no. 6, pp. 3033–3042, 2009.
- [44]H.-T. Lim and V. M. Murukeshan, "Spatial-scanning hyperspectral imaging probe for bioimaging applications," *Rev. Sci. Instrum.*, vol. 87, no. 3, p. 33707, Mar. 2016.
- [45]M. A. Calin, S. V. Parasca, R. Savastru, and D. Manea, "Characterization of burns using hyperspectral imaging technique – A preliminary study," *Burns*, vol. 41, no. 1, pp. 118–124, Feb. 2015.
- [46] H. Chihara *et al.*, "In vivo detection of atherosclerotic plaque using non-contact and label-free near-infrared hyperspectral imaging," *Atherosclerosis*, vol. 250, pp. 106–113, Jul. 2016.
- [47] M. H. E. Hermans, "Results of an Internet Survey on the Treatment of Partial Thickness Burns, Full Thickness Burns, and Donor Sites:," J. Burn Care Res., vol. 28, no. 6, pp. 835–847, Nov. 2007.
- [48]M. Eskandarlou, M. Azimi, S. Rabiee, and M. A. S. Rabiee, "The Healing Effect of Amniotic Membrane in Burn Patients."
- [49]M. Friebel, A. Roggan, G. Müller, and M. Meinke, "Determination of optical properties of human blood in the spectral range 250 to 1100 nm using Monte Carlo simulations with hematocrit-dependent effective scattering phase functions," *J. Biomed. Opt.*, vol. 11, no. 3, p. 34021, 2006.
- [50] M. Friebel, K. Do, A. Hahn, G. Mu, and others, "Optical properties of circulating human blood in the wavelength range 400–2500 nm," J. Biomed. Opt., vol. 4, no. 1, pp. 36–46, 1999.
- [51]C. J. Robison, C. Kolanko, T. Bourlai, and J. M. Dawson, "Imaging of blood cells based on snapshot Hyper-Spectral Imaging systems," 2015, p. 94721L.
- [52]C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, 1998.
- [53]M. Kass, A. Wikin, and D. Terzopoulos, "Snakes: Active Contour Models," Int. J. Comput. Vis., pp. 321–331, 1988.
- [54] P. Version, ICC.1:2004-10 International Color Consortium ® Specification. .
- [55] J. Cheewatanon, T. Leauhatong, S. Airpaiboon, and M. Sangwarasilp, "A new white blood cell segmentation using mean shift filter and region growing algorithm," *Int. J. Appl.*, vol. 4, no. 1, p. 31, 2011.
- [56]C. Zhang et al., "White Blood Cell Segmentation by Color-Space-Based K-Means Clustering," Sensors, vol. 14, no. 9, pp. 16128–16147, Sep. 2014.
- [57]J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundam. Informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.
- [58]X. Zheng, Y. Wang, G. Wang, and Z. Chen, "A novel algorithm based on visual saliency attention for localization and segmentation in rapidly-stained leukocyte images," *Micron*, vol. 56, pp. 17–28, Jan. 2014.
- [59]S. Arslan, E. Ozyurek, and C. Gunduz-Demir, "A color and shape based algorithm for segmentation of white blood cells in peripheral blood and bone marrow images: Color and Shape Based Cell Segmentation," *Cytometry A*, vol. 85, no. 6, pp. 480–490, Jun. 2014.
- [60]E. Górska *et al.*, "Determination of monocyte count by hematological analyzers, manual method and flow cytometry in polish population," *Cent. Eur. J. Immunol.*, vol. 31, no. 1/2, p. 1, 2006.
- [61]Y. Li, R. Zhu, L. Mi, Y. Cao, and D. Yao, "Segmentation of White Blood Cell from Acute Lymphoblastic Leukemia Images Using Dual-Threshold Method," *Comput. Math. Methods Med.*, vol. 2016, pp. 1–12, 2016.
- [62]Z. Liu *et al.*, "Segmentation of White Blood Cells through Nucleus Mark Watershed Operations and Mean Shift Clustering," *Sensors*, vol. 15, no. 9, pp. 22561–22586, Sep. 2015.

- [63] J. Pillay *et al.*, "In vivo labeling with 2H2O reveals a human neutrophil lifespan of 5.4 days," *Blood*, vol. 116, no. 4, pp. 625–627, Apr. 2010.
- [64] M. H. Kim, "The Three-Dimensional Evolution of Hyperspectral Imaging," in Smart Sensors and Systems, Y.-L. Lin, C.-M. Kyung, H. Yasuura, and Y. Liu, Eds. Cham: Springer International Publishing, 2015, pp. 63–84.
- [65]B. Pan, Z. Jiang, H. Zhang, X. Luo, and J. Wu, "Improved grey world color correction method based on weighted gain coefficients," 2014, p. 927331.
- [66]S. M. Yamany, A. A. Farag, and S.-Y. Hsu, "A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems," *Pattern Recognit. Lett.*, vol. 20, no. 11, pp. 1431–1438, 1999.
- [67] W. Ma, C. Gong, Y. Hu, P. Meng, and F. Xu, "The Hughes phenomenon in hyperspectral classification based on the ground spectrum of grasslands in the region around Qinghai Lake," 2013, p. 89101G.
- [68] T. M. Lillesand, R. W. Kiefer, and J. W. Chipman, *Remote sensing and image interpretation*, Seventh edition. Hoboken, N.J: John Wiley & Sons, Inc, 2015.
- [69]D. A. Landgrebe, Signal Theory Methods in Multispectral Remote Sensing. John Wiley & Sons, Inc, 2005.
- [70] J. Prinyakupt and C. Pluempitiwiriyawej, "Segmentation of white blood cells and comparison of cell morphology by linear and naïve Bayes classifiers," *Biomed. Eng. OnLine*, vol. 14, no. 1, p. 63, Jun. 2015.
- [71]N.-Y. Lee, "Block-iterative Richardson-Lucy methods for image deblurring," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, Dec. 2015.
- [72] Sysmex, "DI-60 Integrated Slide Processing System." Sysmex.
- [73] Y. Guan, Q. Li, Y. Wang, H. Liu, and Z. Zhu, "Pathological leucocyte segmentation algorithm based on hyperspectral imaging technique," *Opt. Eng.*, vol. 51, no. 5, pp. 53202–1, 2012.
- [74] "Blood cells." [Online]. Available: http://www.biosbcc.net/doohan/sample/htm/Blood%20cells.htm. [Accessed: 09-Oct-2015].
- [75] "Hemostasis." [Online]. Available: http://www.biosbcc.net/doohan/sample/htm/Hemostasis.htm. [Accessed: 09-Oct-2015].
- [76]G. Mayer, Immunology Chapter One: Innate (non-specific) Immunity. USC School of Medicine, 2006.
- [77] A. Rad, Hematopoiesis\_simple.svg. 2009.
- [78]B. E. A. Saleh and M. C. Teich, *Fundamentals of photonics*. Hoboken, N.J.: Wiley-Interscience, 2007.
- [79]H. Kuzu-Karsilayan, E. Eryilmaz, G. Yillar, G. Deniz, and G. Yanikkaya-Demirel, "Spectrophotometric determination of leukocytes in blood," *J. Clin. Lab. Anal.*, vol. 16, no. 5, pp. 233–236, 2002.
- [80] M. Monici *et al.*, "Natural fluorescence of white blood cells: spectroscopic and imaging study," *J. Photochem. Photobiol. B*, vol. 30, no. 1, pp. 29–37, Sep. 1995.
- [81] "UV-visible absorption spectra." [Online]. Available: http://www.chemguide.co.uk/analysis/uvvisible/theory.html#top. [Accessed: 16-Oct-2015].
- [82]X. Zou and J. Zhao, *Nondestructive Measurement in Food and Agro-products*. Dordrecht: Springer Netherlands, 2015.
- [83]R. Ananthakrishnan and A. Ehrlicher, "The Forces Behind Cell Movement," *Int. J. Biol. Sci.*, vol. 3, no. 5, pp. 303–317, Jun. 2007.
- [84]M. Attas *et al.*, "Near-infrared spectroscopic imaging in art conservation: investigation of drawing constituents," *J. Cult. Herit.*, vol. 4, no. 2, pp. 127–136, Apr. 2003.
- [85] Brimrose Corporation, aotf-diagram.jpg. .
- [86] J. Vila-Francés, J. Calpe-Maravilla, L. Gómez-Chova, and J. Amorós-López, "Analysis of acousto-optic tunable filter performance for imaging applications," *Opt. Eng.*, vol. 49, no. 11, pp. 113203–113203–9, 2010.

- [87]L. Gao, R. T. Kester, N. Hagen, and T. S. Tkaczyk, "Snapshot Image Mapping Spectrometer (IMS) with high sampling density for hyperspectral microscopy," *Opt. Express*, vol. 18, no. 14, pp. 14330–14344, Jun. 2010.
- [88] Y.-L. Lin, C.-M. Kyung, H. Yasuura, and Y. Liu, Eds., *Smart Sensors and Systems*. Cham: Springer International Publishing, 2015.
- [89] Ximea Inc., "XIMEA\_imec\_HSI\_technology-Part-V1.1.pdf.".
- [90] A. Lambrechts, P. Gonzalez, B. Geelen, P. Soussan, K. Tack, and M. Jayapala, "A CMOScompatible, integrated approach to hyper- and multispectral imaging," 2014, p. 10.5.1-10.5.4.
- [91]K. L. Date *et al.*, "Level 1 Product Output Files Data Format Control Book Volume 5, Book 2," 2000.
- [92] A. Khashman, "Blood Cell Identification using Emotional Neural Networks.," J Inf Sci Eng, vol. 25, no. 6, pp. 1737–1751, 2009.
- [93]J. Adams, M. Smith, and P. Johnson, "Spectral Mixture Modeling: a New Analysis of Rock and Soil Types at the Viking Lander 1 Site," J. Geophys. Res., vol. 91, pp. 8098–8112, 1986.
- [94]N. Keshava and J. F. Mustard, "Spectral unmixing," Signal Process. Mag. IEEE, vol. 19, no. 1, pp. 44–57, 2002.
- [95]A. Plaza, Q. Du, J. Bioucas-Dias, X. Jia, and F. Kruse, "Foreword to the Special Issue on Spectral Unmixing of Remotely Sensed Data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4103–4110, 2011.
- [96] T. Peynot, "Chapter 4: FIltering in the Frequency Domain."
- [97] A. Noda, "A modified FFT-based filter for noise reduction of EEG measurements in identifying task learning processes," 1989, p. 1007.
- [98]M. K. Azadboni and A. Behrad, "Text detection and character extraction in color images using FFT domain filtering and SVM classification," 2012, pp. 794–799.
- [99]NIH.gov, FFT\_Lena.jpg. .
- [100] "Johns Hopkins University Spectral Library Spectral Library.html.".
- [101] C. Bateson, G. P. Asner, C. Wessman, and others, "Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis," *Geosci. Remote Sens. IEEE Trans. On*, vol. 38, no. 2, pp. 1083–1094, 2000.
- [102] P. J. Martínez, R. M. Pérez, A. Plaza, P. L. Aguilar, M. C. Cantero, and J. Plaza, "Endmember extraction algorithms from hyperspectral images," 2006.
- [103] J. M. Bioucas-Dias *et al.*, "Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [104] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *Geosci. Remote Sens. IEEE Trans. On*, vol. 26, no. 1, pp. 65–74, 1988.
- [105] "NASA Technical Reports Server (NTRS) Discrimination among semi-arid landscape endmembers using the Spectral Angle Mapper (SAM) algorithm." [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:C5snPHwreRUJ:ntrs.nasa.gov/searc h.jsp% 3FR% 3D19940012238+&cd=2&hl=en&ct=clnk&gl=us. [Accessed: 08-Oct-2015].
- [106] "MATLAB Central MATLAB Hyperspectral Toolbox -." [Online]. Available: https://www.mathworks.com/matlabcentral/linkexchange/links/1845-matlab-hyperspectraltoolbox. [Accessed: 08-Oct-2015].
- [107] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: Wiley, 2001.
- [108] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philos. Mag.*, vol. 2, 1901.
- [109] N. Cristianini, "Support vector and kernel machines," *Tutor. ICML*, 2001.
- [110] L. Wang, *Support vector machines: theory and applications*, vol. 177. Springer Science & Business Media, 2005.

- [111] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.
- [112] NumPy Developers, NumPy. .
- [113] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, Mar. 2011.
- [114] C. Gohlke and Laboratory for Flourescence Dynamics, University of California, Irvine, "Unofficial Windows Binaries for Python Extension Packages."
- [115] SciPy developers, *SciPy: Scientific Python*. 2016.
- [116] Thomas Boggs, *Spectral Python*. 2016.
- [117] JetBrains, PyCharm. .
- [118] Rebellion Photonics, "ARROW Product Specs.".
- [119] Brimrose Corporation of America, "Acousto-Optic Tunable Filter VA210 Video Adapter." Brimrose Corporation of America.
- [120] "Preparing a Buffy Coat from Whole Blood | Human Immunology Portal.".
- [121] Blood-plasma-fractionation.
- [122] "ISO 8601:2004.".
- [123] SCHOTT Glass Manufacturers, "ACE® Light Source." 15-Apr-2009.
- [124] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," J. Mach. Learn. Res., vol. 11, no. Jul, pp. 2079– 2107, 2010.
- [125] R. G. Resmini, C. J. Deloye, and D. W. Allen, "An analysis of the probability distribution of spectral angle and Euclidean distance in hyperspectral remote sensing using microspectroscopy," 2013, p. 87431I.
- [126] "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation - TRRA\_2007.pdf." [Online]. Available: http://www.flinders.edu.au/science\_engineering/fms/School-CSEM/publications/tech\_repsresearch\_artfcts/TRRA\_2007.pdf. [Accessed: 08-Oct-2015].
- [127] "The n-dimensional Fourier Transform.".
- [128] E. R. Dougherty and R. A. Lotufo, *Hands-on morphological image processing*. Bellingham, Wash: SPIE Optical Engineering Press, 2003.
- [129] Y. Chen and I. G. Councill, "An introduction to support vector machines: A review," *AI Mag.*, vol. 24, no. 2, p. 105, 2003.

# Appendix A: Script Provided by Rebellion Photonics to Decode BSQ Files Captured by Arrow System.

```
#main section of convertBsaMulti.pv
if name == " main ":
   pathName = r'/media/crob/HyperSpec/-_Last_Minute_Data_-' #F:\-_Research Data_-
\BLood 9 3 2015\SLide 1 /media/crob/USB30FD/HyperSpec Data/All bsg static
   os.chdir(pathName)
   #objNames = glob.glob1(pathName, "*.bsq")
   #bsqCount = Len(objNames)
   ## bsqCount = len(glob.glob1(pathName,"*.bsq")) Need both number of files as
well as filenames
   ## This loop repeats for all subfolders
   all subDirs = [d for d in os.listdir('.') if os.path.isdir(d)]
   for dirs in all subDirs:
       sDir = os.path.join(pathName, dirs)
       objNames = glob.glob1(sDir,"*.bsq")
       bsqCount = len(objNames)
       ## insert for loop here
       for i in range(0, bsqCount):
           dir = os.path.dirname(os.path.join(sDir,objNames[i]))
           if not os.path.exists(dir+'/TIF/'):
               os.mkdir(dir+'/TIF/')
           outfile = dir + '/TIF/' +
os.path.basename(os.path.join(sDir,objNames[i]))[:-4]
           (dcb, lambdas) = readbsq(os.path.join(sDir,objNames[i]))
           print('dcb.shape=', dcb.shape)
           print('dcb.dtype=', dcb.dtype)
           print('lambdas=', lambdas)
       ## Convert type double pixel data to type uint16 for exporting to TIFF.
           minval = amin(dcb)
           maxval = amax(dcb)
           print('original dcb: minval=%f, maxval=%f' % (minval, maxval))
           new_dcb = int16((dcb - minval) * (2**15 - 1) / float64(maxval - minval))
           print('new dcb.dtype=', new dcb.dtype)
           print('rescaled dcb: minval=%i, maxval=%i' % (amin(new dcb),
amax(new dcb)))
           img = dcb[:, :, 0]
           im = Image.fromarray(img)
           #im.show()
           #imshow(img)
       ## Write the result as a 16-bit TIFF file. Note that very few TIFF viewers
support 16-bit pixels! ImageJ is a free and widely available one, though.
         write_dcb_tiffs(outfile, dcb, lambdas)
```

```
from numpy import *
import matplotlib.pyplot as plt
import struct, os
from PIL import Image
#import imaging
import sys
sys.path
#from libtiff import TIFFfile
#import tifffile.c as tiff
import tifffile as TIFFfile
from scipy import io
import glob ## used to count files in a directory
sys.path
def imshow(image, ax=None, **kwargs):
   image = array(image)
   iscolor = (image.ndim == 3)
   if (ax == None):
       ax = plt.gca()
   def myformat_coord(x, y):
       col = int(x+0.5)
       row = int(y+0.5)
       if iscolor:
           (Nx,Ny,_) = image.shape
       else:
           (Nx,Ny) = image.shape
       if (col \ge 0) and (col < Ny) and (row \ge 0) and (row < Nx):
           if iscolor:
               z = image[row,col,:]
           else:
               z = image[row,col]
           if iscolor or (int(z) == z):
               if iscolor:
                  return('x=%i, y=%i, z=(%i,%i,%i)' % (row, col, z[0], z[1], z[2]))
               else:
                  return('x=%i, y=%i, z=%i' % (row, col, z))
           else:
               return('x=%i, y=%i, z=%1.4f' % (row, col, z))
       else:
          return('x=%i, y=%i'%(y, x))
   plt.imshow(image, **kwargs)
   ax.format coord = myformat coord
   return
```

```
def readbsq(filename, debug=False):
   filebytes = open(filename, "rb").read()
   output = struct.unpack('<25c', filebytes[0:25])</pre>
   \#vv = output.join()
   vv = list(output)
   #in python 3.x you need to put a b'' instead of '' to signify you're reading
bytes into a string/bytearray
   #putting a b in front of a string converts it into a byte array. Why did the
docs never mention this? beats me.
   version string = b''.join(vv)
   #outputL = output.decode()
   #version string = ''.join(outputL)
   if debug: print(version string + '\n')
   if (version string == b'RPSpectralCube v 0.1\x00\x00\x00\x00\x00\x00'):
       version number = 0
       firstbyte = 25
   elif (version string ==
version number = int32(struct.unpack('>I', filebytes[25:29])) #[0]
       firstbyte = 29
   else:
       raise ImportError('readbsg() cannot recognize that BSO file version.')
   print('BSQ file version = ' + str(version number))
   (Nx,Ny,Nw) = struct.unpack('>III', filebytes[firstbyte:firstbyte+12])
   firstbyte += 12
   if debug: print(Nx,Ny,Nw,firstbyte)
   ## From the Nw value, read in the set of wavelengths, each of type float64.
   lambdas = zeros(Nw)
   lambdas = struct.unpack('>' + str(Nw) + 'd',
filebytes[firstbyte:firstbyte+(Nw*8)])
   firstbyte += Nw*8
   if debug: print('lambdas=', lambdas)
   if (version number == 0):
       pass
   elif (version number == 101):
       pixmin = struct.unpack('>d', filebytes[firstbyte:firstbyte+8])
       pixmax = struct.unpack('>d', filebytes[firstbyte+8:firstbyte+16])
       firstbyte = firstbyte + 16
       if debug: print('pixmin,pixmax = ', pixmin, pixmax)
   else:
       raise NotImplementedError('That version number (' + str(version number) + ')
is not implemented yet!')
   dcb = zeros((Nx,Ny,Nw))
   for w in arange(Nw):
       for y in arange(Ny):
           dcb[:,y,w] = struct.unpack('<'+str(Nx)+'f',</pre>
filebytes[firstbyte:firstbyte+(Nx*4)])
           firstbyte += Nx*4
return([dcb, lambdas])
```

```
def write dcb tiffs(fileroot, dcb, lambdas = None):
   (Nx,Ny,Nw) = dcb.shape
   if (lambdas == None):
     lambdas = []
     for w in arange(Nw):
        lambdas.append('%02i' % w)
   for w in arange(Nw):
     wavestr = ('%03i' % int(lambdas[w]))
     filename = fileroot + "-" + wavestr + '.tif'
     print('Writing filename "' + filename + '"')
     img = dcb[:,:,w]
     write_16bitTiff(filename, img)
   return
def readtiff(filename):
   assert os.path.exists(filename), 'File "' + filename + '" does not exist.'
  tif = TIFFfile(filename)
  img = tif.asarray()
  tif.close()
  img = flipud(img)
return(img)
```

This script extracts each band in the ENVI binary sequential formatted datacubes into a TIF image file. This conversion was only used to help visualize the data. Other than Rebellion Photonics' Crystal software, there were no other software tools available to decode and view the datacubes in a relatively simple manner.

# **Appendix B: BSQ to HDF5 Conversion Script**

```
__author__ = "Christo Robison"
import numpy as np
from scipy import signal
from scipy import misc
import h5py
from PIL import Image
import os
import collections
import matplotlib.pyplot as plt
import convertBsqMulti as bsq
import png
'''This program reads in BSQ datacubes into an HDF file'''
def convLabels(labelImg, numBands):
   takes a MxNx3 numpy array and creates binary labels based on predefined classes
   background = 0
   red = 1 WBC
   green = 2 \text{ RBC}
   pink = 3 nuclear material
   yellow = 4 ignore
    ....
   #b = np.uint8(numBands / 31)
   # print(b / 31)
   tempRed = labelImg[:,:,0] == 255
   tempGreen = labelImg[:,:,1] == 255
   tempBlue = labelImg[:,:,2] == 255
   tempYellow = np.logical and(tempRed, tempGreen)
   tempPink = np.logical_and(tempRed, tempBlue)
   temp = np.zeros(np.shape(tempRed))
   temp[tempRed] = 1
   temp[tempGreen] = 2
   temp[tempPink] = 3
   temp[tempYellow] = 4
   print(temp)
   print(tempRed, tempGreen, tempBlue, tempYellow, tempPink)
```

```
return temp
```

```
def loadBSQ(path = '/home/crob/HyperSpec_Data/WBC v ALL/WBC25', debug=False):
    d31 = []
    d31_norm = []
   d25 = []
    d25_norm = []
    125 = []
    1 = []
    13 = []
    lam = []
    for root, dirs, files in os.walk(path):
        print(dirs)
        for name in sorted(files): #os walk iterates arbitrarily, sort fixes it
            print(name)
            if name.endswith(".png"):
                # Import label image
                im = np.array(Image.open(os.path.join(root,name)),'f')
                print np.shape(im)
                im = im[:,:,0:3] \# > 250
                print(np.shape(im))
                im = np.flipud(im)
                l.append(im)
                print(np.shape(im))
                print("Name = " + name)
            if name.endswith(".bsq"):
                bs = bsq.readbsq(os.path.join(root,name))
                print(np.shape(bs[0]))
                print(len(bs[1]))
                #separate bsq files by prism
                if len(bs[1]) == 31:
                    print('BSQ is size 31')
                    print(len(bs[1]))
                    lam = bs[1]
                    #d31.append(np.reshape(np.transpose(bs[0], (1, 2, 0)), 4298429))
                    d31.append(bs[0].astype(np.float32))
                    d31 norm.append(bs[0].astype(np.float32)/np.amax(bs[0]))
                if len(bs[1]) == 25:
                    print('BSQ is size 25')
                    print(len(bs[1]))
                    lam = bs[1]
                    d25.append(bs[0].astype(np.float32))
                    d25_norm.append(bs[0].astype(np.float32)/np.amax(bs[0]))
                    #d25.append(np.reshape(bs[0],[138659,25]).astype(np.float32))
                    # old don't use #d25.append(np.reshape(np.transpose(bs[0], (1, 2,
0)), 3466475))
   out = collections.namedtuple('examples', ['data31', 'data31_norm', 'data25',
'data25_norm', 'labels', 'lambdas'])
    o = out(data31=np.dstack(d31),data31_norm=np.dstack(d31_norm), data25=d25,
data25 norm=d25 norm, labels=np.dstack(1), lambdas=lam) #np.vstack(d25),
Labels=np.hstack(L)
  return o
```

```
def convLabels(labelImg, numBands):
    . . .
   takes a MxNx3 numpy array and creates binary labels based on predefined classes
   background = 0
   red = 1 WBC
    green = 2 \text{ RBC}
    pink = 3 nuclear material
   yellow = 4 ignore
    tempRed = labelImg[:,:,0] == 255
    tempGreen = labelImg[:,:,1] == 255
    tempBlue = labelImg[:,:,2] == 255
    tempYellow = np.logical and(tempRed, tempGreen)
    tempPink = np.logical_and(tempRed, tempBlue)
    temp = np.zeros(np.shape(tempRed))
    temp[tempRed] = 1
    temp[tempGreen] = 2
    temp[tempPink] = 3
    temp[tempYellow] = 4
    print(temp)
    print(tempRed, tempGreen, tempBlue, tempYellow, tempPink)
    return temp
def convert_labels(labels,n_classes, debug = False):
    for j in range(n classes):
       temp = labels == j
       temp = temp.astype(int)
       if j > 0:
            conv labels = np.append(conv labels, temp)
            print(temp[:])
       else:
            conv_labels = temp
    print(np.shape(conv labels))
    conv_labels = np.reshape(conv_labels, [len(labels), n_classes], order='F')
    if debug: print(np.shape(conv labels))
    if debug:
       f = h5py.File("/home/crob/HyperSpec/Python/BSQ whole.h5", "w")
       f.create_dataset('bin_labels', data=conv_labels)
        f.close()
    return conv labels
def getClassMean(data, classNum):
    kee = np.equal(data['label'],classNum)
    out = np.mean(data['data']*kee,axis=0)
    return out
def getAverages(data, numClasses):
   out = []
    for i in range(numClasses):
        a = getClassMean(data, i)
       out.append(a)
  return out
```

```
if __name__ == '__main__':
   path = '/home/crob/-_PreSortedData_Train_-' s = loadBSQ(path)
   print(np.shape(s.data25))
   f = h5py.File("HYPER_SPEC_TRAIN_RED.h5", "w")
   f.create_dataset('data', data=s.data31, chunks=(443, 313, 1))
   f.create_dataset('norm_data', data=s.data31_norm, chunks=(443,313,1))
   f.create_dataset('labels', data=s.labels)
   f.create_dataset('bands', data=s.lambdas)
   #Move data into datasets
   g = np.shape(s.data31)
   b = np.uint16(g[2] / 31) #more than 256 samples
   lab = np.reshape(s.labels, [443, 313, 3, b], 'f')
   numExamples = np.shape(lab)
   a = []
   for j in range(np.uint16(numExamples[3])):
       a.append(convLabels(lab[:, :, :, j], None))
   f.create_dataset('classLabels', data=np.dstack(a))
   f.close()
```

This is the script used to convert data that was separated, masked, and stored as training and testing sets into a HDF database. These database files allow for easy access to the data with little programming experience. To add data to the data set or create a new one, data and labels must be stored into a shared directory. Also, training and testing data (and labels) should not be stored in the same directory or they will be added to the same HDF file.

# **Appendix C: Script Used to Generate HDF Datasets:**

# \_\_author\_\_ = 'Christo Robison'

```
#This program preprocesses the data by removing artifacts caused by the snapshot HSI system
```

```
from spectral import *
import numpy as np
from PyQt4 import QtGui
import pyqtgraph as pg
from pyqtgraph.widgets.MatplotlibWidget import MatplotlibWidget
from scipy import fftpack as ft
import h5py
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import time
from skimage import io, exposure, img_as_uint, img_as_float
import png
import pyfftw
import HyperSpecGui
show full out = False
if show_full_out: np.set_printoptions(threshold=np.nan)
******
def getData(filename=None, dat idx=None, lab idx=None):
   if filename is None: filename = 'D:\- Hyper Spec -\HYPER SPEC TEST.h5'
   f = h5py.File(filename, 'r')
   if dat idx is None:
       dcb = f['data'][:] #Extract normalized data for svm b/c intensity sensitive
   else:
       dcb = f['data'][:, :, dat idx:dat idx+len(f['bands'])]
   if lab idx is None:
       labels = f['labels'][:]
       classLabels = f['classLabels'][:]
   else:
       labels = f['labels'][:, :, lab_idx]
       classLabels = f['classLabels'][:, :, lab_idx]
   bands = f['bands'][:]
   #classLabels = f['classLabels'][:]
   out = {'dcb': dcb, 'labels': labels, 'lambdas': bands, 'classLabels':
classLabels}
   f.close()
return out
```

```
def im2png(img, filename):
   f = open(filename, 'wb')
   w = png.Writer(313, 443, greyscale=True, bitdepth=8)
   if img.dtype is not np.int8:
      if (np.amin(img) < 0): img = img - np.amin(img)</pre>
      img = np.float32(img) * (2.0**7 - 1.0)/ np.amax(img)
      img = np.int8(img)
   w.write(f, img)
   f.close()
def imSave(img, filename, range='float'):
   f = open(filename, 'wb')
   w = png.Writer(313, 443, greyscale=True, bitdepth=8)
   img w = exposure.rescale intensity(img,out range=range)
   img w = img as uint(img w)
   w.write(f, img w)
   f.close()
def shapeData(data, labels, numExamples, numBands, altDims = None):
   '''Takes input data matrix and reshapes it into HW,D format
   i.e. endmembers and their appropriate class labels'''
   if altDims is None: altDims = [443, 313, numBands, numExamples]
   temp = np.reshape(data, altDims, 'f')
   dataR = np.reshape(temp,[-1, numBands])
   labelL = np.reshape(labels, [-1,1])
   out = {'data': dataR, 'label': labelL}
   return out
def hsfft(dcb):
   '''Takes in a datacube (dcb) and returns n-dim FFT of datacube'''
   temp dcb = pyfftw.n byte align(dcb, 16, dtype='complex128')
   dcb fft = pyfftw.interfaces.numpy fft.fftn(temp dcb)
   return dcb fft
*******
def genMask(h=443, w=313, offset=41, debug=False):
   '''Generates a fft mask to eliminate HS system artifacts from image'''
   mask = np.ones((h, w), dtype='float32')
   mask[:, (w/2):((w+2/2)/2)] = 0
   mask[((h/2)-offset):((h/2)+offset), :] = 1
   if debug: print(mask)
   return mask
def applyMask(dcb fft, mask=genMask, debug=False):
   '''Applies mask to spatial information of each band in datacube.'''
   masked fft = np.multiply(np.rollaxis(dcb_fft, 2, 0,), ft.ifftshift(mask))
   if debug: print("Shape of FFT:" + str(np.shape(masked_fft)))
 return masked fft
```

```
def hsifft(dcb_fft, debug=False):
   '''Performs inverse fourier transform on datacube'''
   dcb_out = pyfftw.interfaces.numpy_fft.ifftn(dcb_fft)
   if debug: print("Shape of iFFT:" + str(np.shape(dcb out)))
   return dcb out
******
def dcbFilter(dcb, h=443, w=313, offset=41, debug=False):
   '''A helper function to simplify masking process and reduce confusion'''
   masked fft = applyMask(hsfft(dcb), genMask(h, w, offset, debug))
   dcb out = hsifft(masked fft, debug)
   return dcb out
def dispDCB(dcb):
   dcb = np.swapaxes(dcb, 2, 0)
******
if name == ' main ':
   # initialize Qt, needed once per application
   app = QtGui.QApplication([])
   # Define widget to hold outputs
   MainWindow = QtGui.QMainWindow()
   MainWindow.setWindowTitle('HyperSpec: Test')
   ui = HyperSpecGui.Ui MainWindow()
   ui.setupUi(MainWindow)
   MainWindow.show()
   trainData = getData(filename='D:\-_Hyper_Spec_-\HYPER_SPEC_TEST_RED.h5')
   testData = getData(filename='D:\-_Hyper_Spec_-\HYPER_SPEC_TEST_RED.h5')
   print(np.shape(trainData['dcb']))
   debug = False
   if debug is True:
       for i in range(np.shape(trainData['dcb'])[2]):
          im = exposure.rescale_intensity(trainData['dcb'][:, :, i],
out range='float')
          im = img_as_uint(im)
          io.imsave((r'HYPER SPEC TEST\band image ' + str(i) + '.png'), im)
   ##### Pull out a data cube and display it in a pyqtgraph object#####
   img = trainData['dcb'][:, :, 343:370]
   img1 = np.swapaxes(img, 2, 0)
   ui.ImageView1.setImage(img1)
   img_file = open('fftMask_new.png', 'wb')
   img_w = png.Writer(313, 443, greyscale=True, bitdepth=16)
   img_to_write = exposure.rescale_intensity(img[:, :, 3], out_range='float')
   img to write = img as uint(img to write)
   img_w.write(img_file, img_to_write)
   img file.close()
   ui.ImageView2.setImage(img[:, :, 3])
   mask = np.ones((443, 313), dtype='float32')
   mask[:, 155:157] = 0 #10:430
   mask[180:262, :] = 1
   print(mask)
```

```
ui.ImageView4.setImage(mask)
temp img = pyfftw.n byte align(img, 16, dtype='complex128')
img_fft = pyfftw.interfaces.numpy_fft.fftn(temp_img)
print(img fft.dtype)
print(img_fft.shape)
#if projecting mask along data, need to make sure last two dimensions match
Ex: 4Lx3Wx10H * 3Wx10H = 4Lx3Wx10H
mask_ifft = np.multiply(np.rollaxis(img_fft,2,0), ft.ifftshift(mask))
print("Shape of FFT: " + str(np.shape(mask ifft)))
print("Shape of FFT: " + str(np.shape(mask_ifft)))
img_FFTout = np.log2(pyfftw.interfaces.numpy_fft.ifftn(mask_ifft))
print("Shape of FFT Out: " + str(np.shape(img FFTout)))
img_p = pyfftw.interfaces.numpy_fft.ifftn(mask_ifft)
ui.ImageView3.setImage(img_p)
#Start implementing parts from final_tests.py
(m, c) = kmeans(img p.real, 6, 300)
mw = pg.widgets.MatplotlibWidget.MatplotlibWidget()
subplot = mw.getFigure().add_subplot(111)
for i in range(c.shape[0]):
    subplot.plot(c[i])
mw.draw()
sys.exit(app.exec_())
```

This script consists of simple functions to generate HDF5 datasets.
# **Appendix D: HyperSpec Main Application :**

```
__author__ = 'Christo Robison'
from PyQt4 import QtGui, QtCore
import sys
import HyperSpecGui
import numpy as np
from scipy import ndimage
from scipy import fftpack as ft
import matplotlib
matplotlib.use('QT4Agg')
import matplotlib.pylab as pylab
import matplotlib.pyplot as plt
import time
import pyqtgraph
import hs_imFFTW as hs
from spectral import *
spectral.settings.WX_GL_DEPTH_SIZE = 16
. . .
background = 0
red = 1 WBC
green = 2 RBC
pink = 3 nuclear material
yellow = 4 ignore'''
class HyperSpecApp(QtGui.QMainWindow, HyperSpecGui.Ui MainWindow):
    def init (self, parent=None):
        pyqtgraph.setConfigOption('background', 'k') #before Loading widget
        super(HyperSpecApp, self).__init__(parent)
        self.setupUi(self)
        self.updateBtn.clicked.connect(self.update)
        STEPS = np.array([0.0, 0.2, 0.6, 1.0])
        CLRS = ['k', 'r', 'y', 'w']
        clrmp = pyqtgraph.ColorMap(STEPS,
np.array([pyqtgraph.colorTuple(pyqtgraph.Color(c)) for c in CLRS]))
        data = np.random.normal(size=(100, 200, 200))
        self.ImageView2.setImage(data)
        self.pltView1.plotItem.showGrid(True, True, 0.7)
        self.pltView2.plotItem.showGrid(True, True, 0.7)
    def update(self):
        t1 = time.clock()
        points = 100
        print("update took %.02f ms" % ((time.clock() - t1) * 1000))
```

```
def runKmeans(view, inputImg, clusters=6, iters=300):
   (m, c) = kmeans(inputImg, clusters, iters)
   for i in range(c.shape[0]):
      view.plot(c[i])
   return (m, c)
def adjustLabels(dcb, bkgnd=0.0, unknown=4.0):
   dcb[dcb==0.0] = 5.0
   dcb[dcb==4.0] = 0.0
   dcb[dcb=5.0] = 4.0
   return dcb
def runSpectral(dcb, gt, title = 'dcb'):
   (classes, gmlc, clmap) = runGauss(dcb, gt)
   (gtresults, gtErrors) = genResults(clmap, gt)
   displayPlots(clmap, gt, gtresults, gtErrors, (title+" Gaussian Classifer"))
   return (gtresults, gtErrors)
def runPCA(dcb, gt, title = 'dcb'):
   pc = principal components(dcb)
   pc 0999 = pc.reduce(fraction=0.999)
   img_pc = pc_0999.transform(dcb)
   (classes, gmlc, clmap) = runGauss(img pc, gt)
   (gtresults, gtErrors) = genResults(clmap, gt)
   displayPlots(clmap, gt, gtresults, gtErrors, (title+" PCA Gaussian Classifer"))
   return (gtresults, gtErrors, pc)
******
def genResults(clmap, gt):
   gtresults = clmap * (gt!=0)
   gtErrors = gtresults * (gtresults !=gt)
   return (gtresults, gtErrors)
******
def runGauss(dcb, gt):
   classes = create training classes(dcb, gt)
   gmlc = GaussianClassifier(classes, min samples=200)
   clmap = gmlc.classify image(dcb)
   return (classes, gmlc, clmap)
******
def displayPlots(clmap, gt, gtresults = None, gtErrors = None, title = 'classifier'):
   if (gtresults is None and gtErrors is None):
      (gtresults, gtErrors) = genResults(clmap, gt)
   v0 = imshow(classes=clmap, title=(title+" results"))
   pylab.savefig((title+" results.png"), bbox_inches='tight')
   v1 = imshow(classes = gtresults, title=(title+" gt Results"))
   pylab.savefig((title + " gt Results.png"), bbox inches='tight')
   v2 = imshow(classes = gtErrors, title=(title+" Error"))
  pylab.savefig((title + " Error.png"), bbox inches='tight')
```

```
def cleanResults(inputImg, cls_iter=1, open_iter=1):
   open = ndimage.binary opening(inputImg, iterations=open iter)
   close = ndimage.binary_opening(inputImg, iterations=cls_iter)
   return (open, close)
******
def combineLabels(rbc, wbc, nuc, bkgd):
   out = np.zeros(np.shape(rbc), dtype=np.float64)
   out[bkgd == 1] = 4.0
   out[rbc == 1] = 2.0
   out[wbc == 1] = 1.0
   out[nuc == 1] = 3.0
   out[out == 0] = 0.0
   print(out)
   return out
******
def create rgb(classing, colormap=None):
   if colormap is None:
      colormap = np.array([[0, 0, 0], [255, 0, 0], [0, 255, 0], [255, 0, 255],
[255, 255, 0]], dtype=np.ubyte)
   h,w = np.shape(classing)
   out = np.zeros([h, w, 3], dtype=np.uint8)
   out[classing == 0.0] = colormap[0]
   out[classing == 1.0] = colormap[1]
   out[classing == 2.0] = colormap[2]
   out[classimg == 3.0] = colormap[3]
   out[classing == 4.0] = colormap[4]
 return out
```

```
if __name__=="__main__":
    trainData = hs.getData(filename='D:\- Hyper Spec -\Hyper SPEC TRAIN.h5',
dat_idx=25*49, lab_idx=49)
    app = QtGui.QApplication(sys.argv)
    form = HyperSpecApp()
    form.show()
    form.update() #start with something
    img = trainData['dcb'][:, :, 0:25] #fromn red test 343:370
    img1 = np.swapaxes(img, 2, 0)
    form.ImageView2.setImage(img1)
    form.ImageView2.export("Pre FFT Masking .png")
    #create FFT Plot
    fft example = hs.hsfft(img1)
    log_fft = np.log2(fft_example)
    aaa = ft.fftshift(log fft.real)
    form.ImageView3.setImage(aaa)
    form.ImageView3.export("FFT_DCB_.png")
   mask = hs.genMask(offset=41)
    #form.ImageView3.setImage(mask, levels=[np.amin(mask),np.amax(mask)+.0001])
    #ImageView doesn't seem to display binary arrays very well so add a small value.
    out dcb = hs.dcbFilter(img)
    form.ImageView1.setImage(out dcb.real)
    form.ImageView1.export("Post FFT Masking .png")
    gtbatch = adjustLabels(trainData['classLabels'])
    gt = gtbatch
   t = np.swapaxes(out dcb, 0, 2)
    t = np.swapaxes(t, 0, 1)
    fftImg = t.real.astype(np.float32, copy=False)
    print('SHAPE OF INPUT IMG: ' + str(np.shape(img)))
    print('SHAPE OF FFT OUT: ' + str(np.shape(fftImg)))
    (m, c) = runKmeans(form.pltView1, fftImg)
    (mm, cc) = runKmeans(form.pltView2, img)
   view cube(fftImg)
    (raw results, raw Errors) = runSpectral(img, gt, title="Raw")
    (fft_results, fft_Errors) = runSpectral(fftImg, gt, title="FFT")
    (raw_pc_results, raw_pc_Errors, raw_pc) = runPCA(img, gt, title="Raw")
    (fft_pc_results, fft_pc_Errors, fft_pc) = runPCA(fftImg, gt, title="FFT")
    print('SHAPE of results: ' + str(np.shape(fft_pc_results)))
    print(fft pc results)
    xdata = fft pc.transform(fftImg)
    w = view nd(xdata[:, :, :5], classes=gt.astype(np.int8, copy=False),
title="FFT_DCB PCA Components")
   ydata = fft_pc.transform(img)
    w = view nd(ydata[:, :, :5], classes=gt.astype(np.int8, copy=False), title="DCB
PCA Components")
```

```
# perform mathematical morphology operations to reduce noise in results
   # convert each class to binary images then recombine a the end
   rbc_img = fft_pc_results == 2.0
   wbc img = fft pc results == 1.0
   nuc_img = fft_pc_results == 3.0
   bkg_img = fft_pc_results == 4.0
   (wbc_o, wbc_c) = cleanResults(wbc_img)
    (rbc_o, rbc_c) = cleanResults(rbc_img)
    (nuc o, nuc c) = cleanResults(nuc img)
   (bkg_o, bkg_c) = cleanResults(bkg_img)
   #open_rbc = ndimage.binary_opening(rbc_img)
   #clse rbc = ndimage.binary closing(open rbc)
   #print(rbc_img)
   ti = combineLabels(rbc_o, wbc_o, nuc_o, bkg_img)
   color = np.array([[0, 0, 0], [255, 0, 0], [0, 255, 0], [0, 0, 255], [255, 255,
0]], dtype=np.ubyte)
   tis = create_rgb(ti, color)
   #calculate accuracy for each class.
   v6876 = imshow(tis, title="Cleaned FFT PCA GT Result")
   pylab.savefig("Cleaned FFT PCA GT Result.png", bbox_inches='tight')
   fft classes = create_training_classes(fftImg, gt, True)
   fft_means = np.zeros((len(fft_classes), fftImg.shape[2]), float)
   app.exec ()
   print("DONE")
```

This is the main program that draws a simple GUI and generates images for the user to analyze and record results to the program directory. It calls several helper functions from programs listed in Appendix C which handle most of the heavy computation.

## Appendix E: Rebellion Photonics Arrow Snapshot Hyperspectral Imager Product Specifications:

Tables containing all product specifications available regarding the snapshot hyper-sptectral imaging system used.

Features	Value
Optical Resolution	Optimized for Diffraction limit
Output	Base Camera Link
Display	Easy-to-use Interface
Input Power	12 VDC (10V - 15V)
Power Requirements	Consumption = 3.3 W
Dimensions	17 x 5.6 x 5.7 inches
Weight	10 lbs

Specifications	Value
Operating Temperature	-30 °C to +60 °C
Storage Temperature	-40 °C to +70 °C
Operating Humidity	0 to 100% RH
Vibration	Shock 10G (20 - 200) HZ XYZ, 70 G
System Software	32-bit Windows® XP, Windows® Vista,
	Windows ® 7
Data Format	ENVI HDR (raw binary with text description
	file), 16-bit tiffs
Usable Spectral Range	400- 750 nm
Snapshot Spectral Windows	Prism 1 - VIS1 (400 - 500), VIS2 (450 - 650)
	Prism 2 - VNIR1 (400-800)
Spectral bands	40
Spatial bands	500 x 300
Peak Quantum Efficiency	50% (550)
Avg. Dispersion / pixel	VIS1 (2.75 nm), VIS2(5nm), VNIR1 (10nm)
Stray Light	Minimized

Image Acquisition	Value
Dynamic Range	12 bit
Frame Rate	5.0 (Full Quality)
Computer	Laptop or Desktop
Light Efficiency	85% over complete spectrum
Max. Single Frame Capture	15 microseconds
Integration	Up to 16 seconds
Programmable External	3 trigger sources, 5 triggering
Trigger	modes
Capture Mode	Snapshot and Video

# **Appendix F: Product Specifications for SCHOTT ACE Light Source**

The specifications of the illumination lamps used. Not shown is the gooseneck fiber optic attachment used to guide the illumination output into the sample on the microscope stage.



ACE® Light Source with EKE lamp, A20500

ACE® Light Source Models		
115 Volt	230 Volt	Description
Standard Light Sources		
A20500	Custom	EKE Lamp
A20500.1	A20510.1	EJA Lamp
A20500.2	A20510.2	DDL Lamp
Light Sources with Iris Diaphragm		
A20520	A20530	BKE Lamp
A20520.1	A20530.1	EIA Lamp
A20520.2	A20530.2	DDL Lamp
Light Source without Modulamp® Unit		
A20500.115	A20510.230	No Lamp Power Supply Only

B/E, DDL, and EA lamps and Modulamp® units can be purchased separately.

Modulamp® Unit with Iris Diaphragm

### ACE<sup>®</sup> Light Source

Compact, rugged, AC halogen light source with solid state dimmer for variable light intensity and maximizing lamp life.

#### Standard Features

- Solid state dimmer for continuous control zero to full intensity.
- Heavy duty, grounded metal housing.
- 115 or 230 input voltage versions available.
- IR interference filter for cool illumination.
- Full illumination on .60" active diameter input.
- Interchangeable Modulamp® Units offer 2 positions to maximize individual lamp efficiencies.
- Choice of three, 150 watt halogen lamps EA, BKE, and DDL for intense, uniform illumination.
- Built-in thermal shut down.
- Quiet rear exhaust.
- = CSA approved to UL spec, CE compliant to low voltage and EMC directives.
- Low housing temp slightly above ambient.
- Detachable IEC cord.

#### Options

- Shutter module.
- Private labeling available.
- Infinite setting, twelve blade iris diaphragm version available to reduce intensity while maintaining color temperature.

Power Supply Specifications	
Power Consumption	190 Watts (nominal)
Output Voltage	0 - 21 VAC
Fiber Receptacle	.720 (13.3 mm)
Input Voltage	115 VAC / 60 Hz
	230 VAC / 50/60 Hz
Temperature Range: Operating	0° to 50°C



### ACE<sup>®</sup> Light Source



Input: 115 VAC/60 Hz, 230 VAC/50/60 Hz, Output: 0-21 VAC





Lighting and Imaging SCHOTT AG Otto-Schott-Str. 2 55127 Mainz Germany Phone: +49 (0) 6131/66-7752 Fax: +49 (0) 6131/66-7850 E-mail: lightingimaging@schott.com www.schott.com/lightingimaging

Modulamp®... SCHOTT Innovation! Two positions optimize light intensity output, depending on the lamp in use. Entire housing is interchangeable for quick setup.

Choice of Lamps. SCHOTT Versatility! Three different bulbs (EIA, BKE, DDL) offer light intensity and color temperature options to maximize productivity and accuracy.

### +.005 Ø.720<sup>-0</sup> (Ø18.3 +.13) cĘ 7.4"



(188)

8.6\* (218)



Solid State Dimmer..SCHOTT Control! Continuous intensity control for lamp life management.

SCHOTT glass made of ideas

#### Product improvements may result in specification or feature changes without notice.

## **Appendix G: Hardware Specifications**

All tests were performed on a workstation computer equipped with a 12 Core Intel Core i7 3930K running at 3.2GHz, 32GB DDR3 Memory running two nVidia GTX 780 GPUs.

Workstation Specifications		
Processor	Intel Core i7 3930K 12 Core Socket 2011 CPU	
Memory	32 GB Quad Channel DDR3 @1866 Mhz	
Graphics Card	2x SLI nVidia GTX 780 3GB GDDR5 @800MHz	
Operating System	Windows 8.1/10 and Ubuntu 14.04	
HDD	512GB Samsung Pro SSD	

**Table 16:** Workstation performance specifications used to process image data in this thesis work.

The data used in this research was prepared and collected by the author and participating members of the WVU LCSEE Nanophotonics research group. The HDF5 datasets are available upon request as combined they are roughly 15GB in size.

All python scripts and readmes written are hosted on a private github repository and can be released upon request.