Fatma ÇELEBİ

A Ph.D. Thesis

AGU 2023

IMAGE PROCESSING BASED ANALYSIS AND QUANTIFICATION OF MICRO BIOMATERIALS AND CELLS FOR BIOCHIP

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Fatma ÇELEBİ
May 2023

IMAGE PROCESSING BASED ANALYSIS AND QUANTIFICATION OF MICRO BIOMATERIALS AND CELLS FOR BIOCHIP

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Fatma ÇELEBİ
May 2023

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Fatma ÇELEBİ
Signature :

REGULATORY COMPLIANCE

Ph.D. thesis titled Digital Image Processing Techniques for Biochips has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering and Science.

Prepared By
Fatma ÇELEBİ

Advisor
Assoc. Prof Kutay İÇÖZ

Head of the Electrical and Computer Engineering Program
Assoc. Prof Zafer AYDIN

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled Digital Image Processing Techniques for Biochips prepared by Fatma ÇELEBİ has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering and Science.

…../ …../ 2023

JURY:

Advisor: Assoc. Prof Kutay İÇÖZ

Member: Prof. Bülent YILMAZ

Member: Prof. Mustafa Ersel KAMAŞAK

Member: Assist. Prof. Gülay YALÇIN ALKAN

Member: Assist. Prof. Şerife AYAZ GÜNER

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering and Science, Executive Board dated ….. /….. / ……….. and numbered …………..……. .

……….. /……….. / ………..

(Date)

Graduate School Dean
Prof. İrfan ALAN

ABSTRACT

MAGE PROCESSING BASED ANALYSIS AND QUANTIFICATION OF MICRO BIOMATERIALS AND CELLS FOR BIOCHIP

Fatma ÇELEBİ

Ph.D. in Electrical and Computer Engineering

Advisor: Assoc. Prof Kutay İÇÖZ

May 2023

Quantification of tumor cells is essential for early cancer detection and progression tracking. Multiple techniques have been devised to detect tumor cells. In addition to conventional laboratory instruments, several biochip-based techniques have been devised for this purpose. Our biochip design incorporates micron-sized immunomagnetic beads and micropad arrays, necessitating automated detection and quantification not only of cells but also of the micropads and immunomagnetic beads. The primary function of the biochip is to simultaneously acquire target cells with distinct antigens. As a readout technique for the biochip, this study devised a digital image processing-based method for quantifying leukemia cells, immunomagnetic beads, and micropads. Images were acquired on the chip using bright-field microscopy with image objectives of 20X and 40X. Conventional image processing methods, machine learning methods, and deep learning methods were used to analyze the images. To quantify targets in the images captured by a bright-field microscope, color- and size-based object recognition and machine learning-based methods were first implemented. Secondly, color- and size-based object detection and object segmentation methods were implemented to detect structures in bright-field optical microscope images acquired from the biochip. Third, segmentation of the minimal residual disease (MRD) was achieved by using deep learning. Implemented biochip images comprised of leukemic cells, immunomagnetic beads, and micropads. Moreover, mesenchymal stem cells (MSCs) are stem cells with the capacity for multilineage differentiation and self-renewal. Estimating the proportion of senescent cells is therefore essential for clinical applications of MSCs. In this study, a self-supervised learning (SSL)-based method for segmenting and quantifying the density of cellular senescence was implemented, which can perform well despite the small size of the labeled dataset.

Keywords: Leukemia Cells; Image-Processing; Machine Learning; Deep Learning; Immunomagnetic Beads; Support Vector Machine; MRD Biochip; Bright-Field Microscope; Semantic Segmentation; Transfer Learning; Self-Supervised Learning; Instance Segmentation; Cellular *Senescence*

ÖZET

BİYOÇİPLER İÇİN MİKRO BİYOMALZEMELERİN VE HÜCRELERİN GÖRÜNTÜ İŞLEME YÖNTEMLERİ İLE OTOMATİK OLARAK SAYILMASI VE ANALİZİ

Fatma ÇELEBİ
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Doktora
Tez Yöneticisi**:** Doç.Dr. Kutay İÇÖZ
Mayıs-2023

Tümör hücrelerinin miktarının belirlenmesi, kanserin erken tespiti ve ilerlemesinin izlenmesi için çok önemlidir. Tümör hücrelerini saptamak için çeşitli yöntemler geliştirilmiştir. Standart tezgah üstü cihazların yanı sıra, bu amaç için çeşitli biyoçip tabanlı yöntemler geliştirilmiştir. Biyoçip tasarımımız, mikro ped dizileri ile birlikte mikron boyutunda immünomanyetik boncuklar içerir, bu nedenle yalnızca hücrelerin değil aynı zamanda mikro pedlerin ve immünomanyetik boncukların otomatik olarak algılanmasını ve ölçülmesini gerektirir. Biyoçipin temel amacı, farklı antijenlere sahip hedef hücreleri aynı anda yakalamaktır. Bu çalışmada, biyoçip için bir okuma yöntemi olarak lösemi hücrelerini, immünomanyetik boncukları ve mikro pedleri ölçmek için dijital görüntü işleme tabanlı bir yöntem geliştirilmiştir. Çipte, parlak alan mikroskopi ile 20X ve 40X görüntüler elde edildi. Görüntüler, geleneksel görüntü işleme yöntemleri, makine öğrenmesi yöntemleri ve derin öğrenme yöntemleri ile analiz edilmiştir. Parlak alan mikroskobu ile kaydedilen görüntülerdeki hedefleri ölçmek için renk ve boyut tabanlı nesne tanımlama ve makine öğrenme tabanlı yöntemler uygulandı. Parlak alan optik mikroskobu ile biyoçipten elde edilen görüntülerdeki yapıları tespit etmek için renk ve boyut tabanlı nesne algılama ve nesne bölütleme yöntemleri uygulandı. Lösemik hücreler, immünomanyetik boncuklar ve mikro pedlerden oluşan MRD Biochip görüntülerine derin öğrenme tabanlı segmentasyon uygulandı. Ayrıca, çok soylu farklılaşma ve kendini yenileme potansiyeline sahip mezenkimal kök hücrelerde, yaşlanan hücrelerin yüzdesinin tahmin edilmesi, klinik uygulamalar için kritik öneme sahiptir. Bu çalışmada, sınırlı boyutta etiketlenmiş veri kümesiyle bile verimli bir şekilde performans gösterebilen hücresel yaşlanma yoğunluğunu bölümlemek ve ölçmek için kendinden denetimli öğrenme tabanlı yöntem uygulandı.

Anahtar kelimeler: Lösemi Hücreleri; Görüntü İşleme; Makine Öğrenmesi; Derin Öğrenme; MRD Biyoçip; Semantik Segmentasyon; Transfer Öğrenimi; Kendinden Denetimli Öğrenme; Parlak Alan Mikroskopu; Örnek Segmentasyonu; Hücresel Yaşlanma; Magnetik Boncuk

Acknowledgements

Firstly, I would like to thank my supervisor Assoc. Prof Kutay İÇÖZ very much. All the time, he listened me and helped me during my long Ph.D. study. He taught me how to be a good researcher with patience. It was my chance to be his Ph.D. student.

I would like to thank my co-advisor Assist. Prof. Kasım TAŞDEMİR. He was really kind and patient every time. He has great knowledge about my research topic and he helped me a lot all the time. It was my chance to be his Ph.D. student.

I would like to thank my thesis committee: Prof.Dr. Bülent YILMAZ and Assist.Prof. Şerife Ayaz GÜLER, for their questions and correct guidance that assisted me during my studies.

Sincerely, I would like to thank my parents Halim USLU and Döndü USLU, my older sister Raziye TAŞKIN and her sister Nazile and Fatma TAŞKIN for their endless support. They were with me materially and spiritually.

Finally, I would like to thank my husband, Akif and my child Talha. Without their patient and love, I would not have completed my study. Thanks to you, I was able to earn this degree. I am really happy to have such a family like you.

This Ph.D. study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK), grant number of 115E020.

TABLE OF CONTENTS

1. INTRODUCTION..1

1.1 MOTIVATION OF THE STUDY .... …………..1

1.2 OVERVIEW OF THE IMAGE PROCESSING TECHNIQUES ON MEDICAL IMAGES ... ……4

1.2.1 Digital Image Processing Techniques *5*

1.2.2 Machine Learning Techniques.. *7*

1.2.3 Deep Learning Techniques *9*

2. AUTOMATED QUANTIFICATION OF IMMUNOMAGNETIC BEADS AND LEUKEMIA CELLS FROM OPTICAL MICROSCOPE IMAGES 18

2.1 INTRODUCTION.. 18

2.2 EXPERIMENTAL SECTION 21

2.2.1 Mateirlas and Methods *21*

2.2.1.1 Detection and Quantification of Immunomagnetic Beads and Cells in 40X Objective Images.. *22*

2.2.1.2 Detection and Quantification of Immunomagnetic Beads and Cells in 20X Objective Images.. *24*

2.3 RESULTS 28

2.3.1 Bead Quantification... *28*

2.3.2 Cell Quantification *29*

2.4 DISCUSSION AND CONCLUSION 36

3. IMAGE-ANALYSIS BASED AND DEEP LEARNING BASED READOUT METHODS FOR BIOCHIP: AUTOMATED QUANTIFICATION OF IMMUNOMAGNETIC BEADS, MICROPADS AND LEUKEMIA CELLS...38

3.1 INTRODUCTION.. 38

3.2 EXPERIMENTAL SECTION 41

3.2.1 Materials and Methods *41*

3.2.1.1Micropad Detection and Quantification.. *42*

3.2.1.2 Immunomagnetic Bead Detection and Quantification................................ *43*

3.2.1.3 Cell Detection and Quantification... *45*

3.2.1.4 Cluster Detection and Quantification... *46*

3.2.1.5 Detection of Cell and Cluster Attached to the Micropads *47*

3.2.1.6 Image Annotation... *47*

3.2.1.7 Overview of MRD Biochip Image Segmentation Workflow........................ *48*

3.2.1.8 Semantic Segmentation with U-Net Architecture *49*

3.2.1.9 U-Net, DeepLabV3 and FCN Architectures with Python........................... *51*

3.2.1.10 Quantification of Cells and Clusters *52*

3.3 RESULTS 53

3.3.1 Quantification of Micropads and Immunomagnetic Beads *54*

3.3.2 Quantification of Cells and Clusters Attached to the Micropads.................. *54*

3.3.3 Quantification of Cells for Cell Culture Images.. *54*

3.3.4 Cells Detection and Quantification for Patient Dataset................................ *55*

3.3.5 Segmentation Result of U-Net Architecture... *57*

3.3.6 Segmentation Results of DeepLabV3 and FCN Architectures....................... *60*

3.3.7 Automated Quantification After Segmentation *61*

3.4 DISCUSSION AND CONCLUSION 62

4. IMPROVING MULTI-CLASS SENESCENT CELL SEGMENTATION USING SELF-SUPERVISED LEARNING ON BRIGHT-FIELD MICROSCOPY IMAGES 65

4.1 INTRODUCTION........ 65
4.2 EXPERIMENTAL SECTION 68
  4.2.1 Materials and Methods 68
  4.2.1.1 Instance Segmentation with Mask R-CNN........ 68
  4.2.1.2 Instance Segmentation with Mask R-CNN and SSL........ 70
  4.2.1.3 Implementation Details of Mask R-CNN with Transfer Learning from MSCOCO Dataset........ 71
  4.2.1.4 Implementation Details of Mask R-CNN with Self-Supervised Learning and Fine-Tuning 73
4.3 RESULTS 74
4.4 CONCLUSION........ 78

5. CONCLUSIONS AND FUTURE PROSPECTS........ 79

5.1 CONCLUSIONS 79
5.2 SOCIETAL IMPACT AND CONTRIBUTION TO GLOBAL SUSTAINABILITY........ 81
5.3 FUTURE PROSPECTS 82

LIST OF FIGURES

Figure 1.1 General view of leukemia. Leukemia has a lot abnormal white blood cells …..……2

Figure 1.2 Examples of microfluidic platforms for liquid biopsy ………………………………3

Figure 1.3 General view of biosensors/biochips ………………..………………………………3

Figure 1.4 Examples of structuring elements. The dots show center of structuring element……6

Figure 1.5 Workflow of classic machine learning techniques………………………………......7

Figure 1.6 Workflow of supervised learning technique …………………………………........8

Figure 1.7 Visualization of overfitting, underfitting and balanced on training dataset………......9

Figure 1.8 Workflow of classic deep learning techniques…………………………………….....10

Figure 1.9 Usual deep learning components……………………………………………………11

Figure 1.10 Visualization of transfer learning approaches which is applied to medical images by transferring initial weights from public ImageNet dataset……………………………....12

Figure 1.11 Visualization of image classification task with deep learning………………………14

Figure 1.12 Visualization of image segmentation task [16]. a shows instance segmentation which identifies each instance of an object. b shows deep watershed segmentation which identifies different objects with bounding boxes. c shows application of a deep learning segmentation for breast cancer by Keren et al……………………………………………………………15

Figure 1.13 Schematic diagram of SimCLR framework……………………………………..16

Figure 2.1 The graph of the leukemic cells frequency in peripheral blood or bone marrow of patients after and during the chemotherapy. The graph also shows the detection limit of flow cytometric and morphological techniques………………………………………............19

Figure 2.2 Visualization of detection strategy with magnetic beads……………………………20

Figure 2.3 Typical optical micrographs of cells and beads recorded using A) 40× objective B) 20× objective. Scale bar indicates 20 µm. Blue arrow indicates single beads, pink arrow indicates the cell fragments, green arrow indicates single cell with attached immunomagnetic beads, red arrow indicates fully covered cells with beads as cluster

……………………………………………………………………………………..21

Figure 2.4 (A) Flow diagram of cell and bead detection and counting steps for the images recorded using 40× objective, (B) original RGB cell image, (C) YUV thresholded cell image, (D) RGB thresholded cell image, (E) YUV image after morphological operations ……………………………………………………………………………….....23

Figure 2.5 The flow diagram of the bead detection using Y channel in YUV color space from the images acquired by the 40× objective……………………………………………...24

Figure 2.6 Flow diagram of cell detection and counting steps for the images recorded using 20× objective……………………………………………………………………………...25

Figure 2.7 View of the HOG features histograms in our image…………………………......26

Figure 2.8 Original images (top row) and green detection marks for each cell (bottom row) for the 40× objective……………………………………………………………………......29

Figure 2.9 Comparison of high precision and low precision images. Blue circle indicates the blurry part of the image (top left) and resulted false positives (bottom left) ……….......30

Figure 2.10 The impact of image quality on the performance of the image processing algorithm. Original image (image 9) (left), cropped image without blurry part (right)……………….31

Figure 3.1 Schematic representation of a biosensor (biochip) and essential components……………………………………………………………………..39

Figure 3.2 a) Optical microscopy system with 20X objective and the microfluidic platform b) 20X objective biochip microscopy image which has small micropad. c) 20X objective biochip microscopy image which has bigger micropad areas than a. d) 20X objective biochip microscopy image which has larger micropad area than b and c. Blue arrow indicates cells attached to a micropad, purple arrow indicates a single cell, yellow arrow indicates two single cells with attached bead, red arrow indicates fully covered cells with beads as cluster, and green arrow indicates square pads. In addition, white arrow indicates the part of the biochip for capturing the image…………………………………………………………42

Figure 3.3 a) Original RGB image, b) a micropad from the original image, c) result of micropad extraction algorithm, image is in gray-scale, and d) binary image obtained by using Otsu's threshold on gray-scale image in c. Scale bars = 20 µm……………………………………..43

Figure 3.4 Flow diagram of immunomagnetic bead detection and quantification steps……......43

Figure 3.5 (A)Original RGB image, (B) gray-scale image, (C) filtered image, (D) binary image after Otsu's threshold, (E) masked gray-scale image, (F) binary image of masked image after Otsu's threshold, (G) binary image after watershed segmentation, and (H) binary micropad subtracted image. Scale bars = 20 µm………………………………………......................45

Figure 3.6 Result of the watershed algorithm. In the image there are some cells, micropad and immunomagnetic beads. Background is black. Watershed lines are visible between different objects and even in the same object………………………………………………………..46

Figure 3.7 Flow diagram of cluster finding and counting steps…………………………………47

Figure 3.8 Top row: The general view of image and its annotation. a) Original bright-field microscopy image obtained from the microfluidic platform. b) Cropped image c) Annotated cropped image, green, yellow, red, blue, and white colors, respectively, represent cell, cluster, micropad, immunomagnetic beads, and background. Bottom row: Clusters can appear in 4 different forms as indicated by arrows. d) Three beads, e) more than three beads are sticked together and the shape is circular. f) More than three beads sticked together, but shape is not circular, so it is not cluster. g) Less than three beads bounded to micropad and occluded the cell……………………………………………………………………………...48

Figure 3.9 Architecture of the pixel-wise classification network. The network was trained to classify the pixels into one of the 5 classes: cell, cluster, bead, micropad, background……………………...49

Figure 3.10 U-Net architecture. It has 3x3 Conv2d + BN + ReLU layers, Max Pooling 2x2 layers, 2x2 Transposed Conv2d layers, 2 Input Depth Concat, Softmax layer and Segmentation layer……………………………………………………………………………………..50

Figure 3.11 (a-c): Accuracy and Loss result graphs of U-Net with different hyper-parameters. (d): Parameter values for each training in a, b, and c graphs. The graph shows that with the given hyper-parameter selection, the model (c) can learn the class representations without any overfitting or underfitting of the data……………………………………………………51

Figure 3.12 Quantification of cells and clusters workflow……………………………………..53

Figure 3.13 Cell detection and quantification result examples. Images belong to different patient along with different day. Blue color shows detected cells while yellow color shows detected clusters. Given images are some parts of the whole images. A, B, C, and D are original RGB images, E, F, G, and H are final result images. Scale bars = 20 μm………………………55

Figure 3.14 Results of proposed algorithm on different patients. Each graph shows comparison of manual counting and automated counting for different patient………………………..57

Figure 4.1 MSCs sources and their usage in clinical studies…………………………………..66

Figure 4.2 Relationships between metabolism and cellular senescence……………………….66

Figure 4.3 General view of Mask R-CNN architecture used on our dataset. Left to right; input image, architecture of network and output image. It takes an input image and returns output image with mask of each instance and their bounding boxes……………………………69

Figure 4.4 The proposed self-supervised instance segmentation on microscopic images workflow. It comprises of three steps. First, applying Mask R-CNN on unlabeled high-resolution images to get single object patches. Second, unsupervised learning to learn general representations of our microscopy image patches. Third, fine-tuning and training on labeled dataset, and then instance segmentation by using Mask R-CNN architectures………………………………………………………………………………70

Figure 4.5 Effect of data augmentation on our dataset. Blue color indicates validation loss with augmentation, and red color indicates validation loss without augmentation…………….72

Figure 4.6 Test image samples and their Mask R-CNN without SSL segmentation results. Row (A) shows test images, and row (B) shows results of instance segmentation with masks, object labels, probability values, and bounding boxes……………………………………73

Figure 4.7 Training loss results on labeled dataset. SSL batch 16 indicates ImageNet initialized self-supervised learning with batch size 16 while pre-training, SSL batch 256 indicates ImageNet initialized self-supervised learning with batch size 256 while pre-training, SSL batch 256 end-to-end indicates self-supervised learning without any weights initialization in the proposed method. SSL improves instance segmentation performance………………76

Figure 4.8 Test image result with self-supervised learning and without self-supervised learning……………………………………………………………………………....77

LIST OF TABLES

Table 2.1 Summary of the 20X and 40X objective characteristics……………………………22

Table 2.2 All Extracted Features…………………………………………………………...26

Table 2.3 Average bead counting error rate values for images acquired by 40× and 20× objectives……………………………………………………………………………..29

Table 2.4 The impact of cropping out the blurry region of the image on the performance of the algorithm…………………………………………………………………...…...31

Table 2.5 Fully Covered Cell (Bead Cluster) Detection Results………………………………32

Table 2.6 Comparison of machine learning algorithms for each image for isolated and partially covered cell detection (in Weka)……………………………………………………...33

Table 2.7 Average accuracy of machine learning methods in Weka for isolated and partially covered cell detection…………………...……………………………………………… 34

Table 2.8 Cell counting results for each image……………………………………………. 35

Table 2.9 Summary Table for 20X images. Correctly Classified Accuracy =100*Correctly Classified / (Correctly Classified + Missed Detection)……………………………………35

Table 3.1 Total number of cropped images for each class to create a balanced dataset..………51

Table 3.2 Average cells and cluster attached to micropads detection and counting performance results..54

Table 3.3 Cell culture experiments: Average cell counting performance results for 20X objectives……………………………………………………………………………..55

Table 3.4 U-Net class metrics for each class…………………………………………………58

Table 3.5 U-Net segmentation average metrics for 5 classes……………………………………58

Table 3.6 U-Net segmentation average metrics with BN, and without BN layer.....……………...59

Table 3.7 U-Net segmentation class metrics with BN, and without BN layer.............................59

Table 3.8 Evaluation matrices result of U-Net + BN architecture with Pytorch……………….…60

Table 3.9 Evaluation matrices result of DeepLabV3-Resnet50 architecture…………………....60

Table 3.10 Evaluation matrices result of DeepLabV3-Resnet101 architecture…..……………...60

Table 3.11 Evaluation matrices result of FCN-Resnet50 architecture…...………………………61

Table 3.12 Evaluation matrices result of FCN-Resnet101 architecture…..……………………...61

Table 3.13 Overall performance comparison between 5 different semantic segmentation architectures……………………………………………………………………………….61

Table 3.14 Overall performance of quantification of cells and clusters on test images………...62

Table 3.15 Comparison of quantification results between our previous study and proposed study on test images. Improvement rate is calculated as (Proposed method's result – the previous method's result) / |ideal result - previous result| * 100…………………………………....62

Table 4.1 Number and properties of images used for self-supervised learning and fine-tuning…71

Table 4.2 Evaluation results of Mask R-CNN on our microscopy test images with fine-tuning from MSCOCO dataset and self-supervised learning…………………..………………75

Table 4.3 Comparison of the methods by total number of correctly found cells..………………77

Table 4.4 Quantitative results of Performance Assessment Procedures (Mean, [min, max]) for Proposed Approach on microscopy cell images..…………………………………………78

LIST OF ABBREVIATIONS

ML	Machine Learning
DL	Deep Learning
RIO	Region of Interest
SVM	Support Vector Machine
HCA	Hierarchical Cluster Analysis
ReLU	Rectified Linear Unit
SimCLR	Simple Framework for Contrastive Learning
MoCo	Momentum Contrast for Unsupervised Visual Representation Learning
MICLe	Multi-Instance Contrastive Learning
PCR	Polymerase Chain Reaction
NSCLC	Non-Small Cell Lung Cancer
PANC	Pancreatic Cancer
HPS	Horseradish Peroxidase
GLCM	Gray Level Co-occurrence Matrix
HOG	Histogram of Oriented Gradient
LBP	Local Binary Pattern
SVM	Support Vector Machine
TP	True Positive
FP	False Positive
FN	False Negative
MRD	Minimal Residual Disease
FC	Flow Cytometry
ECM	Extracellular Matrix
SEM	Scanning Electron Microscope
RBCs	Red Blood Cells
WBCs	White Blood Cells
CNN	Convolutional Neural Network
k-NN	k-Nearest Neighbor
BN	Batch Normalization
FCN	Fully Convolutional Network
ACC	Accuracy

IoU	Intersection Over Inion
MSCs	Mesenchymal Stem Cells
SASP	Senescence-Associated Secretory Phenotype
b-Gal	Beta-Galactosidase Assay
GMP	Good Manufacturing Practices
SSL	Self-Supervised Learning
FBS	Fetal Bovine Serum
FPN	Feature Pyramid Network

To my lovely family…

Chapter 1

Introduction

1.1 Motivation of the Study

The cancer process starts with risk evaluation on healthy blood, and after cancer screening, disease determination is completed, and based on the disease, therapy selection is done. Then, the selected therapy is applied to the patient, and the results are monitored to analyze whether the therapy is suitable or enough for the patient or not. After treatment, follow-up care studies are done with clinical and microscopic examination. If treatment is suitable for the patient, the patient achieves remission and does not show any cancer symptoms. However, after treatment, a small number of cancer cells that do not show any symptoms and are resistant to treatment may remain, and those cells can cause a relapse of the cancer. Those remaining cells after treatment are called as minimal residual disease (MRD). MRDs cannot be detected by cancer scans or tests [1]. Relapse still occurs in a significant proportion of patients, that's why MRD must be monitored during and after treatment.

Leukemia is the most common cancer type in childhood (in the US, UK, and Turkey). When a person becomes a leukemia patient, his bone marrow produces an excessive amount of white blood cells that don't function properly. Normal blood and leukemia samples are shown in Figure 1.1. There are different therapies for leukemia, such as drug therapy (chemotherapy), bone marrow transplantation, radiation treatment, and immunotherapy. The most commonly used therapy is chemotherapy. Not all leukemia patients have the same response to chemotherapy. Since after chemotherapy, some leukemia cells may remain and cause repeating leukemia, the number of cancer cells, which is MRD, must be monitored.

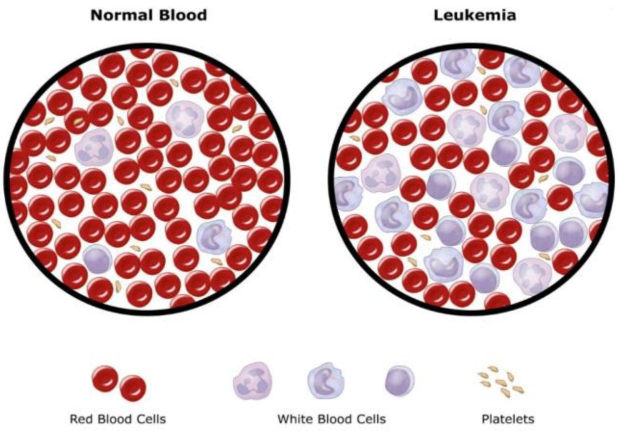
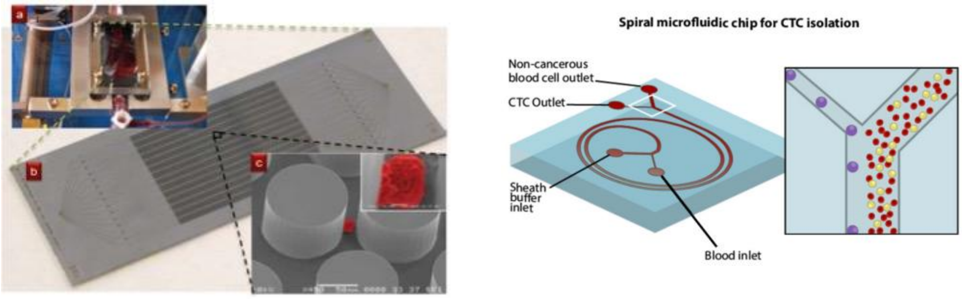


Figure 1.1 General view of leukemia. Leukemia has a lot abnormal white blood cells [2]**.**

Mesenchymal stem cells (MSC) are used in clinical studies. When a stem cell starts aging, it causes negative effects instead of positive ones, so before using MSCs in clinical studies, their cellular senescence density should be identified. Cellular senescence in MSCs is estimated manually by using beta-galactosidase (SA-β-Gal) activity. After activity, the number of senescent cells is calculated manually.

Currently, MRD is monitored by flow cytometry (FC) and polymerase chain reaction (PCR) devices. Those devices are widely used, and they estimate MRD accurately. On the other hand, they are not suitable for all laboratory conditions, they need further standardization, and they need a high level of expertise [3].

There are new alternative microfluidic platforms that are used for liquid biopsy in the literature [4]. Microfluidic platforms are used to capture and isolate the target objects from the environment. Some of the examples of microfluidic platforms for liquid biopsy are shown in Figure 1.2. Microfluidic systems are low-cost and enable integration, automation, and parallelization [5].


a
b
c
Spiral microfluidic chip for CTC isolation
Non-cancerous blood cell outlet
CTC Outlet
Sheath buffer inlet
Blood inlet

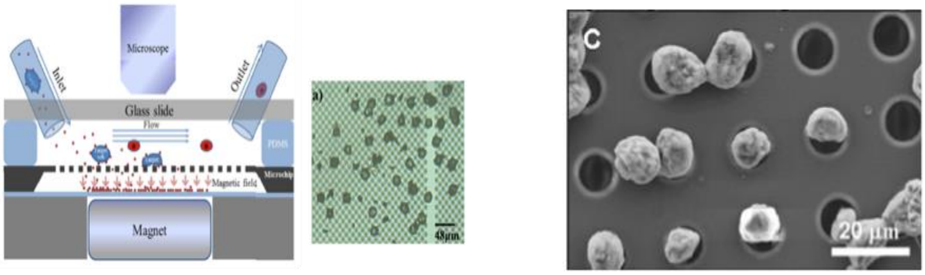


Figure 1.2 Examples of microfluidic platforms for liquid biopsy [4], [6]–[8]**.**

Biosensors and biochips consist of a receptive layer, and the receptive layer contains biomolecular receptors to capture the target molecules from the environment. Bio-molecular receptors can be enzymes, antibodies, or cells, and those receptors produce a physicochemical reaction with target molecules to capture by showing a key-lock relationship. After capturing, biochips produce an output. Finally, on this output, data analysis is done and meaningful information is gathered. A general view of the biosensors is given in Figure 1.3.

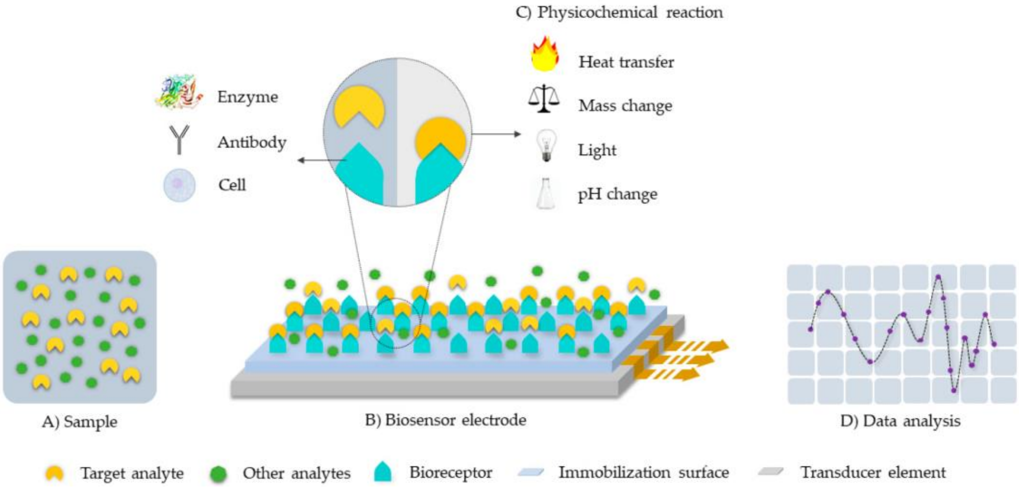


Figure 1.3 General view of biosensors/biochips [9]**.**

In addition, immunomagnetic beads are used to capture and separate target molecules from complex biological medium since they have the magnetic property to apply external magnetic force and can be coated with antibodies, aptamers, or peptides to bind target molecules. Magnetic beads ensure resuspension of target molecules in smaller volumes because they can be nano or microsized and are integrated with biosensor platforms. Magnetic beads are used to capture breast cancer cells, non-small lung cancer cells, and pancreatic cancer cells from blood samples. After capturing, loaded and unloaded beads, which are sent to a size-based micro filter, are distinguished from each other by applying magnetic force. Then cells on the chip surface are counted using fluorescent images, which require staining steps [10]. However, staining requires secondary antibodies, and this situation increases the complexity and cost of the process.

In this study, we propose image processing techniques to quantify MRD in microfluidic biochip images. Leukemia cells are captured and isolated from the sample with immunomagnetic beads, and then, using bright-field microscopy, images are taken from a microfluidic platform to analyze.

In addition, manual counting of MSCs causes some problems. For example, some aged cells don’t show beta-galactosidase activity, which increases the false negative rate. In addition, manual counting may result in operator bias. Furthermore, senescent cells show some distinct features compared to young cells, like enlarged and flattened shapes and more than one nucleus. Those features can be used during automated detection and counting of senescent cells with image processing techniques. Below is a broad overview of image processing techniques for medical image studies.

1.2 Overview of the Image Processing Techniques on Medical Images

The existence of medical imaging science started in 1895 with the discovery of X-rays by W.C. Röentgen. However, computers in medical imaging owed its existence to G. Hounsfield with practical computed tomography scanners in 1970. After that, computers have been used as components of medical imaging systems for data acquisition, image generation, image display, and image analysis.

The growth in the number of medical imaging modalities led to an increase in the number of medical image datasets, which included large scope and complex problems, and this required advanced techniques for their analysis and solution. Imaging techniques can host noise, blurring, and illumination effects, and this leads to poor image quality and complex problems to analyze. In addition, medical images are visual in nature, but human analysis of those images has some limitations. Those limitations are associated with expert bias, environmental distractions, a low rate of incidence, etc. Consequently, image processing and computer vision techniques are developed for quantitative and objective analyses of medical images [11].

However, it is not an easy task to develop an algorithm for medical images since they may have variability in features and dataset size. It means that a feature that exists in one image may not appear in another image, and one system may not be suitable for another dataset due to its size. As a result, a system that works well on certain image sets may not work in another similar situation, so it is really crucial to develop an appropriate system for the given problem. Medical images can contain image enhancement, segmentation, classification, or image restoration problems. There are some techniques that are applied to medical image analysis problems: (1) digital image processing techniques; (2) machine learning techniques; and (3) deep learning techniques.

1.2.1 Digital Image Processing Techniques

A medical image is a two- or three-dimensional matrix of a scene. It consists of rows and columns. Each row and column has a pixel value ranging from 0 to 255. Image processing is a technique to perform some operations on an image to get an enhanced image or to extract useful information from the image. Digital image processing is a method to manipulate digital images using computers. Image analysis involves taking an image that has objects and assigning them to groups and classes. While analyzing the image, it is important to isolate the object of interest from the rest to measure the size, shape, or texture of the object. This task is known as segmentation. In addition, it can be necessary to classify an object in the image into a certain group, such as benign or malignant. This task is known as classification or pattern recognition. There are some digital image processing techniques to segment or classify objects in medical images.

1.2.1.1 Thresholding

Thresholding is a method that separates the foreground pixels from the background pixels. It plays an important role in image segmentation. It is easy to implement, and it has a high computation speed. The thresholding is applied to each pixel value in the given image by selecting a threshold value to find an object of interest. According to the given threshold value, if the pixel value is larger, value 1 is assigned to that pixel. If the pixel value is smaller than the threshold value, a value of 0 is assigned to that pixel. This type of thresholding is called global thresholding, and the equation is given as follows;

$$\boldsymbol{g(x,y)} = \begin{cases} \boldsymbol{1} & \boldsymbol{if\ f(x,y) > T} \\ \boldsymbol{0} & \boldsymbol{if\ f(x,y) \leq T} \end{cases} \tag{1.1}$$

where $g(x,y)$ represents the new binary image, which has object point and background point, pixel value, $f(x,y)$ represents original image pixel value, and T represents threshold value. For this method, deciding the optimum threshold value is important. If T value changes over the image, that method is called as variable thresholding. In addition, a threshold value can be found automatically based on the given image by Otsu's thresholding method [12].

1.2.1.2 Morphological Operations

The term morphology offers numerous image processing operations that process images based on shapes and structures[13]. Commonly, morphological operations are applied to binary images, but they can also be extended for use with gray-scale images. They use a structuring element to apply to a given image and create an output image of the same size. The shape and size of the structuring element vary based on the given problem, and it is important to find a suitable structuring element for the problem. Examples of structuring elements are given in Figure 1.4. Morphological operations are used for filtering, edge detection, feature detection, counting objects, segmentation, and noise reduction. Common operators are erosion, dilation, opening, and closing.

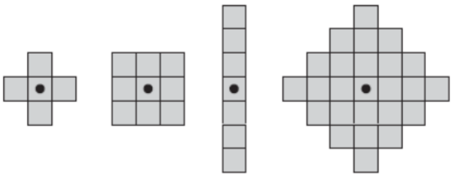


Figure 1.4 Examples of structuring elements. The dots show center of structuring element [14].

Erosion is an operation to shrink foreground objects. As a result, the operation sharpens the corners of objects and enlarges background features and holes. In contrast, dilation operations bridge gaps and connect objects. Besides, dilation operations can be used to create the outline of foreground objects in an image.

While opening is defined as applying erosion followed by dilation by using the same structuring element, closing is defined as applying dilation followed by erosion by using the same structuring element. Binary opening operations can be used as shape detectors to preserve certain shapes. Conversely, binary closing operations can be used to smooth the contours of foreground objects, narrow breaks or gaps, and eliminate small holes[15].

1.2.1.3 Image Enhancement

Before applying some methods to images, it may have been necessary to do some pre-processing operations to obtain better images. Some images can have artifacts like noise, blur effects, illumination effects, etc. Initially, those artifacts should be cleaned, and then other methods should be applied. The most commonly used methods for pre-processing the images are sharpening, blurring, histogram equalization, and filtering in the frequency domain or spatial domain.

1.2.1.4 Hough Transform

The Hough transform is a feature extraction method that is used in image analysis to isolate regular shapes like lines, circles, ellipses, etc. The method is computationally complex. For instance, in two-dimensional space, a circle is represented by the below equation;

$$\boldsymbol{(x-a)^2+(y-b)^2=r^2} \tag{1.2}$$

Where (a, b) is the center of the circle and (r) is radius, which results in a 3D parameter hough space. Then, the procedure is to increase a and b, solve for the r that satisfies equation (1.2). The complexity of the hough transform depends on the number of coordinates and coefficients in the given equation. Hough transform works well even when there are incomplete or distorted versions of circles. In [16], diameter distribution of collagen fibers was analyzed by hough transform since injured ligaments show a unique distribution while normal samples show a wide range of diameter.

1.2.2 Machine Learning Techniques

Machine learning (ML) is a part of artificial intelligence, and its main purpose is to learn like a human being by using data and algorithms. It learns from the given data. Machine learning techniques are iterative, and they can independently adapt to unseen data. Machine learning techniques are efficient for medical images. They are used for diagnosis, classification, prediction, and segmentation. Figure 1.5 shows the typical architecture of machine learning. Firstly, region of interest (RIO) images are segmented either automatically or manually and sent to the system. Then, hand-crafted features are extracted, and ineffective features are eliminated to decrease computation complexity and increase performance. Finally, based on the remaining features, algorithms are used for further prediction.

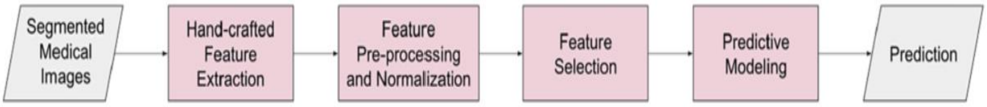

Segmented Medical Images
Hand-crafted Feature Extraction
Feature Pre-processing and Normalization
Feature Selection
Predictive Modeling
Prediction

Figure 1.5 Workflow of classic machine learning techniques [17].

Medical image data is large and diverse, so it is difficult to analyze. They need an expert in the field, and it can take a long time. However, machine learning techniques are used to reduce operator dependency, reduce time cost and get better performance. Machine learning has two approaches. 1) supervised learning; 2) unsupervised learning.

1.2.2.1 Supervised Learning

The supervised learning technique uses the labeled data during training to categorize the unseen test data. The model is built based on the extracted features of the labeled dataset. Because the labels of the training phase are known, supervised learning techniques outperform unsupervised learning techniques. It is crucial to label the data correctly and sufficiently to get good performance from the model. The workflow of supervised learning is shown in Figure 1.6. Labeled benign and malignant tumors are trained with extracted features, and then a model is obtained. After that, a model is used to predict new image classes.

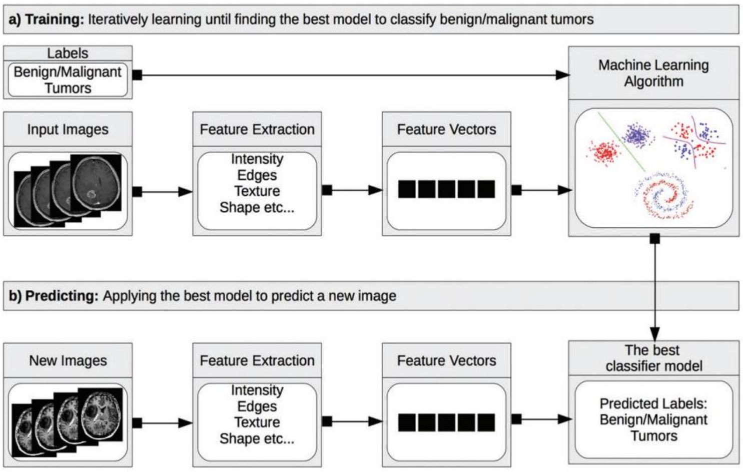

a) Training: Iteratively learning until finding the best model to classify benign/malignant tumors
Labels
Benign/Malignant Tumors
Machine Learning Algorithm
Input Images
Feature Extraction
Intensity
Edges
Texture
Shape etc...
Feature Vectors
b) Predicting: Applying the best model to predict a new image
New Images
Feature Extraction
Intensity
Edges
Texture
Shape etc...
Feature Vectors
The best classifier model
Predicted Labels: Benign/Malignant Tumors

Figure 1.6 Workflow of supervised learning technique [18]**.**

There are some models that are used in machine learning. They are k-nearest neighbor, support vector machine (SVM), random forests, and artificial neural networks. The model classifies the given input based on the training model, which is trained on extracted features from labeled data. It is important to choose the appropriate classifier based on your dataset and hardware system, since some architectures need strong hardware or are suitable for larger datasets. In addition, to get good performance from the system, it is crucial to have a balanced and sufficient dataset. Machine learning benefits from large amounts of data. Also, in the dataset, samples from each class must be balanced; otherwise, system performance decreases. There is some terminology that must be understood;

Overfitting refers to the case where a model learns noise and details from the training dataset, but because the model cannot generalize the dataset, it cannot fit unseen data. The clear sign of overfitting is a larger validation or testing error than a training error. Overfitting is also known as high variance.

Underfitting refers to the scenario where neither the model learns from the training dataset nor generalizes from the unseen test dataset. The clear sign of underfitting is poor performance on the training dataset. Underfitting is also called as high bias. A small number of datasets can cause underfitting.

Good-fit refers to the case where the model both generalizes well on the training dataset and shows good performance on the unseen test dataset. Good-fit is also known as balanced.
The visualization of the above terminology on the training dataset is given in Figure 1.7

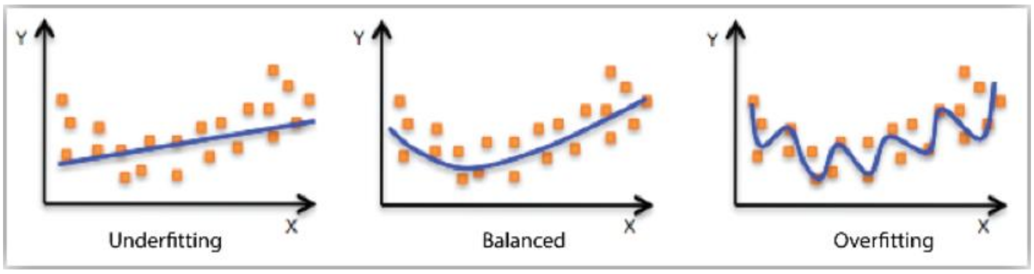


Figure 1.7 Visualization of overfitting, underfitting and balanced on training dataset [19].

1.2.2.2 Unsupervised Learning

Contrary to supervised learning, an unlabeled dataset is used while training the model. One of the techniques is clustering. K-means clustering is one of the most frequently used clustering methods. It is simple, it has significant speed, and it is efficient on large datasets [20]. The number of clusters in the problem is provided by the user initially. After that, algorithm randomly selects the centroids that are used as initial points for every cluster. Then, the algorithm iteratively calculates the distance of each centroid to the data points to optimize the positions of the centroids.

Another popular technique of supervised learning is hierarchical clustering, also known as hierarchical cluster analysis (HCA). This time, initially, the algorithm takes each sample as one distinct cluster and then computes the distance (most commonly the Euclidian distance) to measure the similarity of two clusters. Then it identifies the closest two clusters and merges them. This distance calculation and merging operation continue iteratively until all the clusters are merged.

1.2.3 Deep Learning Techniques

Deep learning (DL) is a subfield of machine learning that extracts image features automatically. It uses artificial neural networks to process raw data. It is computationally complex and needs powerful hardware. Improvements in computer hardware expand the use of deep learning techniques on images and other data types in the physical [21], chemical [22], medical [23], [24], and biological sciences [25]. There are some applications of deep learning: image classification, segmentation, object tracking, and augmented microscopy. Figure 1.8 shows a general view of the deep learning architectures. It can be seen that raw training images are sent to the system, then features are extracted automatically by neural network architectures, and finally, prediction is done on an unseen test dataset. There are two approaches to training: (1) start learning with initialized weights from a pre-trained model; (2) end-to-end learning that doesn't use any pre-trained model to initialize weights.

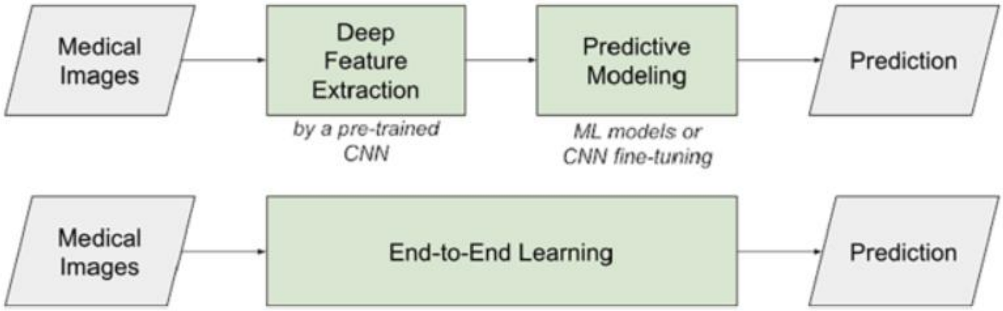


Figure 1.8 Workflow of classic deep learning techniques [17].

Like in machine learning, training datasets can be labeled or unlabeled. If a labeled dataset is used while training, it is called as supervised learning, or if an unlabeled dataset is used while training, it is called as unsupervised learning (self-supervised learning). It is really required to have sufficient high-quality data for successful applications of deep learning, but it can take a long time. Besides, since deep learning extracts features and learns from the raw data, image diversity is needed. While training, if dataset size is limited, there are common computational operations to increase dataset size, which are known as data augmentation and applied to raw datasets, such as rotation, flipping, cropping, and shearing. These operations can be applied regardless of the dataset size or type [26].

Deep learning has mathematical underlying. For example, there are gray-scale images, and their sizes are (M_x, M_y, 1), where M_x and M_y are number of rows and columns, and 1 is the number of channels in the image. Let’s assume that images are needed to classify them into two classes (1 for malignant, 0 for benign). In this case, a linear classifier is used, and during training, images are first converted to vectors. The linear classifier takes those vectors as input and produces class scores for each pixel value. Mathematically, it is given as;

$$\begin{bmatrix} \boldsymbol{y_0} \\ \boldsymbol{y_1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w_{0,0}} & \cdots & \boldsymbol{w_{0,M_xM_y-1}} \\ \boldsymbol{w_{1,0}} & \dots & \boldsymbol{w_{1,M_xM_y-1}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_0} \\ \vdots \\ \boldsymbol{x_{M_xM_y-1}} \end{bmatrix} = \begin{bmatrix} \sum \boldsymbol{w_{0,j}x_j} \\ \sum \boldsymbol{w_{1,j}x_j} \end{bmatrix} \tag{1.3}$$

Where y_0 and y_1 are the class scores, *w* is the class weights and *x* is the image vector. The class with the highest scores is taken as predicted class. Then, the training task calculates the loss function to measure the performance of the classifier. Generally, cross-entropy or softmax loss are used as loss functions. To measure the loss, class scores y_0 and y_1 are transformed into probabilities by

$$\boldsymbol{p_i} = \frac{e^{\boldsymbol{Class\ i\ score}}}{\sum_{\boldsymbol{All\ classes}} e^{\boldsymbol{Class\ score}}} \tag{1.4}$$

After probability transformation loss value of each image computed as;

$$\boldsymbol{Loss} = -\sum\nolimits_{\boldsymbol{Images}} \boldsymbol{log\, p_{Correct}} + .\, \boldsymbol{\lambda} \sum\nolimits_{\boldsymbol{i,j}} \boldsymbol{w_{i,j}^2} \tag{1.5}$$

Where $p_{Correct}$ is the correct class probability, and λ is L2 regularization value that was used to penalize the large weights in order to overcome overfitting. The minimization of the loss function is achieved by optimization algorithms. These algorithms are variations of stochastic gradient descent, which includes Adam [27], Adagrad [28], momentum [29]. It is important to choose a suitable optimizer for the given problem. In addition, in deep learning, there are some components that are commonly used. General components are given in Figure 1.9.

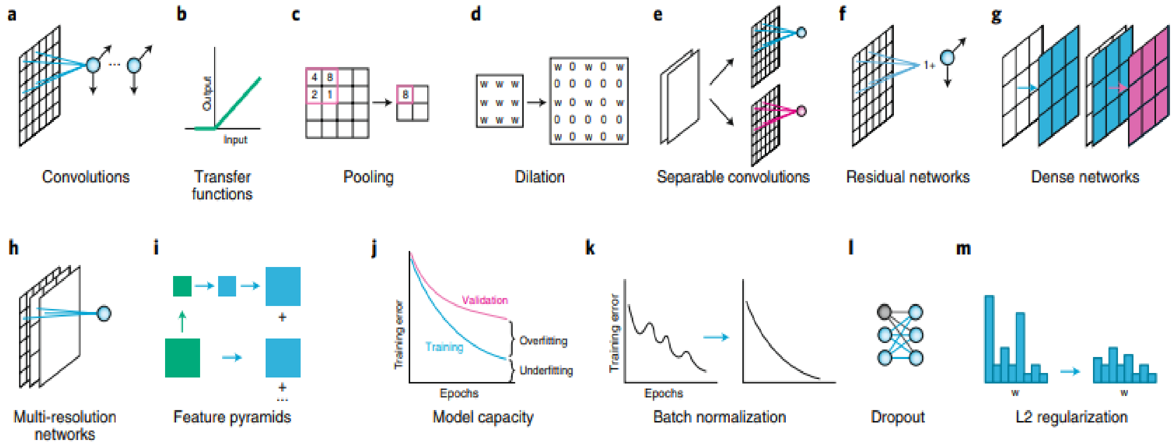


Figure 1.9 Usual deep learning components [26].

During the convolution, local features are extracted by a squared filter like 2x2 or 3x3. Filter size can change. After convolution, then, transfer function is applied to have a certain range after the convolution. Transfer function is also known as activation function, and rectified linear unit (ReLU) is a commonly used activation function. It takes the output of convolution as input, and if the value is negative, the function maps that value to zero; if the value is positive, the function takes the value as it is. After activation function, a pooling operation is applied to get coarse feature maps. There are two types of pooling: min pooling and max pooling. Max pooling technique is the most frequently used pooling technique. In deep learning, convolution, activation function and pooling operations are iteratively used to obtain low-dimensional versions of the original image. Separable convolutions, residual networks, dense networks, multi-resolution networks, and feature pyramids are several architectural elements of deep learning. They are not mandatory to use. A plot of training and validation loss shows overfitting, underfitting, or good-fit, so while training, plotting that graph is critical and helps to understand how training goes, where to stop training, or whether the used model is suitable to the given task or not. Generally, in deep learning, increasing model capacity decreases underfitting, but it can also cause overfitting. Increasing model capacity means adding more layers to the network. Furthermore, deep learning techniques have numerous regularization methods to learn general features from the data. Those methods are batch normalization, dropout, and L2 regularization. The batch normalization technique is used to regularize the networks and decrease training time [30]. The dropout technique randomly ignores some number of layers. Generally, batch normalization and dropout techniques are not used in the same model. Lastly, the L2 regularization technique decreases overfitting by penalizing large weights. By default, these three regularization methods don't exist in deep learning models, but they can be added later if needed. When the dataset size is limited, using regularization techniques makes the system more robust.

When dataset size is limited, another alternative approach to regularization techniques is transfer learning. Transfer learning is a very effective method that is frequently used in deep learning methods. For medical images, transfer learning from out-of-domain pre-training or from in-domain pre-training can be used. Figure 1.10 demonstrates how transfer learning techniques are used for medical image classification via deep learning.

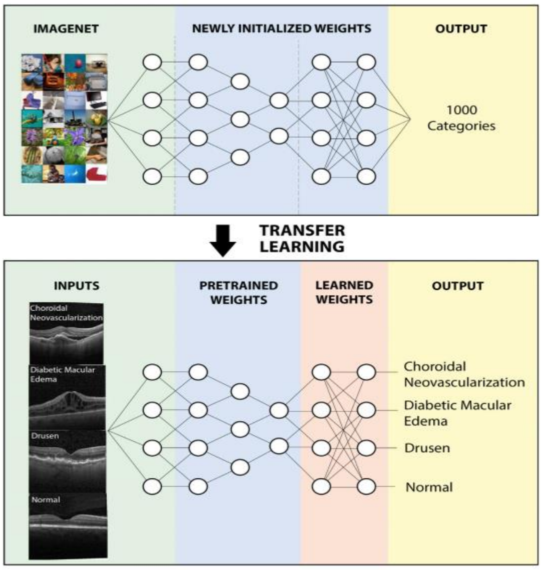


Figure 1.10 Visualization of transfer learning approaches which is applied to medical images by transferring initial weights from public ImageNet dataset [31]**.**

Deep learning models, such as ResNet, AlexNet, DenseNet, UNet, etc., are initially trained on large datasets like ImageNet, MSCOCO, Pascal, etc. Those datasets are publicly available and large datasets that contain natural images. At the end of the training, each layer of the network model gets a weight value. After that, those weight values are used initially on the model, and training starts with those initial weight values while training on a new dataset. A new dataset can be natural images, medical images, biological images, etc. From Figure 1.9, it can be seen that ImageNet has 1000 categories, but the new dataset has just 4 categories. It is not required to have the same number of categories for pre-training and post-training images. Besides, training a deep learning model on a large dataset like ImageNet needs powerful hardware. However, pre-trained models of networks are also publicly available and can be used directly. In deep learning, there are some issues to understand and overcome, including overfitting, training performance, unbalanced dataset problems, and hyper-parameter optimization.

The Overfitting problem occurs when the model learns details and noises about the training dataset but cannot learn general representation. Then, the model cannot show good performance on an unseen dataset. This problem is measured by checking differences in training loss and validation loss. From problem to problem, the importance of the amount of overfitting changes. For example, while for segmentation problems some percentage of overfitting can be tolerated, for classification problems that percentage cannot be tolerated and causes misclassification problems. To overcome overfitting, batch normalization, dropout, or L2 regularization methods can be used. Also, for small dataset sizes, simple models can be used, and for large datasets, larger model architectures can be selected.

Training performance should be monitored while training. Poor training performance means there are some problems with the data or training parameters. Image normalization can lead to poor performance since datasets may lose important features with normalization. Besides, code errors or the wrong augmentation technique can cause poor performance.

Unbalanced dataset problems occur when one class samples more than another class or classes. This problem leads to poor performance for the model. For instance, there are two classes, X and Y, and 80% of total data samples belong to the X class and 20% to the Y class. Then the model may learn to predict every sample as X, but it cannot learn Y samples. Classification accuracy can show high value, but it is misleading because the model is incorrect for class Y. There are some techniques to reduce class imbalance. Those techniques are upsampling and downsampling. While upsampling is a technique to increase the number of small size samples, downsampling is a technique to decrease the number of larger size samples to get a balanced dataset. Another technique to solve the class imbalance problem is to give the class weight in the loss function. The general equation for class weight calculation is given as;

$$\left(\frac{N_{Total\ examples}}{N_{Examples\ in\ class\ i}}\right) x \left(\frac{1}{N_{Classes}}\right) \tag{1.6}$$

Where *N* is the number of samples. This equation 1.6 calculates contribution of each class to the loss. This loss can be calculated through the entire dataset or just for each mini-batch that is processing.

Hyperparameter optimization is very important and critical for training performance. Hpyerparameters such as learning rate, mini bath size, max number of epochs, and L2 regularization strength dramatically affect model performance. Before training starts, those hyperparameters should be properly selected based on the problem and dataset. In addition, after giving an initial value to these parameters, their values can be updated during training. Choosing the initial value for the parameters is difficult. However, some models and packages come with an initial value for those parameters. It can be a way to start with those initial conditions and then update or change them. In addition, there are some optimization algorithms that can be applied to the dataset before training starts. Some algorithms use grid search, and some algorithms use random search. Random search algorithms generally try some combination of parameters and return the results of the best combinations for the given task. In the following section, there is an explanation of the use of deep learning in medical images.

1.2.3.1 Image Classification

Image classification is the task of predicting the class of the given image, such as whether the given image is a cancer cell or not. The number of classes depends on the given problem. Classification is the first success of deep learning. In Figure 1.11, a schematic of the classification task on the medical images is shown. Images are trained, and the model learns which image belongs to which class by using the labels of each training image. In Figure 1.11, all images are categorized into three classes.

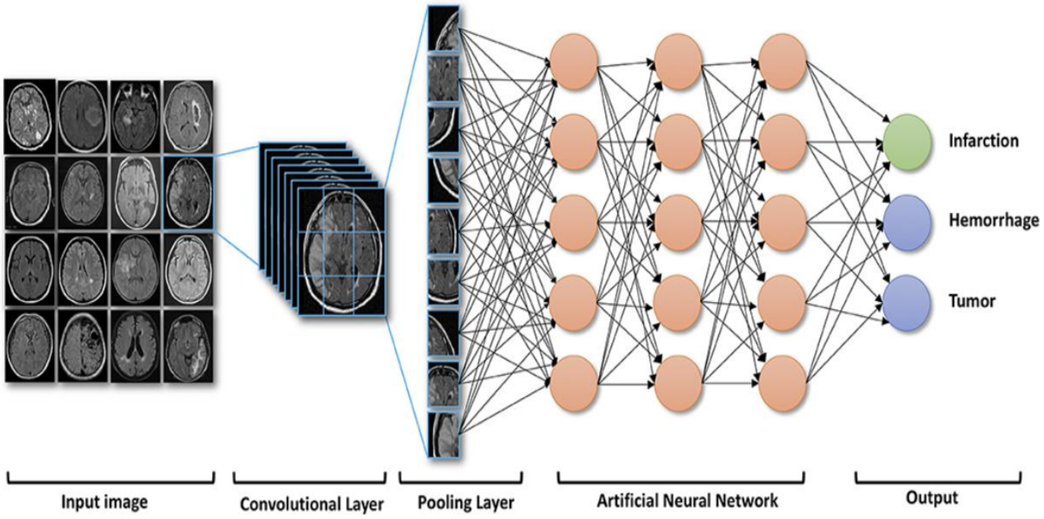


Figure 1.11 Visualization of image classification task with deep learning [32].

It is necessary to have annotated data while training. However, sometimes there may not be enough high-quality annotated datasets. In this case, the powerful approach is to use transfer learning. Both transfer learning from natural images and transfer learning from medical images can be applied, since natural images are similar to medical images. The transfer learning method uses a pre-trained model of a large dataset like ImageNet to start training, and then the final layer of the model is modified based on the new task.

1.2.3.2 Image Segmentation

Image segmentation is the task of dividing each part of an image based on its group of classes. Each object in the image is segmented according to its class. In Figure 1.12, a schematic of the segmentation task on the medical images is shown. Image segmentation tasks can be used for identifying single cells in the image, detecting cells or cell nuclei, detecting cancer cells, etc. In addition, segmentation has two types. One is called as semantic segmentation, and the other is called as instance segmentation. Even though both of them are segmentation, they are different, and they have different application areas. Semantic segmentation refers to the partitioning of an image based on object classes. If there are more than one of the same class of object, those objects are painted the same color. Some examples of semantic segmentation architectures are U-Net, DeepCell, DeepLabV3, and SegNet. Instance segmentation refers to partitioning an image based on the object classes and identifying each instance of an object by painting it with a different color. Some examples of instance segmentation architectures are Faster R-CNN, Retinanet, and Mask R-CNN.

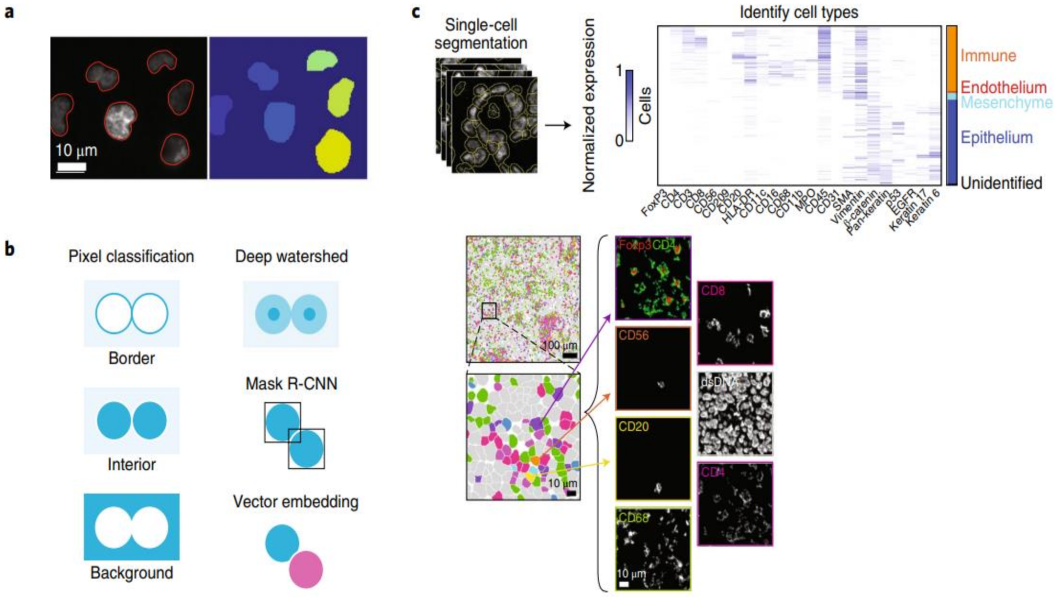


Figure 1.12 Visualization of image segmentation task [26]**. a shows instance segmentation which identifies each instance of an object. b shows deep watershed segmentation which identifies different objects with bounding boxes. c shows application of a deep learning segmentation for breast cancer by Keren et al** [33]**.**

1.2.3.3 Object Tracking

Object tracking is the task of automatically identifying objects in a video or in a series of time-lapse images. During the video or real time images, objects move. Object tracking is a complicated task, and it is difficult to adapt it to medical images because the number of objects in medical images can be hundreds or thousands. Also, objects in the medical images can be touched, merged, or disappear. The object tracking task consists of the detection of objects and the linkage of objects. DeepLabCut is one of the object tracking packages. In the following section, self-supervised learning techniques, which have become popular recently in deep learning, are explained.

1.2.3.4 Self-Supervised Learning

As long as data labeling in medical images takes a long time and requires an expert in the given field, it is really important to use self-supervised learning methods while training. Self-supervised learning is the task of training the model on unlabeled images and learning useful representations just from the data in a contrastive learning manner. After self-supervised pre-training, the pre-trained model is used in the downstream task by fine-tuning the representations. Downstream tasks can be image classification, semantic or instance segmentation, object detection, etc. SimCLR, MoCo, and MICLe are contrastive self-supervised pre-training techniques. The diagram of the SimCLR method is given in Figure 1.13. The method takes a raw image and creates two augmentations of the raw image based on some augmentation techniques. Then, it uses an encoder network such as ResNet-101 to encode two augmented images, and it learns generic representations by simultaneously maximizing agreement between the same image transformations and minimizing agreement between the different image transformations.

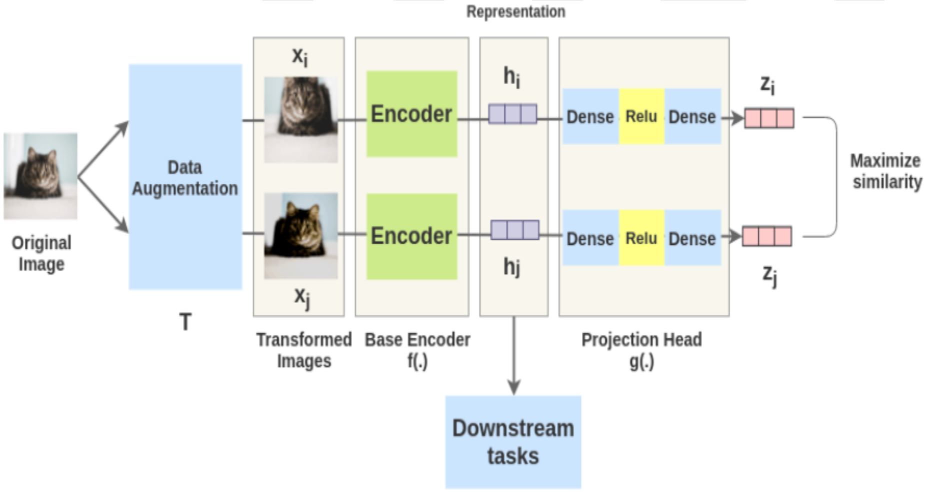


Figure 1.13 Schematic diagram of SimCLR framework [34].

It is stated in the literature that transfer learning from in-domain data gives better performance than transfer learning from out-domain data. In addition, there is no publicly available large medical image dataset or pre-trained models. That’s why self-supervised techniques can be really effective in increasing the performance of medical image tasks. Though the success of self-supervised learning is proven on natural images, it has a dark side for medical images, so it needs to be exposed.

In the following section, automated detection and quantification of leukemia cells and microbeads in optical microscope images are given in Chapter 2. The study mainly focuses on digital image processing and machine learning techniques. In chapter 3, automated detection and quantification of leukemia cells, beads, and micropads on images that are acquired from a microfluidic platform by an optical microscope are explained. The study focuses on digital image processing and deep learning techniques. In chapter 4, segmentation and quantification of cellular senescence from microscopy images are given. The study mainly focuses on deep learning and self-supervised learning techniques.

Chapter 2

Automated Quantification of Immunomagnetic Beads and Leukemia Cells from Optical Microscopy Images

2.1 Introduction

Cancer is the second leading cause of death in the United States and is expected to surpass heart diseases in the next five years [35]. Early detection of tumor cells and quantifying the tumor cells during therapy are crucial for a successful treatment. There are different treatments for cancer patients, such as surgery, drug therapy (chemotherapy), transplantation, radiation treatment, and immunotherapy. Among those methods, chemotherapy is a widely used first-line treatment for leukemia, but its results differ from patient to patient [35]. The effect of chemotherapy on the patient is shown in Figure 2.1. It can be seen that while some patients with chemotherapy achieve remission, some patients can have cancer cells that are resistant to treatment, and these cancer cells (blast cells) can cause a relapse of cancer. This case is known as minimal residual disease (MRD) [1]. To detect MRD, circulating tumor cells, or disseminated tumor cells [36], there are mainly two approaches: immunological assays and polymerase chain reaction (PCR)-based molecular assays.

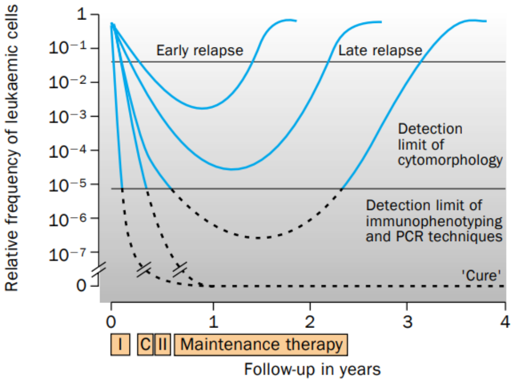


Figure 2.1 The graph of the leukemic cells frequency in peripheral blood or bone marrow of patients after and during the chemotherapy. The graph also shows the detection limit of flow cytometric and morphological techniques [37].

As an alternative to bench-top instruments, novel microfluidics and lab-on-a-chip systems have been demonstrated to capture, isolate, and count tumor cells. Some examples of the systems are microfluidic channels [38], immunomagnetic bead-based isolation, and magnetic fixation in a chip [10]. To get and analyze the samples in lab-on-a-chip, devices such as optical microscopes for characterizing and counting tumor cells, phase contrast, and immunofluorescence images have been used [10], [38], [39]. The immunomagnetic beads have some extrusive features. For example, they can be coated with receptors to capture target molecules, or external magnetic fields can be applied to move and immobilize immunomagnetic beads. These features make immunomagnetic beads favorable for lab-on-a-chip systems. Magnetic beads can be coated with antibodies. There are some examples in [10]. Anti-EpCAM, anti-EGFR and anti-CEA were used to capture breast cancer cells (MCF-7), non-small cell lung cancer (NSCLC) cells, and pancreatic cancer (PANC) cells, respectively, from blood samples. The schematic illustration of cell detection by immunomagnetic bead coating and magnetic field usage is shown in Figure 2.2. Magnetic beads are coated with CD_{10} antibody to capture and separate target cells. Then in the second sorting micron size magnetic beads are coated with another CD_{19} antibody. In each separation external magnetic field is used. The captured cells by immunomagnetic beads are tagged with CD_{45} antibody that is conjugated with gold pads. After magnetic separation, images are captured from the microfluidic system using an optical microscope to count and analyze the cells.

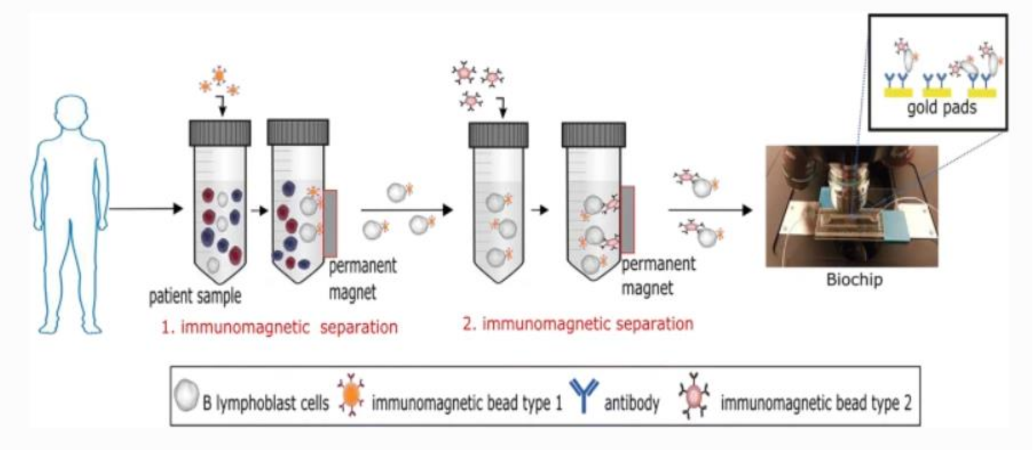

gold pads
patient sample
permanent magnet
1. immunomagnetic separation
permanent magnet
2. immunomagnetic separation
Biochip
B lymphoblast cells
immunomagnetic bead type 1
antibody
immunomagnetic bead type 2

Figure 2.2 Visualization of detection strategy with magnetic beads [40].

Besides, to separate target biological molecules, fluorescent or horseradish peroxidase (HPS) labels are used. However, those labels use secondary antibodies, which increases the complexity and cost of the process. Alternatives to optical methods, such as magnetic beads, have been shown. Even though magnetic bead systems are successful, they can only provide cell number data but not shape, size, or morphology features, which image analysis-based methods can provide.

Automated image analysis methods for digital pathology and microscopy images for cell detection and segmentation are reviewed in [41]. It can be concluded from the review that, to segment and detect cells or nuclei in the image, recent studies used digital image processing, machine learning, and deep learning techniques. They concluded that there is no way to determine which method is better than the other, and it is needed to evaluate methods based on the given problem and dataset. Bright-field microscopy image analysis is the easiest and has the lowest cost compared to fluorescence [42], phase contrast, and electron microscopy. White blood cell classification and leukocyte identification systems were developed by automated image processing techniques based on thresholding [43] and segmentation [44] algorithms. However, proposed studies were completed on the stained cell images since the cell's properties make it difficult to detect cells in bright-field microscopy images [45]. Machine learning-based methods have been developed to detect cells from bright-field images. In the study [46], background correction is done with machine learning. In another study, unstained living and non-living cells in bright-field images were detected by the support vector machine (SVM) method [47]. Scale-invariant feature transform (SIFT), which is a machine learning method, is applied to detect adherent and suspension cell lines in bright-field microscopy images [48].

Though bright-field microscopy images are analyzed to detect and count cells, the enumeration of both beads and leukemia cells has not been reported, and in this study, we introduce an automated detection and quantification of immunomagnetic beads and suspended leukemia cells in bright-field microscopy images. The images have 20x and 40x objectives, and beads are conjugated with anti-human CD_{19} antibodies that specifically bind to the B-type leukemia cells. We have utilized image processing and computer vision algorithms benefiting from the color, size, and distinct pattern features of immunomagnetic beads and cells to detect and quantify cells.

2.2 Experimental Section

2.2.1 Materials and Methods

The images contain immunomagnetic beads and cells in various shapes and sizes. Example images of my study are given in Figure 2.3. Cells in the images might appear as either isolated single cells covered by beads or smaller fragments. When a cell or cell fragment is not bound by any beads, its boundary and characteristic inner texture can be observed. However, when it is partially covered by one or more beads, the bound beads might obstruct some part of the cell boundary. Another case is the full coverage of cells by beads. In this case, a cluster of beads appears in the image. Hence, the cell-counting process includes all three of these cases.

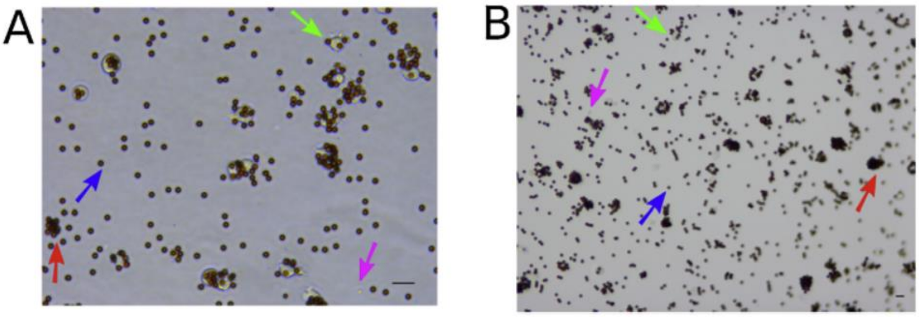


Figure 2.3 Typical optical micrographs of cells and beads recorded using A) 40× objective B) 20× objective. Scale bar indicates 20 μm. Blue arrow indicates single beads, pink arrow indicates the cell fragments, green arrow indicates single cell with attached immunomagnetic beads, red arrow indicates fully covered cells with beads as cluster [49].

In addition, beads can appear as a single bead or as a cluster of beads that are bound to a cell or cell fragments. Table 2.1 shows the two different objective characteristics of my dataset images. It can be seen that 40X images have a higher resolution than 20X images. This means that they have clearer and sharper objects. Therefore, 40X objective images have higher accuracy and require less effort to analyze compared to 20X objective images.

Table 2.1 Summary of the 20X and 40X objective characteristics.

	Objective 20X	Objective 40X
Numerical aperture	0.40	0.65
Working distance (mm)	1.2	0.56
Field of view (mm^2)	0.278	0.069
Pixel resolution (µm/pixel)	0.46	0.23

Nonetheless, even 40X objective images have advantages; the field of view of a 40X objective is 0.069, while the field of view of a 20X objective is 0.278. Thus, analyzing a target area on a 40X objective is slower than analyzing a target area on a 20X objective. Using real-time image processing as a signal readout method for a biochip requires scanning a sensor surface in an admissible amount of time. For these reasons, this study focuses more on the images acquired by 20X, which is time efficient but challenging.

In this study, beads are obtained from the same manufacturer in [50], and they reported a coefficient of variation of 0.024 for the beads, so bead size variation was inconsideration. The uniformity of the immunomagnetic bead size makes them easy to detect from the images. 20X objective images may have color distortions, the edges of the images might be out of focus, and the light intensity may vary from region to region. For those properties of images, only classical digital image processing techniques were not efficient alone, so machine-learning methods were incorporated into color-based methods to identify cells, immunomagnetic beads, and clusters of beads.

2.2.1.1 Detection and Quantification of Immunomagnetic Beads and Cells in 40X Objective Images

Since cells and beads on the 40X objective images are clear and have high resolution, only classical image processing techniques are applied to the images, and the workflow of the study is given in Figure 2.4. The same figure also shows the impact of color space conversion from RGB to YUV and thresholding on these color spaces for a sub-image that contains only cells. The images are acquired in raw RGB color space with 0.23 µm/pixel resolution. Since cells are more clearly visible in YUV color space, as shown in Figure 2.4, RGB images were converted to YUV color space.

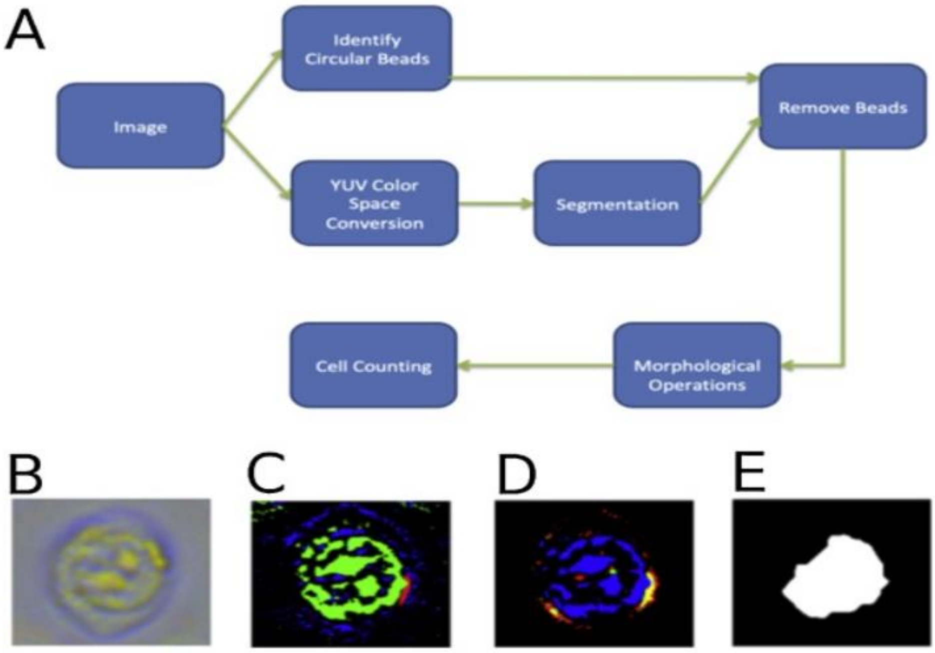


Figure 2.4 (A) Flow diagram of cell and bead detection and counting steps for the images recorded using 40× objective, (B) original RGB cell image, (C) YUV thresholded cell image, (D) RGB thresholded cell image, (E) YUV image after morphological operations [49]**.**

For color conversion, each image can be considered as three different layers and equation is given as:

$$\boldsymbol{I = I_{bg} + I_c + I_b} \tag{2.1}$$

where I_{bg}, I_c and I_b represents background, cells and beads respectively. Cells have a distinct U dominant color, corresponding to more blue color component. To distinguish cells using color feature, a three-channel multilevel thresholding is applied. The binary image of cells layer, B_c, is first estimated by applying a multilevel global threshold (same threshold for all images) as:

$$\boldsymbol{B_c} = \begin{cases} \boldsymbol{1\ if\ I_y(x,y)\ \in\ T_y\ and\ I_U(x,y)\ \in\ T_U\ and\ I_V(x,y)\ \in\ T_V} \\ \boldsymbol{0\ if\ otherwise} \end{cases} \tag{2.2}$$

Where T_y, T_U and T_V are threshold ranges for Y, U, and V channels, I_y, I_U, I_V are Y, U, V channels of the image. This thresholding operation detects most of the cells in the image, but since beads have a fringing effect around them, some bead parts also fall into this threshold range. Opportunely, beads have uniform circular shapes, and so to detect bead circular shapes, the hough transform method [51] is used. After finding circular beads, those beads were removed from the cell layer, followed by dilation and hole filling operations. Those operations provide for eliminating the fringing effects of beads in the new cell layer as follows:

$$\boldsymbol{\widehat{B}_c = B_c - (B_b \oplus s)} \tag{2.3}$$

where B_b is binary image that shows beads, $\oplus$ is dilation operation with s shape structuring element. The structuring element is the disc shape. After bead removal, the noise that represents the not-cell object in the binary image is removed by estimating its total number of pixels. Later, to combine those that are close to each other, a closing operation with a 20-pixel disc-shaped structuring element is applied. After all these morphological operations, on the binary image, the objects that represent the cells are counted. All of those steps provided consistent results because the global threshold value and morphological operation parameters were obtained with optimization.

The circular hough transform method can find most of the beads, but bead clusters and overlapping beads may not be detected by the method. When several beads overlap, they can lose their original circular shape and total number of pixel values. As a result, bead counting is achieved by finding the proportion of the pixel area of whole beads to that of a single one. A general view of the bead detection and counting process is given in Figure 2.5. Initially, raw RBG images are converted to YUV color space to get rid of noise. Then, one global threshold is applied to the image, and beads are segmented. After that, morphological operations are applied to remove small objects and noises. Finally, bead counting is achieved by counting the number of objects in a binary image.

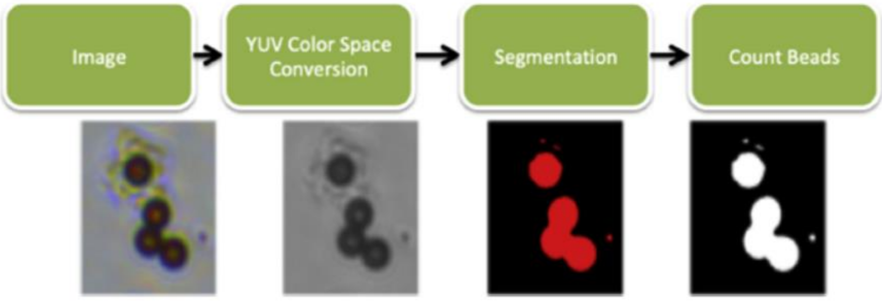


Figure 2.5 The flow diagram of the bead detection using Y channel in YUV color space from the images acquired by the 40× objective [49].

2.2.1.2 Detection and Quantification of Immunomagnetic Beads and Cells in 20X Objective Images

Detection and quantification of immunomagnetic beads in 20X objective images are achieved by the same method as in 40X objective images, which is shown in Figure 2.4. However, cells are not clearly visible in 20X objective images, so further image analysis methods are applied. In the image, cells may appear as isolated single cells or cells that are fully covered by immunomagnetic beads. Detection of those two cases needs different approaches because when a cell is covered by beads, its texture, color, or edges cannot be seen and used in the detection algorithm. Therefore, the detection of cells has two main branches: (1) the detection of isolated cells and (2) the detection of fully obstructed cells. The detection schema is shown in Figure 2.6.

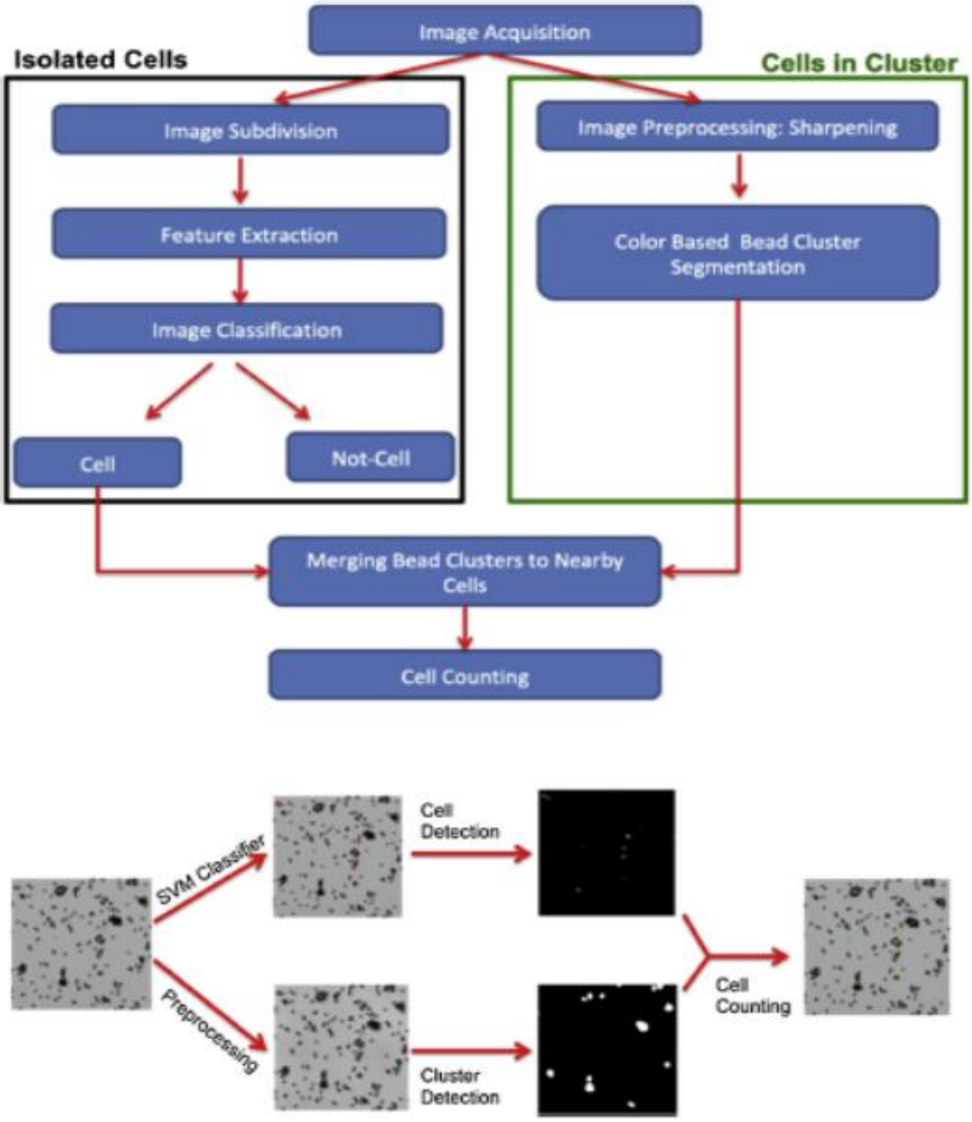


Figure 2.6 Flow diagram of cell detection and counting steps for the images recorded using 20× objective [49].

Before presenting the proposed algorithm, it should be mentioned the difficulties of cell detection in 20X objective images. When the zoom factor is halved, sensor noise, lens distortions, non-uniform illumination, softer edges, and sensor resolution drastically affect cell detection performance. As a result, only applying color- and size-based methods does not achieve desirable results in the 20X objective and needs further analysis. It can be seen from Figure 2.5 that the proposed method has two branches. The first branch uses machine learning-based methods to segment isolated cells in three stages. The first stage is preprocessing the images to get small images. During the pre-processing stage, raw, high-resolution images are divided into 12 x 12 pixel, non-overlapping sub-images. In the experiments, 8 high-resolution images are divided into 707,200 sub-images in total. The second stage is feature extraction for the machine learning model, and the third section is learning model construction to predict new test images. The second and third stages are carried out on the sub-images. All extracted features for the machine learning algorithm are given in Table 2.2. 264 features from the Gray Level Co-occurrence Matrix (GLCM) [52], 9 features from mean, variance, and skewness, 81 features from the Histogram of Oriented Gradient (HOG) [53], [54], 59 features from the Local Binary Pattern (LBP) [54] are extracted.

Table 2.2 All Extracted Features

Statistical Based	Texture Based		Others
Mean	Autocorrelation [55],	Maximum Probability [55]	HOG
Variance	Contrast [52], [55],	Sum of Squares [52]	LBP
Skewness	Correlation [52], [55]	Sum Average [52]	
	Correlation [56]	Sum Variance [52]	
	Cluster Prominence [55]	Sum Entropy [52]	
	Cluster Shade [55]	Difference Variance [52]	
	Dissimilarity [55]	Difference Entropy [52]	
	Energy [52], [55]	Information Measure of Correlation1 [52]	
	Entropy [55]	Information Measure of Correlation2 [52]	
	Homogeneity [55]	Inverse Difference Normalized [57]	
	Homogeneity [56]	Inverse Difference moment Normalized [57]	

Totally, GLCM selects 22 features, and we used distance = 1 and angle degrees (0, 45, 90, 135). So, features come in different degrees, and we add them up in one row: 22* 4 = 88. Also, our images were color images, and they have RGB values. GLCM is applied to gray-scale images. For that reason, we took R, G, and B separately, and for each, we selected 88 features (which come from four degrees). Then, for one image, we add all R, G, and B features in one row, which makes 88* 3 = 264 features.

The HOG features are extracted from local regions with 12x12 pixels. Then these pixels are divided into nine rectangular cells, and nine bin histograms per cell are used. The nine histograms with nine bins are concatenated to make an 81-dimensional feature vector. The general view of HOG feature extraction is given in Figure 2.7.

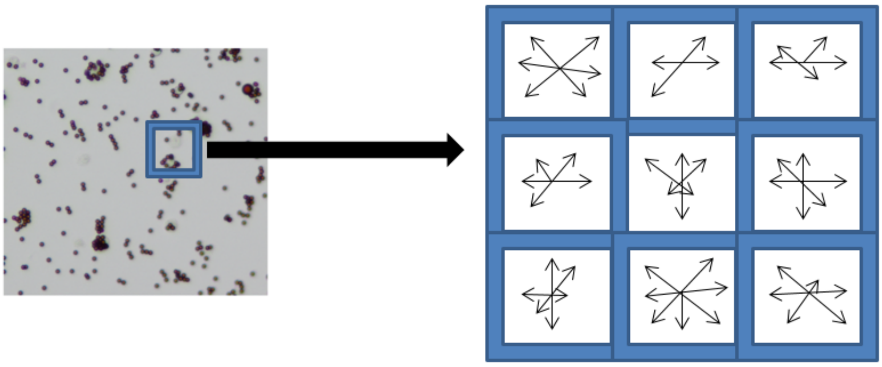


Figure 2.7 View of the HOG features histograms in our image [49].

LBP feature length depends on the number of cells and the number of bins where,

LBP feature length = numberOfCells x numberOfBins

In Matlab by default;

Cellsize equals size(I);

where I is the image

numCells = prod(floor(size(I)/CellSize));

⇒ numCells = 1

Number of neighbors = P = 8

B = (P × P–1) + 3

⇒ B = 59, so LBP features = 1x59

In the learning model construction stage, the Support Vector Machine (SVM) method is used for two class classification. Extracted 413 features were directly used while training with the SVM method. Any feature reduction method was not used. In a study [47], it was proven that SVM-based methods increased the detection of viable and non-viable cells in unstained images. Training and test datasets are labeled by an expert as positive for cells and negative for non-cells. However, the number of negative samples in the dataset is much higher than the number of positive samples. That makes the dataset unbalanced, and an unbalanced dataset problem leads to poor network performance. That's why the dataset is balanced before training by using the down-sampling method by decreasing the number of negative samples [58]. While training the dataset, the leave-one-out method, in which all test set sub-images are left out, is applied to be sure that the network not only learns specific image features and learns general image features. There are N images in the dataset, and N-1 images are used as training, and one remaining image is used to estimate the test performance of the network. This training of N-1 images and testing on one image operation is applied iteratively and N times. Then, average performance is estimated. Totally, 100,000 sub-images are classified by the network system. It is also important to choose the optimum parameters for the SVM network, and in this study, the libSVM toolbox with a radial basis function kernel and multi-class classification type is employed as a classifier. The training dataset is trained by this classifier. Kernel function is set to radial basis. The hyper-parameters (C, γ) of the LibSVM are determined with a five-fold cross-validation on the grid space, and the equation is given as;

$$(C,\gamma)\ \epsilon\ \{\ (\ 2^{m},2^{n})\ |\ m=-2,-1,\ldots\ldots,2, n=-2,-1,\ldots\ldots,3\} \tag{2.4}$$

After cell detection by a machine learning algorithm, bead clusters that are counted as cells are detected using color-based methods and morphological operations. First of all, broken connections in bead clusters are fixed by a morphological closing operation with a disc-shaped structuring element that has a 10-pixel size. In general, bead clusters are larger than 400 pixels.

Therefore, objects that are smaller than 400 pixels are removed, and then counting objects on binary images is applied, and the cluster detection part is completed. After completing isolated cell and cluster detection, which is shown in Figure 2.5, bead clusters and nearby cells are merged to count the total number of cells in each raw image. During the merging operation, a Gaussian filter with $\sigma = 2$ is used. Then, following a binarization of the image, a morphological closing operation with disc SE = 20 is performed. The algorithm is completed by counting the resulting objects. This gives the number of cells in the image.

2.3 Results

The images in the dataset were acquired by different users at different times. This case causes illumination light variations and focusing problems at the image edges. In the dataset, sixteen images are acquired with a 40X objective, and nine images are acquired with a 20X objective. For each image set, bead and cell detection and counting operations are applied by the proposed methods. After that, the performance of the proposed methods is completed with the manual counting of an expert. The used evaluation metrics are given below:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{2.5}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{2.6}$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} \tag{2.7}$$

$$\boldsymbol{F-Measure} = \boldsymbol{2x}\frac{\boldsymbol{Precision\ x\ Recall}}{\boldsymbol{Precision+Recall}} \tag{2.8}$$

$$\boldsymbol{Detection\ Error} = \frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{x}[(\boldsymbol{1-Precision}) + (\boldsymbol{1-Recall})] \tag{2.9}$$

$$\boldsymbol{False\ Negative\ Rate} = \frac{\boldsymbol{FN}}{\boldsymbol{FN+TP}} \tag{2.10}$$

In the following subsections, the results of the algorithms explained previously are reported for immunomagnetic bead and cell quantification. Fully covered, isolated, and partially covered cell detection results and the combination of those three approaches to obtain an overall quantification of the cells are explained.

2.3.1 Bead Quantification

The performance of bead quantification algorithm is estimated by the error rate , E which is defined as:

$$\boldsymbol{E} = \left|\frac{\boldsymbol{\tilde{N}-N}}{\boldsymbol{N}}\right| \tag{2.11}$$

Where N is the actual number of beads in the image, and $\widetilde{N}$ is the number of found beads by the proposed method that is shown in Figure 2.5. The error rate value is used as an evaluation metric for the bead detection and quantification methods for both 20X and 40X images, and results are given in Table 2.3. It can be concluded that error rate for both objectives is low because beads are opaque and have a darker texture than cells and background to detect. Nonetheless, due to lower resolution and noise in 20X images, the error rate of those images is higher than that of 40X images.

Table 2.3 Average bead counting error rates for images acquired by 40× and 20× objectives

	40× Images	**20× Images**
E	0.0171	0.0384

Error rates for bead quantification show satisfactory results even when only size- and color-based methods are used. However, since cells have a semi-transparent texture, it is more difficult to detect and quantify them. Besides, the larger accuracy of cell quantification is crucial for the users than bead quantification accuracy. Those effects and challenges make cell quantification more difficult.

2.3.2 Cell Quantification

For evaluating cell quantification algorithms performance, true positive (TP), false positive (FP), and false negative (FN) values are determined to calculate precision, recall, specificity, F-measure, false negative rate, and detection error metrics. Some samples of the original 40X images and their cell detection results are shown in Figure 2.8. Detection and quantification accuracy for 40X images are higher, and detection and quantification methods are easier compared to 20X objective images.

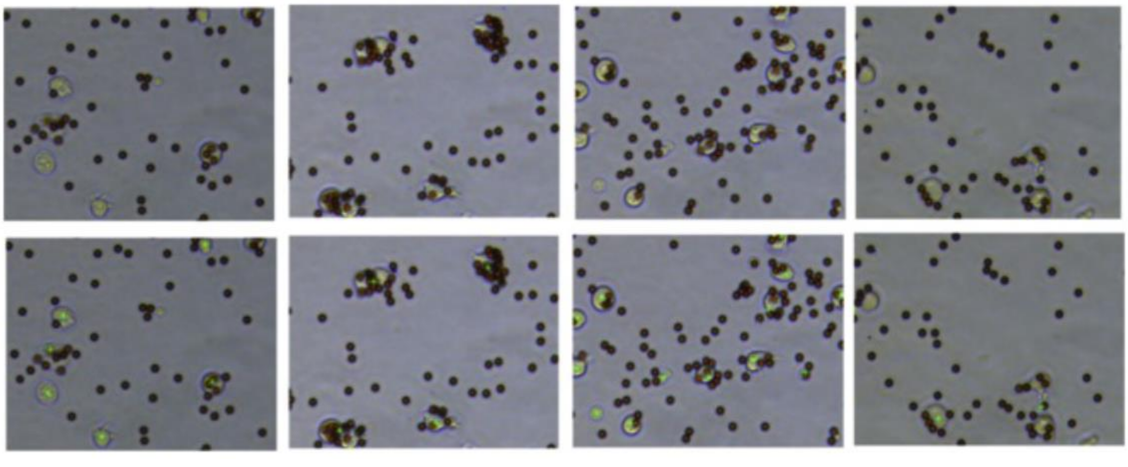


Figure 2.8 Original images (top row) and green detection marks for each cell (bottom row)

for the 40× objective [49].

In order to see the performance of the proposed algorithms visually, the images that have the lowest precision and the highest precision are compared. It can be seen from the right column of Figure 2.9 that the highest precision image is clear, uniform, and light variations are minimal. The left column of Figure 2.9 shows the lowest precision image in our dataset. This image has blurry edges, causing false positives and thus decreasing the precision rate. Also, to see the effect of image quality on precision rate, the blurry part of the lowest precision image is cropped, and precision is calculated again. Results for the new cropped image are given in Figure 2.10. After removing the blurry edges, the precision metric was increased from 0.54 to 0.85, while the detection error was decreased from 0.34 to 0.23. Results are shown in Table 2.4. It is clear that clearing images and removing problematic regions before detection and quantification increase the performance of the algorithm, but for practical reasons, the proposed methods are applied to whole images, and some image processing techniques are applied to reduce noise and problems.

Cells may appear in three forms: (1) isolated, (2) partially covered by beads, and (3) fully covered by beads. Those three forms have different textures, colors, and appearances, and these different properties require different image processing techniques to detect and analyze. First, all three categories are analyzed separately, and then the results are combined to get the overall performance of the proposed methodology for cell detection. In the following sections, the results that are obtained for each of the three categories are given for 20X objective images.

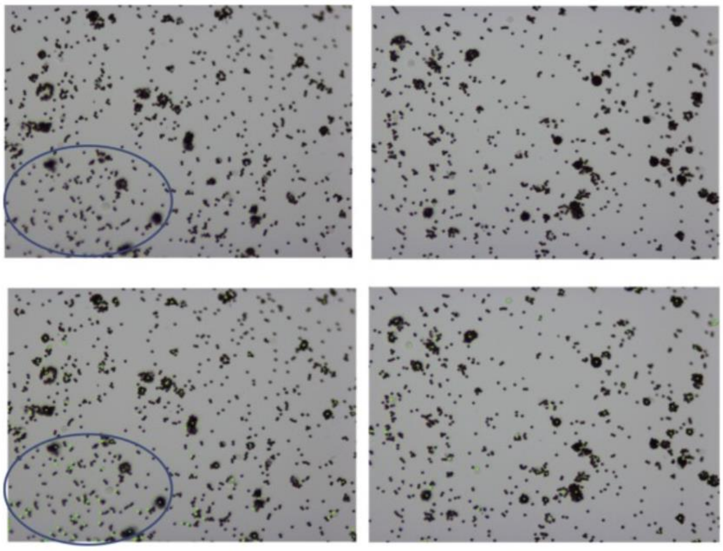


Figure 2.9 Comparison of high precision and low precision images. Blue circle indicates the blurry part of the image (top left) and resulted false positives (bottom left) [49]**.**

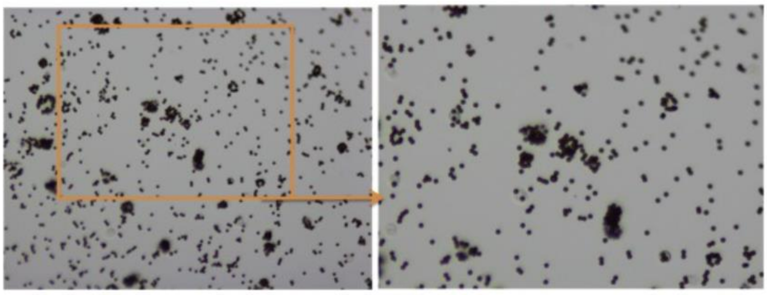


Figure 2.10 The impact of image quality on the performance of the image processing algorithm. Original image (image 9) (left), cropped image without blurry part (right) [49].

Table 2.4 The impact of cropping out the blurry region of the image on the performance of the algorithm.

	Precision	Recall	Specificity	False Negative Rate	F-Measure	Detection Error
Original Image	0.54	0.76	0.99	0.23	0.63	0.34
Excluding the Blurry region	0.85	0.68	0.99	0.32	0.75	0.23

One common form of cell appearance is clusters, which are covered by beads. If a cell is covered by a bead, the cell is not seen, and the bead appearance also changes compared to the single original bead. The actual texture of the cell is usually concealed, but bead clusters are indicators of obscured cells. As a result, those objects in the image are counted as cells. The result of cluster detection is given in Table 2.4. The table shows the result of 20X objective images. On average, 45.77 bead clusters out of an average of 58.3 were correctly detected, and 12.5 were incorrectly detected. The recall value is higher than 94%, which indicates that most of the bead clusters are detected correctly and there are a low number of missed detections. On the other hand, an 80% precision value indicates that some objects, such as the one in the out-of focus region, are detected as bead clusters, which causes a lower precision rate. For example, image 3 from Table 2.5 has blurry and out-of-focus corners, and this causes the highest false-positive measure. Unlike image 3, image 4 has clearer and more in-focus corners, and this causes the lowest false positive rate.

Table 2.5 Fully Covered Cell (Bead Cluster) Detection Results.

	True Positive	False Positive	Missed Detection	Precision	Recall	F-Measure
Image1	55	22	0	0.71	1	0.83
Image2	55	16	2	0.77	0.96	0.85
Image3	55	24	2	0.69	0.96	0.80
Image4	32	2	2	0.94	0.94	0.94
Image5	52	13	1	0.80	0.98	0.88
Image6	54	13	2	0.81	0.96	0.87
Image7	36	8	2	0.81	0.94	0.87
Image8	40	11	2	0.78	0.95	0.86
Image9	33	4	3	0.89	0.91	0.90
Average	45.77	12.55	1.77	0.80	0.94	0.87

Another challenge for this part is to decide on the bead cluster size and how many beads indicate an underlying cell. In this experiment, a minimum bead cluster size is incorporated into the method based on an expert’s view.

Another type of cell is isolated and partially covered. Those cell types have visible cell textures. For this reason, the method described in Figure 2.6 as an SVM branch extracts cell features, and those features are trained by SVM to predict unseen image’s cell parts. Also, to test another machine learning method with SVM, neural networks and random forests are used and compared in Weka data mining software [59], [60]. The raw images are divided into 88,000 overlapping sub-images with a size of 12 x 12 pixels, and 413 features, which are described in the method section, are extracted for each sub-image. The dataset was unbalanced, and balanced datasets are obtained by down-sampling negative examples. For each image in the dataset, test data was generated to compare the results of each method. To obtain reliable test results, a leave-one(image)-out approach is implemented [61]. The approach leaves out one test image completely from the training dataset. During train-test cycles, all images are used as tests with the leave-one-out method. To give the overall performance of the proposed methods, the result of each test image in train-test cycles is averaged. The results of different machine learning methods are presented in Table 2.6.

Table 2.6 Comparison of machine learning algorithms for each image for isolated and partially covered cell detection (in Weka)

SVM / Random Forest / Neural Networks									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Accuracy
Image1									
SVM	0.865	0.135	0.868	0.865	0.865	0.733	0.865	0.817	86.54
RF	0.888	0.112	0.892	0.888	0.888	0.781	0.960	0.959	88.83
NN	0.855	0.145	0.861	0.855	0.855	0.716	0.912	0.907	85.53
Image2									
SVM	0.898	0.102	0.899	0.898	0.898	0.797	0.898	0.858	89.83
RF	0.923	0.077	0.924	0.923	0.923	0.847	0.963	0.958	92.30
NN	0.907	0.093	0.907	0.907	0.907	0.813	0.949	0.940	90.65
Image3									
SVM	0.872	0.128	0.873	0.872	0.872	0.745	0.872	0.824	87.17
RF	0.910	0.090	0.924	0.910	0.910	0.834	0.913	0.906	91.02
NN	0.808	0.192	0.810	0.808	0.807	0.617	0.843	0.810	80.76
Image4									
SVM	0.690	0.310	0.702	0.690	0.686	0.392	0.690	0.634	69.04
RF	0.802	0.198	0.805	0.802	0.801	0.607	0.879	0.871	80.15
NN	0.643	0.357	0.732	0.643	0.605	0.364	0.727	0.738	64.28
Image5									
SVM	0.894	0.106	0.899	0.894	0.893	0.793	0.894	0.854	89.35
RF	0.935	0.065	0.939	0.935	0.935	0.874	0.977	0.975	93.51
NN	0.866	0.134	0.886	0.866	0.864	0.752	0.938	0.933	86.57
Image6									
SVM	0.860	0.140	0.870	0.860	0.859	0.730	0.860	0.814	86.03
RF	0.896	0.104	0.904	0.896	0.896	0.800	0.962	0.960	89.63
NN	0.676	0.324	0.716	0.676	0.660	0.390	0.860	0.860	67.56
Image7									
SVM	0.846	0.154	0.876	0.846	0.843	0.721	0.846	0.803	84.59
RF	0.912	0.088	0.922	0.912	0.911	0.833	0.986	0.985	91.19
NN	0.789	0.211	0.836	0.789	0.782	0.623	0.931	0.934	78.93
Image8									
SVM	0.677	0.323	0.682	0.677	0.675	0.359	0.677	0.621	67.70
RF	0.726	0.274	0.729	0.726	0.725	0.455	0.783	0.758	72.56
NN	0.587	0.413	0.587	0.587	0.587	0.174	0.630	0.621	58.68

The overall performance of different machine learning algorithms on our dataset is given in Table 2.7. Random forest has the highest accuracy with 87.4%, and neural networks have the lowest accuracy with 76.6%, but neural network settings can be optimized for better accuracy. In this experiment, the neural network has 413 nodes, 208 hidden neurons, and 1 hidden layer. The data for all methods is normalized before training.

Table 2.7 Average accuracy of machine learning methods in Weka for isolated and partially covered cell detection.

Cell Detection			
	SVM (Support Vector Machine)	Neural Network	Random Forest
Average Accuracy	82.5372	76.6250	87.4058
Average TP Rate	0.825	0.766	0.874
Average FP Rate	0.174	0.233	0.126
Average Precision	0.833	0.789	0.879
Average Recall	0.825	0.766	0.874
Average F-Measure	0.823	0.758	0.873
Average MCC	0.658	0.556	0.753
Average ROC Area	0.825	0.848	0.927
Average PRC Area	0.778	0.842	0.921

In this part, the results of all three cell categories are combined to give an overall cell detection and quantification performance. The performance of the proposed algorithms for each image in the dataset is presented in Table 2.8 and a summary for 20X objective images is given in Table 2.9. When finding the location of the cell, the algorithm can point to different positions in one cell. Each location is considered as correct as long as it points to a point in the cell. However, when counting those points, multiple marks in one cell are counted as a false positive because in one cell there can be more than one point, but more than one point doesn’t mean more than one cell. And this case is calculated as multiple marked and shown in Table 2.9. In the table, multiple marked columns show how many times a cell is falsely detected more than once. From Table 2.9, it can be concluded that the majority of the images are correctly counted, with fewer missed detections and multiple detection errors.

Table 2.8 Cell counting results for each image

	Isolated Cell			Fully Covered Cell			Partially Covered Cell			False Alarm
	Correctly Marked	Multiple Marked	Missed Detection	Correctly Marked	Multiple Marked	Missed Detection	Correctly Marked	Multiple Marked	Missed Detection	
Image1	15	1	4	27	1	1	23	1	0	37
Image2	16	1	2	45	0	2	10	0	8	18
Image3	8	0	5	50	0	3	12	1	1	21
Image4	8	1	2	36	2	3	9	2	1	15
Image5	12	0	1	44	0	3	17	2	4	25
Image6	9	1	1	53	2	1	15	1	3	17
Image7	8	1	2	31	0	8	14	0	4	4
Image8	3	1	6	41	2	4	4	1	1	16
Image9	6	1	9	28	3	1	13	1	5	38

Table 2.9 Summary Table for 20X images. Correctly Classified Accuracy =100*Correctly Classified / (Correctly Classified + Missed Detection)

	Correctly Classified	Multiple Marked	Missed Detection	Correctly Classified Accuracy %
Image 1	65	3	5	92.8
Image 2	71	1	12	85.5
Image 3	70	1	9	88.6
Image 4	53	5	6	89.8
Image 5	73	2	8	90.1
Image 6	77	4	5	93.9
Image 7	53	1	14	79.1
Image 8	48	4	11	81.3
Image 9	47	5	15	75.8
Average	61	3	9	86.32

2.4 Discussion and Conclusion

Cells can be imaged by microscopy, which has the forms of bright-field, phase contrast, fluorescence, and electron, and those images can be analyzed by image processing techniques. On the one hand, electron microscopy images result in high resolution and variable images [62]. Fluorescent microscopy requires labeling cells, and after capturing images, cell counting algorithms are developed [63], [64]. On the other hand, bright-field microscopy is the easiest and lowest cost method compared to other methods since it doesn't need labeling operations or optics. Bright-field cell images are transparent, and this causes challenges for detection. In some studies [65], [66], cells are stained for better visualization and differentiation from other molecules in the image. However, in this study, cell detection and quantification are reported on unstained cell images. Also, in this study, the detection of immunomagnetic beads, which are used for cell separation and better cell detection. As shown in Table 2.4, blurry edges and illumination changes affect the performance of the algorithm. This impact was also reported in a similar study that detected cells using machine learning with 82% accuracy [67]. Stained cell detection studies [68]–[70] show higher accuracy and precision compared to the proposed mythology since staining makes cells distinctive. In another study [71], SVM-based cell detection accuracy on fluorescent labeled cells was reported as 90%, whereas our method results in 86.32% accuracy with blurry images. Also, this study images not only include cells but also include beads and cells that are fully or partially covered by beads. In this study, the effect of image quality on the algorithm is shown, so to maximize the quality of images, automated illumination and focusing adjustments can be added to the MRD biochip. Our dataset's image size is 12 megapixels. This high image size causes an unbalanced dataset. As a further issue, decreasing the image size can solve the unbalanced dataset issue. In addition, to increase the performance of the proposed method, a top-hat transformation to overcome the illumination effect can be used before the segmentation [72]. Also, to reduce the noise, adaptive local noise reduction method can be used [73]. Moreover, in the feature extraction part, more features can be added or some features can be removed by principle component analysis (PCA) to increase performance. The advantages of feature extraction by increasing accuracy, decreasing cost, or decreasing data storage are stated in [74]–[77]. Besides, a comparison of different machine learning algorithms shows that selecting the right machine learning algorithm for the given dataset can increase accuracy. It is also crucial to optimize machine learning parameters.

As a conclusion, the proposed study is aimed to detect and quantify cells in a biochip. The biochip includes immunomagnetic beads for the separation of the target B lymphoblast cells.

During the study, 20X and 40X objective images are used. The maximum precision rate for a 40X objective image was 90% without cell staining and with color-based methods. In addition, cell detection in 20X images presented more challenges, and only color-based methods were not enough, so a machine learning based method was proposed to detect cells in 20X objective images. Overall results show that image quality has a great impact on the performance of the algorithm, and the highest precision is 94% for clear image.

Chapter 3

Image-Analysis Based and Deep Learning Based Readout Method for Biochip: Automated Detection and Quantification of Immunomagnetic Beads, Micropads and Leukemia Cells

3.1 Introduction

Cancer therapy requires a number of cancer cells to monitor the effect of the treatment. One of the most common treatment methods is chemotherapy. However, chemotherapy can show different effects on different patients, like refractory, relapse, or complete remission [35]. Optimized based on a personalized dose of chemotherapy can decrease the risk of harmful toxicity and increase the positive effect [78]. Minimal Residual Disease (MRD), as stated in Chapter 2, is a relapse of cancer cells that are resistant to chemotherapy [3]. To estimate MRD, flow cytometry (FC) and polymerase chain reaction (PCR), next-generation sequencing methods which are molecular or immunological assays are widely used [79]–[81]. Microfluidic platforms appear as an alternative to those methods [82]. In this study, the images are acquired from a biochip (biosensor), which is also known as a microfluidic chip. The general view of a biosensor device is shown in Figure 3.1. A biochip (biosensor) device has three main components: (1) a biomolecular probe, (2) a transducer, and (3) a signal processor. First of all, by using biomolecular probes, target molecules are captured, and biological detection of target molecules is accomplished. Some examples of biomolecular probes are antibodies, DNA, aptamers, and extracellular matrix (ECM) proteins. Then, the transducer part converts biomolecular recognition into measurable signals. Those signals can be optical, electrochemical, magnetic, mechanical, thermal, or a combination of those. Finally, the signal processing unit displays the result, or signal. After displaying the results, microscopes and attached cameras are used to record images from the chips, and then some algorithms can be used to analyze the data [83]. To analyze image data, image processing techniques can be applied.

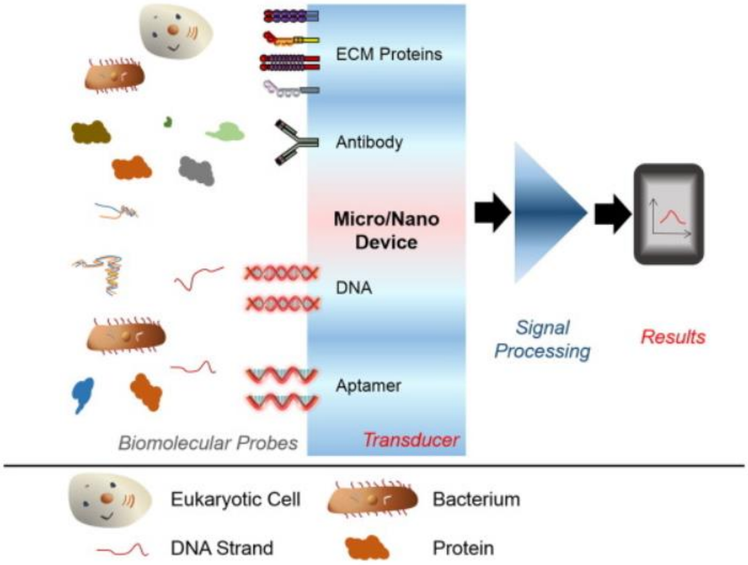


Figure 3.1 Schematic representation of a biosensor (biochip) and essential components [84].

The microfluidic platform (biochip) of this study includes immunomagnetic beads and micropads. Both of them are functionalized with different types of antibodies, which enables screening target molecules for multiple antigens in the cell membrane. In affinity-based biosensors, which use antibodies in sandwich assay are used. One antibody is used on the surface as a capture antibody, and the second antibody is used as a detection antibody. In this study, antibody-functionalized micropads were used to capture target cells on the microfluidic platform, and thus target cells were quantified. Bright-field microscopy is used to capture images on the microfluidic platform. Bright-field microscopy is easy and has the lowest cost; that's why, in this study, we used bright-field microscopy. Since the optical properties and appearance of the cells in the bright-field microscopy images make it challenging to detect and identify them. There are some open-access software tools, such as Fiji [85], and Icy [86], to detect, segment, and analyze the cell images. In addition, machine learning algorithms are used to detect cells automatically [47], [48]. Besides, there is open access image processing software called PIACG. This software is used to estimate the total area of the viable cells in fetal bovine serum from the microscopy images that are acquired from microfluidic chips. The software is Python-based and can segment a large number of unstained cells. The software is specialized to recognize other objects that are not cells. The main methods in the software are thresholding and filtering operations. Also, software does not need any pre-processing to segment cells. However, the software is proposed to segment and analyze the specific cell type, so it is not capable of detecting and counting different cell types and objects in the image [87].

In the previous chapter, leukemia cells and immunomagnetic beads were detected and quantified based on machine learning-based and classical image processing based methods. The images had 20X and 40X objectives, and their results were compared. In this chapter, automated detection and quantification methods will be explained as a readout mechanism for the microfluidic platform. The main purpose of the microfluidic platform is to monitor the patient's response to the chemotherapy treatment. The main purpose of the developed algorithm is to count patient leukemia cells, immunomagnetic beads, and micropads in bright-field microscopy images with a 20X objective. In this study, the images were acquired from real patients. In the first part of this chapter, image processing techniques will be explained, then in the second part, deep learning-based techniques will be explained, and finally, results will be compared. In the first part of our study, we used image processing-based techniques that benefit from the color, size, and shape identification of objects in the image to reduce the processing time. In the second part of our study, we used deep learning-based techniques to improve the results of the first study and have a more generalized system.

In the first part of this study, separate algorithms are used to detect and quantify cells, beads, and micropads since they have different shapes, colors, and sizes. Also, it is needed to optimize the parameters separately. For example, morphological operations need a structuring element, and structuring elements need shape and size parameters. Also, since cells, beads, and micropads are different from each other, it is necessary to optimize and define different sizes and shapes for structuring elements. In addition, not all images in the dataset are clear. Some images are blurry and have an illumination effect. So they need pre-processing before implementing the proposed algorithm. To overcome all these limitations and get better performance, in the second part of this study, deep learning-based algorithms are applied to the dataset.

Deep learning techniques are broadly applied to biomedical images for segmentation, classification, and object detection. In [88], accuracy in nucleus segmentation was achieved by DeepCell and U-Net architectures. In another study, classification with deep learning was used to classify yeast cells in microscopy images [89]. Blood cell images, which are obtained from a scanning electron microscope (SEM), were segmented into red blood cells (RBCs), white blood cells (WBCs), and platelets. To segment objects into the given classes, a novel convolutional encoder-decoder framework was proposed. VGG-16 was used as a baseline network to extract pixel-wise features [90]. To classify single cell images into four classes, a convolutional neural network (CNN) was used. The accuracy of the CNN was 99%. Later, CNN performance was compared with machine learning methods such as k-nearest neighbor (k-NN) and support vector machine (SVM) [91]. In another study, the transfer learning method was used, and the initial weights of AlexNet, ResNet, VGG-16, and DenseNet were used for malaria classification [92]. In addition, the U-Net architecture was adapted to ImageJ, which is an image processing tool, in order to enable non-machine-learning experts to analyze their own data [93]. In review, it is stated that deep learning methods solve the diagnosis and monitoring of cancer problems [94]. Stained leukemia cell detection from microscopy images was achieved by transfer learning with AlexNet, and classification accuracy was given as 100% [95]. In another study [96], classification of acute lymphoblastic leukemia (ALL) into B-cell type and T-cell type was achieved by pre-trained convolutional neural networks, AlexNet and LeukNet. The reported classification accuracy was 94.12%.

In the second part of this study, as a readout mechanism for the microfluidic platform, an automated semantic segmentation algorithm was proposed to segment leukemia cells, immunomagnetic beads, and micropads. During the study, parameter optimization and the solution of the unbalanced dataset problem were completed. After the segmentation, automated quantification of objects was studied.

3.2 Experimental Section

3.2.1 Materials and Methods

In this study, two types of immunomagnetic beads were used. Micron-size beads were coated with CD19 antibody, and nano-size beads were coated with CD10 antibody to separate the target cells. Micropads in the biochip were coated with gold and functionalized with CD45 antibodies. After the separation operation, target cells were introduced to the biochip, and cells were captured with microscopy for automated segmentation and quantification. By using coated beads and coated micropads, it was shown that the MRD biochip was able to monitor cells that have three different markers.

Images were captured by bright-field microscopy from the MRD biochip. Image sizes were 3116 x 4076 with a 20X objective. A general view of the microfluidic system is demonstrated in Figure 3.2. Cells in the images were unstained, and fluorescent dyes were not used.

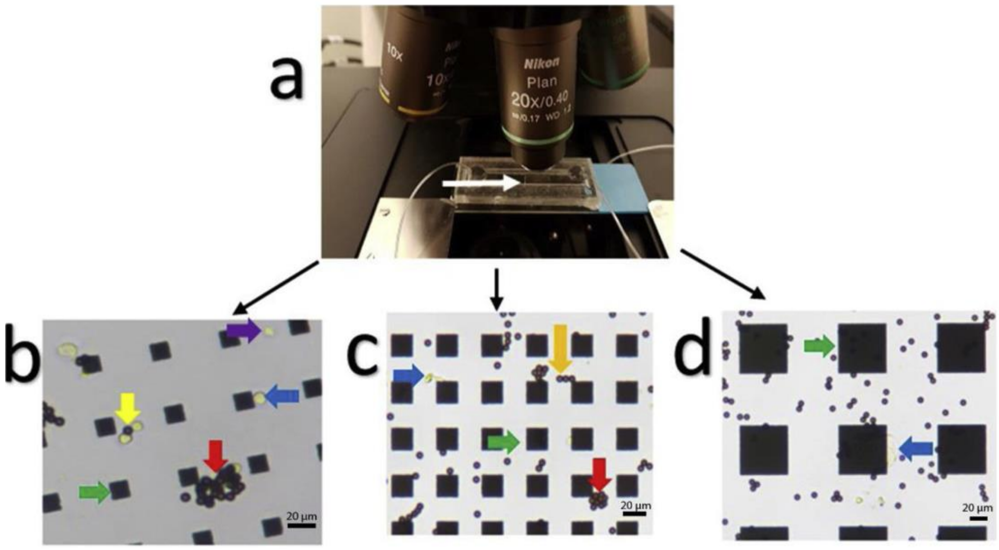


Figure 3.2 a) Optical microscopy system with 20X objective and the microfluidic platform b) 20X objective biochip microscopy image which has small micropad. c) 20X objective biochip microscopy image which has bigger micropad areas than a. d) 20X objective biochip microscopy image which has larger micropad area than b and c. Blue arrow indicates cells attached to a micropad, purple arrow indicates a single cell, yellow arrow indicates two single cells with attached bead, red arrow indicates fully covered cells with beads as cluster, and green arrow indicates square pads. In addition, white arrow indicates the part of the biochip for capturing the image [97].

3.2.1.1 Micropad Detection and Quantification

In the images, micropads have regular shapes and sizes with grid-order places. Micropad sizes may change based on the production of the chip. The original micropads in the images and the general workflow of the study to detect and quantify them are given in Figure 3.3. Micropad sizes that are used in this study are in the 15 µm x 15 µm-35 µm x 35 µm range. Compared to bead sizes, micropads are larger, and compared to cells, micropads are opaquer. To get only micropad images in binary form, a closing operation that has a larger structuring element size than a single bead in the image is applied. With this operation, a gray-scale image that has only micropads is obtained. To determine the size of the structuring element, the single micropad edge length is calculated. This length is calculated manually. After a grayscale image is obtained, Otsu's threshold method is applied to convert it to a binary image. At the end, the binary image, which has only micropads accessed. On the image, the number of micropads is achieved by connected components operation.

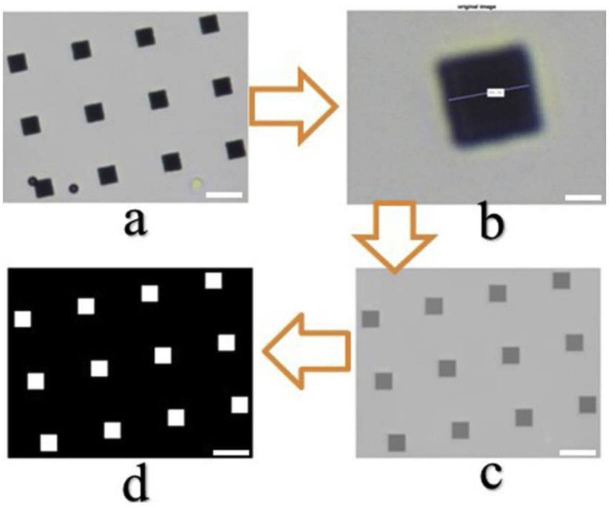


Figure 3.3 a) Original RGB image, b) a micropad from the original image, c) result of micropad extraction algorithm, image is in gray-scale, and d) binary image obtained by using Otsu's threshold on gray-scale image in c. Scale bars = 20 µm [97]**.**

3.2.1.2 Immunomagnetic Bead Detection and Quantification

A schematic diagram of bead detection and quantification steps is given in 3.4. In the images, the beads have regular and circular shapes. Sometimes, beads can attract to each other because of their magnetization property; thus, they can form clusters, which are not uniform, and some clusters may have a different color than the original beads. In that case, it is needed to define a new class for clusters to detect and quantify them.

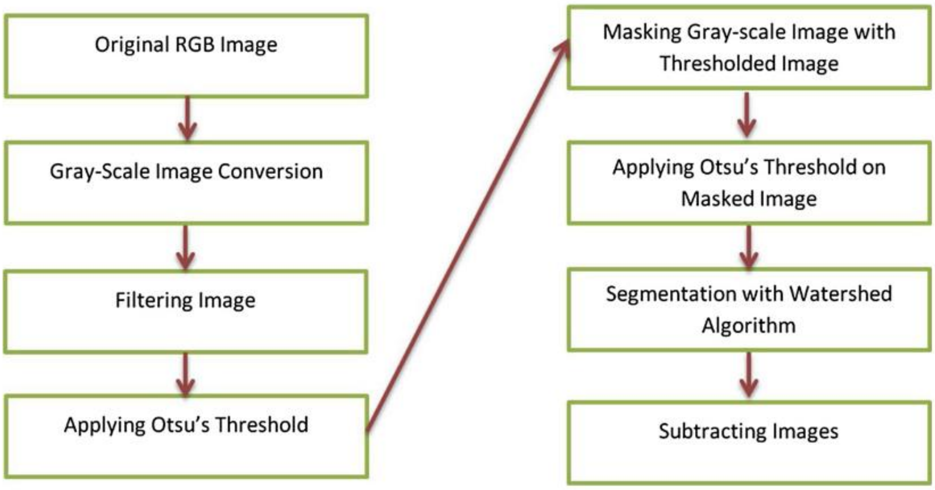

Original RGB Image
Gray-Scale Image Conversion
Filtering Image
Applying Otsu's Threshold
Masking Gray-scale Image with Thresholded Image
Applying Otsu's Threshold on Masked Image
Segmentation with Watershed Algorithm
Subtracting Images

Figure 3.4 Flow diagram of immunomagnetic bead detection and quantification steps [97].

In addition, just like micropads, beads are also opaquer than cells. The main purpose of the study is to detect and count the cells, so only bead images are obtained with the given algorithm in Figure 3.4 and subtracted from the original image to get only cell images. The main operations are filtering, thresholding, segmenting, and morphological operations to get images that have only beads. Having the information of the bead number in the image can help optimize the initial concentration of added beads, so the quantification of beads is also important for this study. In the first part of the bead detection method, beads and micropads are segmented together. Some beads may be attached to micropad borders. In order to define attached bead and micropad borders, a watershed algorithm is implemented. The watershed algorithm is not directly applied to the original RGB image. Raw images are converted to gray-scale images, I_{gray}, and to clear some noise, a Gaussian filter with 3 pixels is used. After noise cleaning, Otsu's threshold method is applied. Otsu's threshold method is given in equation 3.1

$$\boldsymbol{I_{mask}(x, y)} = \left\{ \begin{matrix} \mathbf{1}, & \boldsymbol{if\ \tau_{otsu} > I_{gray}(x,y)} \\ \mathbf{0}, & \boldsymbol{if\ \tau_{otsu} < I_{gray}(x,y)} \end{matrix} \right\} \tag{3.1}$$

Where τ_{otsu} is global threshold value found by Otsu's method, (x,y) is a pixel position in a gray-scale image. The result of the equation, I_{mask}, is a binary image and it contains beads and micropads. After this step, gray-scale version of original image is masked with I_{mask} and I_{masked} image is obtained. Then on I_{masked} image, Otsu's threshold method is applied and binary image that has bead and micropad, B_{bp}, is obtained. Then distance map, $I_{distance}$, is generated by using the Euclidean distance method [98]. After that, the watershed method labels objects in the image (O_l) by the given equation:

$$\boldsymbol{O_l \leftarrow Watershed(I_{distance})} \tag{3.2}$$

Where $l = \{1,2,3, \ldots N\}$ refers to the index of labeled objects. The main purpose of using the watershed method is to find the border of each connected object to spate them. Finally, to obtain only the bead image, B_p is subtracted from O_l, and only the bead image, which is in binary form, B_b, is obtained. Steps that are given in Figure 3.4 are visualized by original images and shown in Figure 3.5.

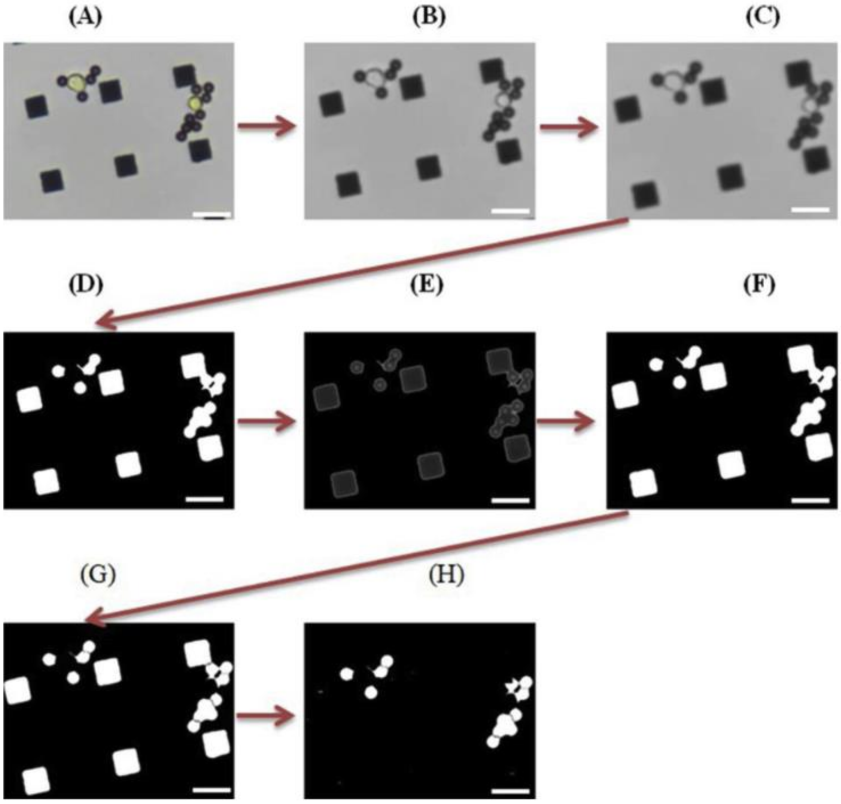


Figure 3.5 (A)Original RGB image, (B) gray-scale image, (C) filtered image, (D) binary image after Otsu's threshold, (E) masked gray-scale image, (F) binary image of masked image after Otsu's threshold, (G) binary image after watershed segmentation, and (H) binary micropad subtracted image. Scale bars = 20 µm [97].

3.2.1.3 Cell Detection and Quantification

Unlike others, cells in the images generally have irregular shapes and sizes. Also, they can be in different forms, like single cells, bounded to single or more beads, bounded to micropads, or bounded to both micropads and beads. This form makes cell detection more challenging. To detect cells, first of all, the variance matrix of the original RGB image, which is in double format, is found by the given equation:

$$\boldsymbol{V_w = \frac{1}{N-1} \sum_{i=1}^{N} \left|A_{w_i} - \mu\right|^2} \tag{3.3}$$

Where A_w is a small size matrix made up of N scalar observations, and μ is a mean of A_w. After this step, to see cells, immunomagnetic beads and micropads clearly on the variance matrix, the global threshold method is applied. The value of threshold, $2.7\text{x}10^{-4}$, is found by image segmenter tool with a trial-and-error process. Then the image contains cells, beads, and micropads in binary format is formed (B_{cbm}). The main goal of the cell detection step is to get an image that only has cells, so the next part explains how beads and micropads are removed from the B_{cbm} image.

In the watershed segmentation part of the bead detection steps, beads, micropads, and small cell fragments, which are given in Figure 3.6, are obtained. It can be seen in the image that there are small cell parts. This means that if this segmented image is directly deleted from B_{cbm}, some cells will be missed to detect and count. As a result, the cell part from the image should be cleared. To clear cells, an opening operation is applied. However, directly applying the opening to O_l image causes the remaining part of the micropads. To solve this problem, B_p is added to O_l, and then opening is applied to O_l. The opening parameter value is 225 that is half of the pixel area of the single immunomagnetic bead. The resulting image, B_s, contains beads and micropads. On B_s dilation operation is applied, and finally dilated B_s is subtracted from B_{cbm}, and only cell image, B_c is produced as:

$$\widehat{\boldsymbol{B}}_c = \boldsymbol{B}_{cbm} - (\boldsymbol{B}_s \oplus \boldsymbol{s}) \tag{3.4}$$

Where ⊕ is a dilation operation, and s is a disc shape structuring element.

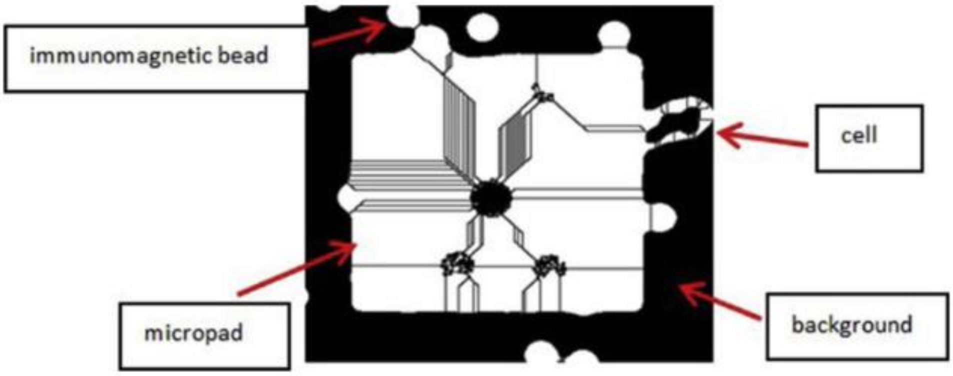


Figure 3.6 Result of the watershed algorithm. In the image there are some cells, micropad and immunomagnetic beads. Background is black. Watershed lines are visible between different objects and even in the same object [97].

After that, to clear noises from the resulted image, morphological operations are applied, and in binary image cell are counted with connected component operations [14].

3.2.1.4 Cluster Detection and Quantification

The steps of the cluster detection methods are shown in Figure 3.7. In the image, a larger micropad than the original one means that cells and beads are attached to micropads and form a cluster. So, these clusters must be identified to calculate the correct number of cells. The first operation to detect those clusters is identifying larger micropads by checking their size and producing a new image that only has larger micropads. After that, this image is added to B_b. These operations find clusters formed by micropads. In addition, it is needed to find clusters formed by beads. To do so, an opening operation with a disc-shaped structuring element is applied to find circular-shaped bead colonies (when beads form clusters around a single cell, the cluster shape becomes almost circular). As a next step, non-cluster small objects are removed, and finally, on the image, larger micropads, circular-shaped bead clusters, and certain-size bead clusters remain. On those binary images, object counting is processed to find the number of clusters.

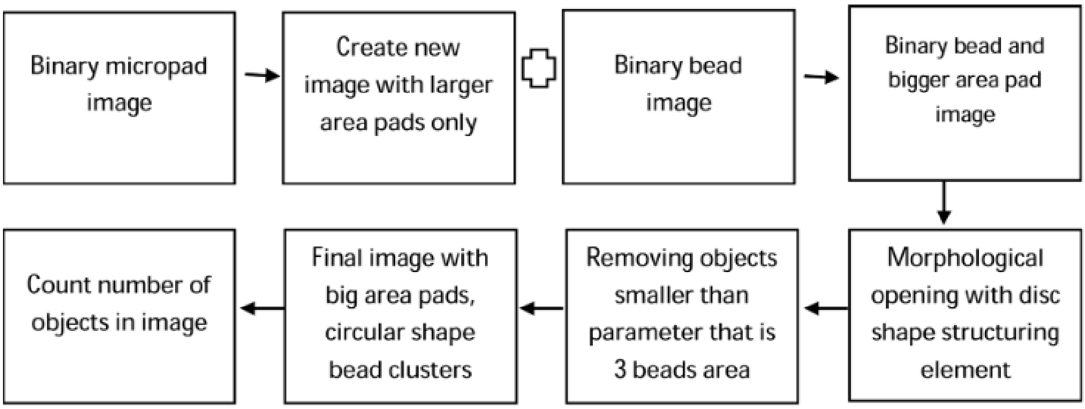

Binary micropad image
Create new image with larger area pads only
Binary bead image
Binary bead and bigger area pad image
Morphological opening with disc shape structuring element
Removing objects smaller than parameter that is 3 beads area
Final image with big area pads, circular shape bead clusters
Count number of objects in image

Figure 3.7 Flow diagram of cluster finding and counting steps [97].

3.2.1.5 Detection of Cell and Cluster Attached to the Micropads

The micropads in this study are functionalized with the CD45 antibody. It means that if a cell has CD45 antigen in the cell membrane, micropads capture this cell, and this situation gives information about the cell's immunophenotype. This case is analyzed using the following strategies: Firstly, B_p is dilated by morphological operation. Then, B_c is eroded by morphological operation. After that, dilated micropad image, eroded cell image and binary cluster are compounded C_{cp} image is created. On C_{cp} image object finding is performed with connected component algorithm and larger micropads are marked. In this step, a larger micropad means that if an object area was larger than the area of one micropad, cells or clusters were attached to the corresponding micropad.

In the following section, deep learning based methods will be explained.

3.2.1.6 Image Annotation

Deep learning techniques need well-annotated datasets for good performance. A deep learning model is trained on a labeled dataset that contains images and their corresponding pixel-labeled ground-truth images and tested on an unseen dataset. As a result, dataset images in this part of the study are annotated using Adobe Photoshop software. Some examples of dataset images and their annotations are shown in Figure 3.8. In the images, there are five different classes to segment: (1) cell, (2) immunomagnetic bead, (3) micropad, (4) cluster, and (5) background, and so the images are annotated to these five classes. During the annotation, it must be taken into consideration that not all bead clusters indicate underlying cells. If a bead cluster has an underlying cell, it must have flowing properties; three or more beads have to bound each other, and after bounding, their shape has to be circular. Besides, sometimes less than three beads can be bound to the micropad and make the cell invisible. This type of bounding is also annotated as a cluster. In Figure 3.8, examples of different cluster forms are demonstrated.

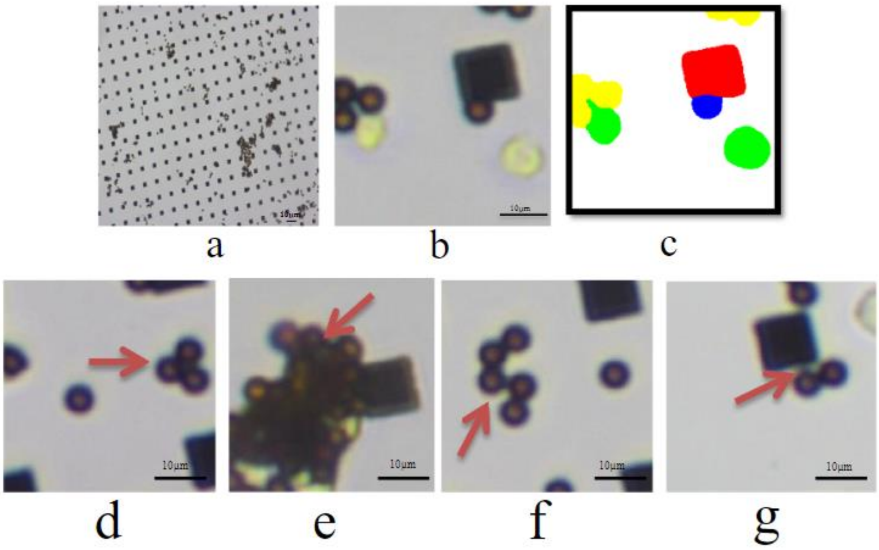


Figure 3.8 Top row: The general view of image and its annotation. a) Original bright-field microscopy image obtained from the microfluidic platform. b) Cropped image c) Annotated cropped image, green, yellow, red, blue, and white colors, respectively, represent cell, cluster, micropad, immunomagnetic beads, and background. Bottom row: Clusters can appear in 4 different forms as indicated by arrows. d) Three beads, e) more than three beads are sticked together and the shape is circular. f) More than three beads sticked together, but shape is not circular, so it is not cluster. g) Less than three beads bounded to micropad and occluded the cell [99]**.**

3.2.1.7 Overview of the MRD Biochip Image Segmentation Workflow

The semantic segmentation workflow of this study is shown in Figure 3.9. It can be seen that the original image is obtained from the MRD biochip platform, and then the image is divided into non-overlapping pieces to get small-sized images. The image size of this deep learning study is 224x224x3. Then, small images are sent to the semantic segmentation network, and as a result, each pixel is classified into five classes.

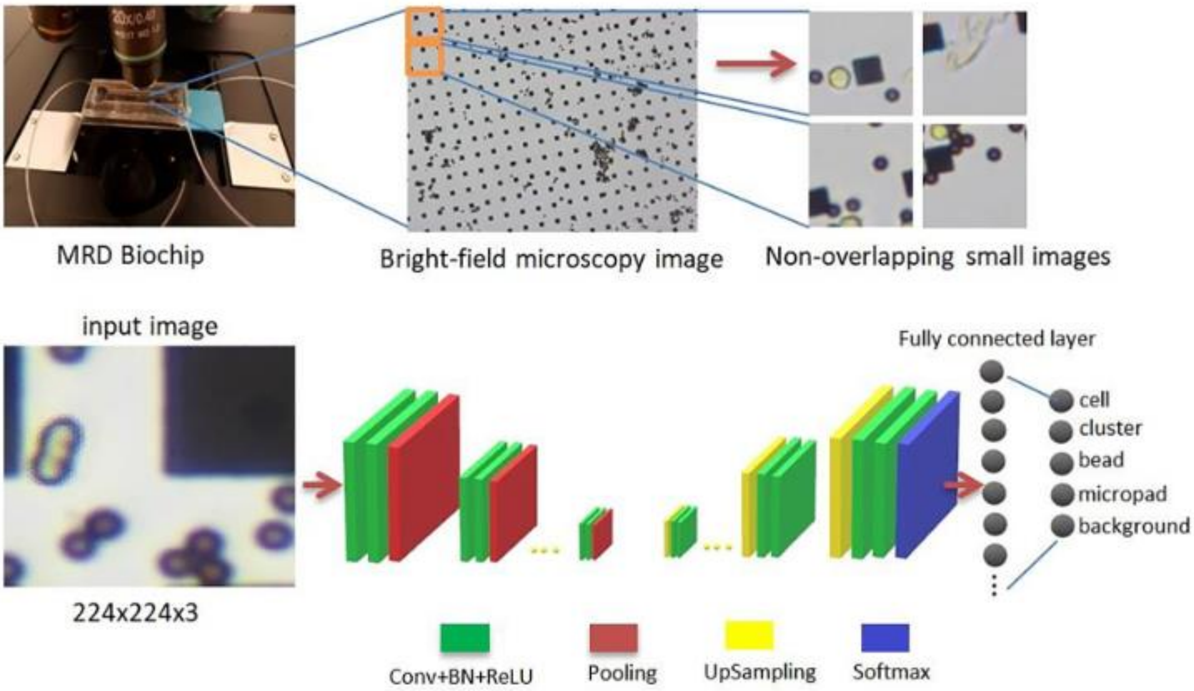


Figure 3.9 Architecture of the pixel-wise classification network. The network was trained to classify the pixels into one of the 5 classes: cell, cluster, bead, micropad, background [99].

3.2.1.8 Semantic Segmentation with U-Net Architecture

U-Net is a deep learning network that is used for semantic segmentation. It consists of a contracting path and an expansive path. A general view of the U-Net architecture in this study is given in Figure 3.10. The original U-Net architecture doesn't contain a batch normalization (BN) layer, but in this study, a BN layer is added after each convolution layer to regularize the network. As stated in [100], [101], adding a batch normalization layer to the U-Net architecture may increase performance. Architecture is commonly used for cell detection and shape measurements in biomedical images [93]. In addition, deep learning techniques benefit from large datasets, but it is proved in [102] that U-Net architecture can perform well even with small datasets.

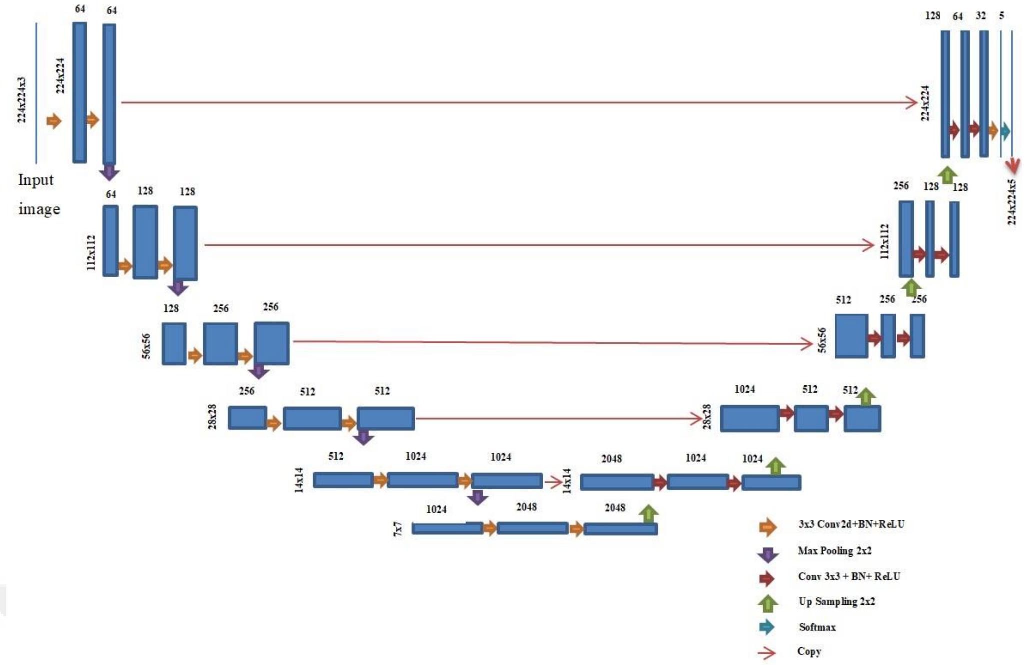


Figure 3.10 U-Net architecture. It has 3x3 Conv2d + BN + ReLU layers, Max Pooling 2x2 layers, 2x2 Transposed Conv2d layers, 2 Input Depth Concat, Softmax layer and Segmentation layer [99]**.**

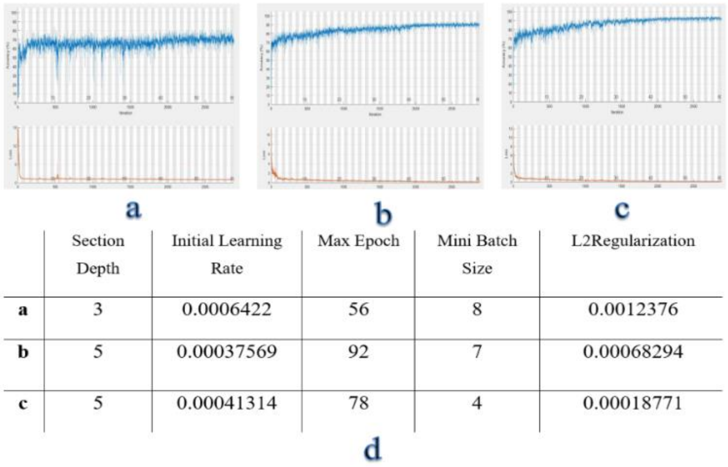
For semantic segmentation models, there is one issue that is called the unbalanced class problem [103]. In this case, the network shows poor performance on minority classes, and in this study, minority classes are the center of attention. Images for this study's coverage in terms of pixel area ratio are 72% background, 22% micropads, 5% immunomagnetic beads, 0.2% cells, and 0.8% clusters. This case shows that classes are unbalanced, so the network learns and segments cells as background and clusters as micropads or beads. To solve this problem, a transfer-learning-oriented method is used. The method mainly consists of training the network on a balanced dataset first, then using the weights of the first training to start training with those weights, and then training the network on an unbalanced dataset again [104]. Then, to obtain a balanced dataset, the same number of cell, bead, cluster, and micropad images were cropped from the big image. The number of cropped images for each class is given in Table 3.1. After cropping, images are resized by bicubic interpolation since the network needs standard and same-size images.

The dataset for this part of the study contains 90 high-resolution images, 65 of which were used for training, 7 for validation, and 18 for testing. Those numbers roughly correspond to an 80%/20% train-test and 90%/10% train-validation distribution. The network image size is immobilized at 224x224x3. That’s why high-resolution images are divided into smaller, non-overlapping images, and 21,294 image patches are obtained for the dataset.

Table 3.1 Total number of cropped images for each class to create a balanced dataset

	Single Cell	Single Cluster	Single Immunomagnetic Bead	Single Micropad	Total small images
Number of cropped images	311	229	230	221	991

For deep learning architectures, the initial value of hyperparameters is crucial and affects performance dramatically. Initial learning rate, maximum epoch, mini batch size, dropout value, and optimizer are critical hyperparameters. One example of the effect of hyperparameters on the dataset is given in Figure 3.11. It can be observed from the figure that the optimum value of parameters can increase accuracy by 20%. So it is important to find the optimum value for the network and dataset. To do that, a Bayesian optimization algorithm, which uses the information from the previous experiments to enable optimal choice of the next parameters [105] is applied to a balanced dataset. With the algorithm depth of the U-Net architecture, the initial learning rate, maximum epoch, mini batch size, and L2Regularization value are optimized. As a solver, the Adam optimizer is used, and the training dataset is reshuffled at every epoch. Besides, a learning rate scheduler with a drop factor of 0.1 and a drop period of 40 is used to decrease the learning rate while training.



	Section Depth	Initial Learning Rate	Max Epoch	Mini Batch Size	L2Regularization
a	3	0.0006422	56	8	0.0012376
b	5	0.00037569	92	7	0.00068294
c	5	0.00041314	78	4	0.00018771

Figure 3.11 (a-c): Accuracy and Loss result graphs of U-Net with different hyper-parameters. (d): Parameter values for each training in a, b, and c graphs. The graph shows that with the given hyper-parameter selection, the model (c) can learn the class representations without any overfitting or underfitting of the data [99].

3.2.1.9 U-Net, DeepLabV3 and FCN Architectures with Python

This time, codes are implemented using the Pytorch library to compare the performance of U-Net with other popular semantic segmentation algorithms, such as DeepLabV3 and fully convolutional networks (FCN). For each network, 15,444 image patches were used for training, 1638 image patches were used for validation, and 4212 image patches were used for testing. During training, the performance of networks is tested with different maximum epoch numbers and mini batch sizes. For DeepLabV3 and FCN networks, ResNet-50 and ResNet-101 backbones are implemented separately. Hyper-parameters are set to the same value for all networks to have a fair comparison. For all network models, different epoch numbers are tied, and the best epoch is chosen based on achieving minimal loss and higher cross-validation accuracy performance [106].

3.2.1.10 Quantification of Cells and Clusters

It has been mentioned before that the critical and important part of this study is to gather the number of leukemia cells to calculate density. Cells generally appear in two forms. One of them is a clearly visible cell, and the other is a cluster that is covered by immunomagnetic beads. The workflow of the counting process is shown in Figure 3.12. Segmented images are used to count cells and clusters. The segmentation algorithm in deep learning is completed on small patches, so first of all, small patches that belong to one high-resolution image are merged again to get a big image, and counting is done on this big image. The reason to quantify cells on a large image is that one cell part can be seen in different patches, but this case refers to one single cell instead of multiple cells. In addition, another reason is that it is needed to get total number of cells and clusters on the high-resolution images that represent the original dataset image. After the merging operation, cells and clusters are obtained in binary form. After this step, to fill the holes and connect broken cell parts that belong to the same cell, a morphological closing operation with a disk-shape structuring element is applied. After that, a morphological opening operation with a disk-shape structuring element to delete unconnected cell parts, cluster parts, and other noises is applied. At the end, on a binary image, cell and cluster quantification are completed by the connected components method. For the evaluation of automated quantification, the results are compared with the expert's evaluation results.

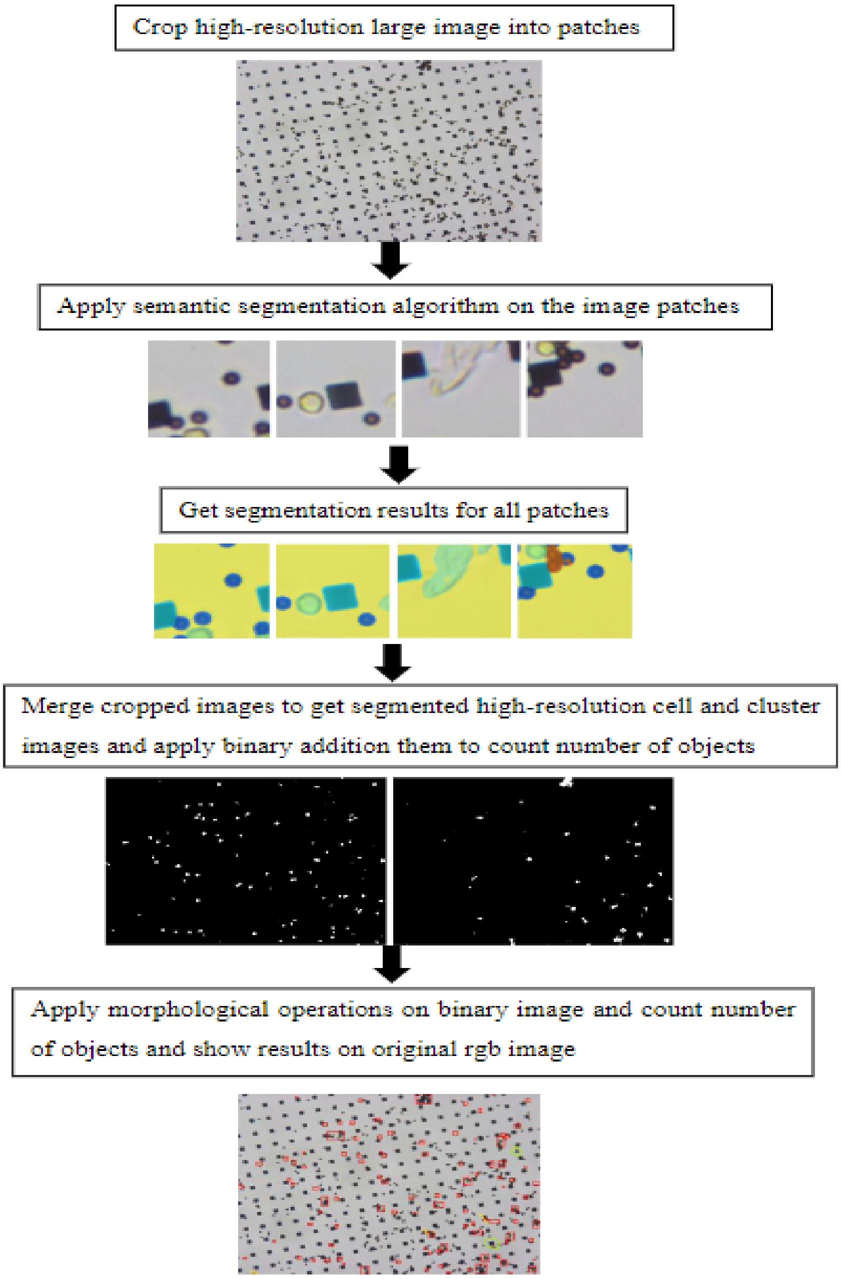

Crop high-resolution large image into patches
Apply semantic segmentation algorithm on the image patches
Get segmentation results for all patches
Merge cropped images to get segmented high-resolution cell and cluster images and apply binary addition them to count number of objects
Apply morphological operations on binary image and count number of objects and show results on original rgb image

Figure 3.12 Quantification of cells and clusters workflow [99].

3.3 Results

The result part consists of three parts: (1) the results of the first study; (2) the results of the second study; and (3) the comparison results of the first and second studies. The images that are used in two studies are recorded at different times, such as on diagnosis or after 15 days from the diagnosis, and in the first study, leukemia cells, immunomagnetic beads, clusters, and micropads are segmented and counted. In the second study, leukemia cells, immunomagnetic beads, clusters, and micropads are segmented and cells are counted. In both studies, counting results are compared with manual counting by experts. In order to determine the proposed methods in the first study, true positive (TP), false positive (FP), and false negative (FN) were found to calculate precision, recall, F-measure, and absolute percentage error [107], [108]. In order to determine the proposed methods in the second study, accuracy (ACC), intersection over union (IoU), and BF Score are calculated.

3.3.1 Quantification of Micropads and Immunomagnetic Beads

Since the micropads in the images have a regular size and shape for the same patient, they are detected with less error. In addition, immunomagnetic beads also have uniform shape and size, but when they are attached to each other or to micropads and cells, their shape and color change, and they need extra processing steps. For both immunomagnetic bead and micropad detection performance, the error rate given in equation 3.5 is calculated.

$$\boldsymbol{E} = \left|\frac{\tilde{\boldsymbol{N}}-\boldsymbol{N}}{\boldsymbol{N}}\right| \tag{3.5}$$

Where N is the actual number of the micropads or beads, $\tilde{N}$ is the number of micropads or beads which is found by the proposed algorithms. Calculated error rate for micropad is 0.01, and for bead is 0.02.

3.3.2 Quantification of Cells and Clusters Attached to the Micropads

In this study, the size of micropads changed from 15 µm x 15 µm to 35 µm x 35 µm. This situation affects the accuracy of detecting cells and clusters that are attached to the micropads. On the one hand, if the micropad size is small, cells and beads do not overlap with the micropads, so they can be identified with a higher precision rate. On the other hand, if the micropad size is large, cells and beads cannot be seen clearly, making it difficult to detect them. Table 3.2 shows the performance of the proposed methods for quantification. It can be seen that all evaluation metrics are higher than 80%.

Table 3.2 Average cells and cluster attached to micropads detection and counting performance results.

Average TP	Average FP	Average FN	Precision	Recall	F-Measure	APE	Detection Error
9.38	2.09	1.76	0.81	0.84	0.82	2.94	0.17

3.3.3 Quantification of Cells for Cell Culture Images

Cell quantification performance was estimated on two different datasets. One dataset contains images from cell culture samples, and the other dataset contains images from real patient samples. The average results of cell counting on cell culture images are given in Table 3.3. The cells in the cell culture images have higher uniformity and larger sizes compared to patient samples, so higher precision and recall rates are achieved for cell culture images. In addition, partially covered or isolated cell detection reaches a 90% recall rate, and fully covered cell detection reaches a 93% precision rate. High precision means that most of the clusters are detected correctly.

Table 3.3 Cell culture experiments: Average cell counting performance results for 20X objectives.

	Average TP	Average FP	Average FN	Precision	Recall	F-Measure	APE	Detection Error
20X objective	134.5	1.3	2.3	0.99	0.98	0.98	-0.73	0.01

3.3.4 Cell Detection and Quantification for Patient Dataset

Cell and cluster detection results for different patient images are demonstrated in Figure 3.13. From the images, it can be concluded that micropad sizes, clarity of the image depending on the micropad size, illumination effect, color, and immunomagnetic bead intensity may change.

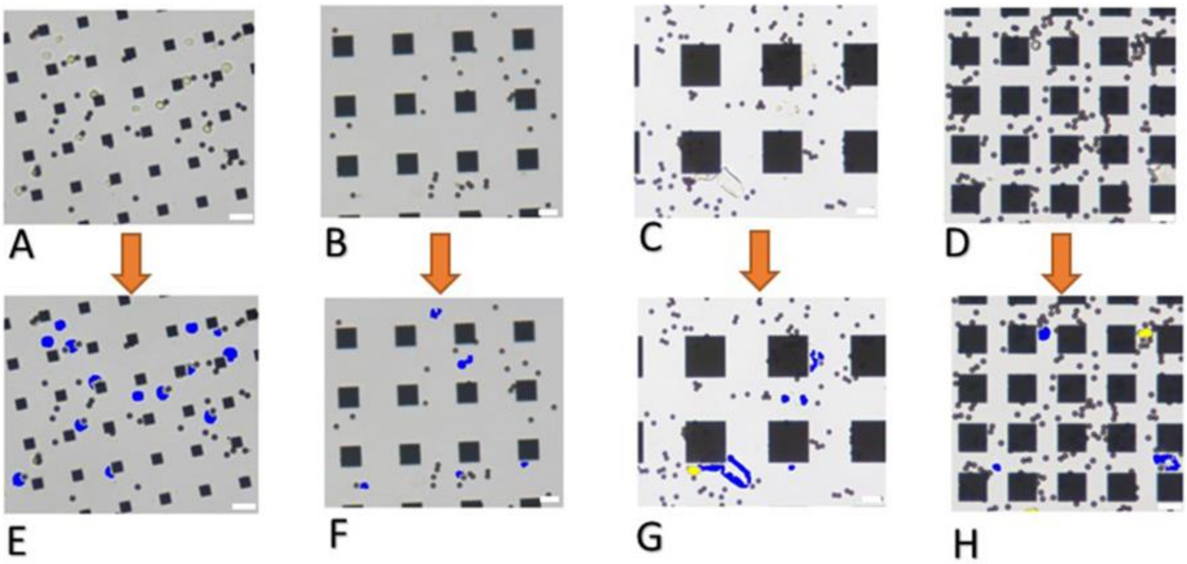


Figure 3.13 Cell detection and quantification result examples. Images belong to different patient along with different day. Blue color shows detected cells while yellow color shows detected clusters. Given images are some parts of the whole images. A, B, C, and D are original RGB images, E, F, G, and H are final result images. Scale bars = 20 µm [97]**.**

If the micropad size is small and the cells are clearer, the proposed algorithm can reach a 95% precision rate, which means it can detect most of the cells with fewer missed detections.

Also, if the micropad size is large and the bead intensity is low, the proposed algorithm can reach a 90% precision rate. Conversely, large micropad sizes cover cells and beads, making identifying cells and clusters in those images more difficult. Different experts recorded the images in the dataset, and each of them individually adjusted the microscope's light settings. This case results in variations in brightness and color. Even if images are different from each other, cells are detected with an average of 85% precision and an 86% recall value.

In this study, 12 samples, which were obtained from 4 patients, were processed. For each patient's diagnosis time, 15^{th}-day and 90^{th}-day images are collected. Besides, different-sized micropads are used to investigate the effect of size. As a result, it is concluded that the size of the micropad affected the results of the proposed algorithms. For example, a larger micropad size resulted in lower precision. In total, there are 75 images, which have a 4076 x 3116-pixel size that represents a 540 µm x 720 µm area of the chip surface. No cell exists; chip surfaces are excluded from the dataset. To see the efficiency of the proposed methods, manual counting by the expert is achieved, and results are shown in Figure 3.14. As an expected result of the applied chemotherapy, the number of blast cells should decrease on the 15th day compared to the instant of diagnosis. On the following days after the 15th day, the number of healthy cells with CD45 antigen on the cell surface increased, and the obtained trends are mostly in good agreement with the literature reporting the changes in cell numbers during the treatment [109]. Figure 3.14 shows that the trend of manual counting and automated counting is highly similar.

In addition, to investigate the statistical relationship between manual and automated counting, non-parametric The Mann-Whitney U test is used since there are two groups with independent samples, there are a limited number of samples, and the data distribution is not normal [110]. The Mann-Whitney U test is used to know if two groups are different on a variable of interest. To test two hypotheses, two are set; H_0: Statistically, there are no differences between manual and automated counting, H_1: Statistically, there are differences between manual and automated counting. Two hypotheses estimate whether the rank sum of two groups differs significantly or not. Tests are completed for all four patients separately, and patient1 has a 0.87 Asymp. Sig. value, patient2 has a 0.51 Asymp. Sig. value, patient3 has a 0.12 Asymp. Sig. value, and patient4 has a 0.56 Asymp. Sig. value. This means that four values are higher than 0.05, the hypothesis H_0 is retained, and there are no differences between manual and automated counting.

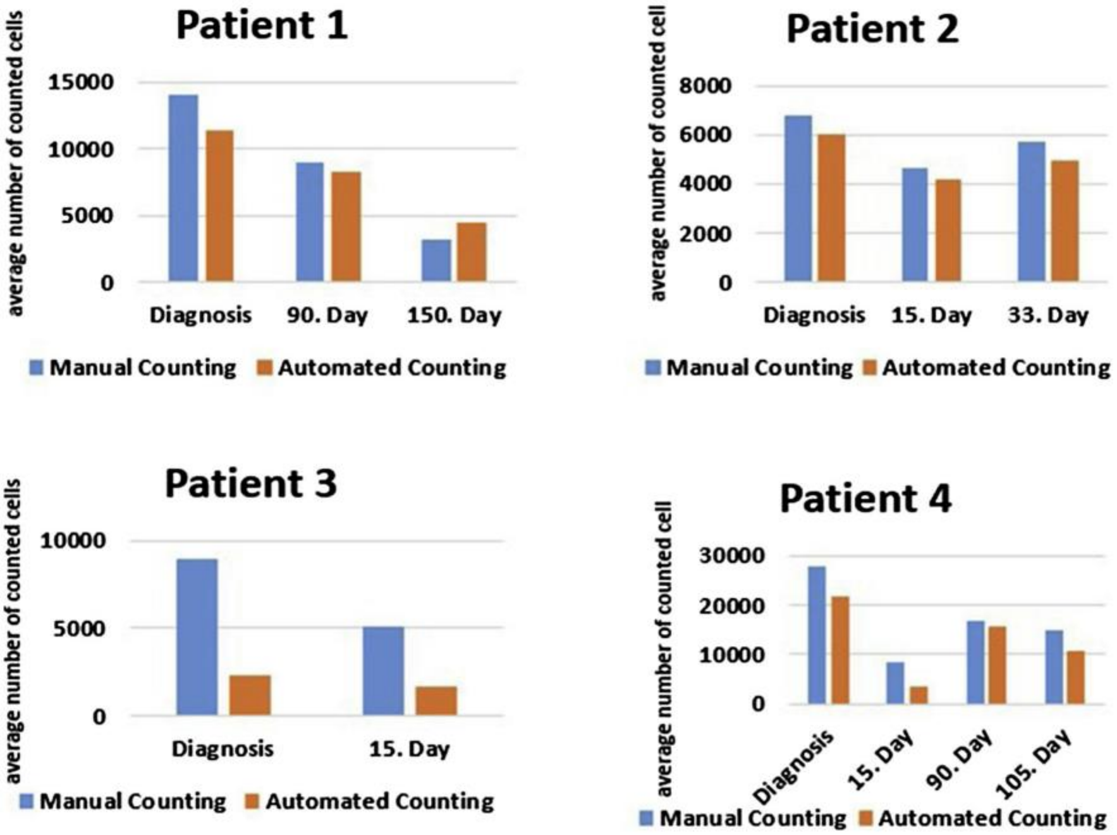


Figure 3.14 Results of proposed algorithm on different patients. Each graph shows comparison of manual counting and automated counting for different patient [97].

The smallest Asymp. Sig value was calculated for patient 3, and in Figure 3.14, even though the trends were similar, the average number of cells obtained from the manual counting and automated counting deviated more compared to the results for the other patient samples. The main reasons for this deviation were 1) using larger micropads (35 µm x 35 µm) for this patient, and 2) the intensity of the micropads was higher compared to other images. When micropads overlapped and blocked the visibility of the cells, the proposed algorithm could not detect every cell in the image. For patient 1, small micropads (15 µm x 15 µm) were used, and cells were not blocked by the micropads in the image. As a result, automated cell counting yielded similar numbers to manual counting.

3.3.5 Segmentation Result of U-Net Architecture

Performance measurement of the architecture on this study dataset is achieved by calculating accuracy, IoU [111], and BF-Score [112] values. Accuracy represents the percentage of correctly classified pixels and is calculated as follows;

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.6}$$

Where TP, TN, FP, FN represents true positive, true negative, false positive and false negative, respectively.

The IoU metric is commonly used as a performance estimator for segmentation algorithms in deep learning, and it refers to the ratio between overlap and union of ground-truth labels and prediction labels. It is calculated as follows;

$$\boldsymbol{IoU = \frac{ground\ truth\ labels \cap prediction\ labels}{ground\ truth\ labels \cup prediction\ labels}} \tag{3.7}$$

BF-Score indicates how well boundary of prediction of each class aligns with the true boundary. Equation of BF-score is given as follow;

$$\boldsymbol{BF\ Score = \frac{Precision * Recall * 2}{Precision + Recall}} \tag{3.8}$$

The best performance of U-Net architecture is obtained with Adam optimizer, learning rate 0.00041314, maximum epoch 16, and mini batch size 8. The result matrices for each class are given in Table 3.4. Image annotation is completed on a high-resolution image, and then the image is divided into non-overlapping patches to train and test. That's why the part in which beads are attached to each other and form a circular shape cannot seem as circular in patches. Thus, in patches, those parts are missed and segmented as beads, and the result of IoU for cluster in Table 3.4 is lower since IoU value is calculated by division of overlap area between ground truth and prediction image. The average value of matrices for all classes on test data is shown in Table 3.5. Results show that each matrix reaches the values of more than 80%.

Table 3.4 U-Net class metrics for each class.

Class Name	**Accuracy**	**IoU**	**MeanBFScore**
Cell	0.856	0.717	0.713
Cluster	0.902	0.637	0.639
Bead	0.917	0.893	0.941
Micropad	0.985	0.981	0.983
Background	0.997	0.991	0.991

Table 3.5 U-Net segmentation average metrics for 5 classes.

Global Accuracy	**Mean Accuracy**	**Mean IoU**	**Weighted IoU**	**Mean BF Score**
0.99	0.93	0.84	0.98	0.95

The original U-Net architecture doesn't have a batch normalization (BN) layer, but in this study, a BN layer is added after each convolutional layer as a regulator. The effect on average metrics of adding the BN layer to the dataset is given in Table 3.6. In addition, adding BN layer effects to class matrices is given in Table 3.7. Those results were obtained with a small number of datasets. Both table results show that adding the BN layer increased the overall performance of the U-Net network by increasing the performance of all evaluation matrices.

Table 3.6 U-Net segmentation average metrics with BN, and without BN layer.

	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean-BF Score
U-Net	97.0%	83%	73%	94%	85%
U-Net+BN	**97.5%**	**87%**	**78%**	**95%**	**87%**

Table 3.7 U-Net segmentation class metrics with BN, and without BN layer.

Class Name	Accuracy	Accuracy(U-Net+BN)	IoU	IoU(U-Net+BN)	MeanBF Score	MeanBFScore(U-Net+BN)
Cell	0.746	**0.799**	0.533	**0.593**	**0.536**	0.497
Cluster	0.575	**0.656**	0.524	**0.632**	0.262	**0.396**
Bead	0.909	**0.941**	0.701	**0.760**	0.775	**0.820**
Micropad	0.978	**0.983**	0.943	**0.948**	0.902	**0.921**
Background	**0.986**	0.979	0.979	**0.979**	0.948	**0.951**

Moreover, test results are rechecked with the Pytorch implementation of the U-Net architecture. As performance matrices on the dataset, accuracy, precision, recall, f1-score, and mean IoU values (the same as BF score) are calculated. Results are given in Table 3.8. The mean IoU for all classes is 86.1%. The given results are achieved with an initial learning rate of 0.00014314, a maximum epoch of 50, a mini batch size of 16, and a U-Net depth of 5. Accuracy for cell, bead, and micropad segmentation is increased since a larger mini batch size and maximum epoch number are able to train the architectures.

Table 3.8 Evaluation matrices result of U-Net + BN architecture with Pytorch

	cell	cluster	bead	micropad	background
Accuracy	89%	78%	96%	97%	99%
Precision	84%	86%	93%	99%	99%
Recall	89%	78%	96%	98%	100%
F1-score	86%	82%	95%	98%	99%

3.3.6 Segmentation Results of DeepLabV3 and FCN Architectures

The proposed U-Net architecture is compared with other semantic segmentation algorithms, such as DeepLabV3 and FCN. To compare the results, the same evaluation matrices are calculated, and the results are given in Tables 3.9, 3.10, 3.11, and 3.12. Results reveal that different architectures give different performances on this study dataset. However, to see the overall performance of each architecture on this study dataset, the mean accuracy, mean IoU, mean precision, mean recall, and mean F1-score are calculated and given in Table 3.13.

Table 3.9 Evaluation matrices result of DeepLabV3-Resnet50 architecture.

	cell	cluster	bead	micropad	background
Accuracy	90%	78%	96%	95%	99%
Precision	82%	87%	93%	99%	99%
Recall	91%	78%	96%	95%	100%
F1-score	86%	82%	94%	97%	99%

Table 3.10 Evaluation matrices result of DeepLabV3-Resnet101 architecture.

	cell	cluster	bead	micropad	background
Accuracy	90%	79%	96%	95%	99%
Precision	83%	87%	92%	99%	99%
Recall	91%	79%	96%	96%	99%
F1-score	87%	83%	94%	98%	99%

Table 3.11 Evaluation matrices result of FCN-Resnet50 architecture.

	cell	cluster	bead	micropad	background
Accuracy	91%	79%	96%	95%	99%
Precision	82%	87%	93%	99%	99%
Recall	91%	79%	96%	95%	100%
F1-score	87%	83%	95%	97%	99%

Table 3.12 Evaluation matrices result of FCN-Resnet101 architecture.

	cell	cluster	bead	micropad	background
Accuracy	90%	80%	95%	95%	99%
Precision	83%	85%	93%	99%	99%
Recall	90%	80%	96%	96%	100%
F1-score	86%	83%	94%	98%	99%

Table 3.13 Overall performance comparison between 5 different semantic segmentation architectures.

	U-Net +BN	**DeepLabV3-Resnet50**	**DeepLabV3-Resnet101**	**FCN-Resnet50**	**FCN-Resnet101**
Accuracy	**98.7%**	98.2%	98.3%	98.2%	98.3%
IoU	**86.1%**	85.6%	86%	85.8%	85.8%
Precision	**92.2%**	92%	92%	92%	91.8%
Recall	**92.2%**	92%	91.6%	92.2%	92.4%
F1-Score	**92.2%**	92%	91.7%	92.1%	92%

From Table 3.13, it can be concluded that the proposed U-Net, DeepLabV3, and FCN semantic segmentation algorithms have similar performance on this study dataset when overall performance matrices are compared. But even though the difference is small, the proposed U-Net gives the best results on matrices. Besides, deeper networks such as DeepLabV3-ResNet101 and FCN-ResNet101 show better performance on accuracy, IoU, and recall values compared to ResNet50 networks.

3.3.7 Automated Quantification After Segmentation

In the first study of this chapter, automated quantification results for each object are given. In the second study of this chapter, segmented cells and clusters are counted by morphological operations, and manual counting is used for comparison. In both studies, cell and cluster counting are critical, and other objects are counted with less error (0.02 in the first study), so in this part only cell and cluster counting are completed. The quantification results of test images are shown in Table 3.14. It can be seen that recall reaches 97% success. This indicates that there are a low number of false positives.

Table 3.14 Overall performance of quantification of cells and clusters on test images.

Average TP	Average FP	Average FN	Precision	Recall	F-Measure	Detection Error
46.5	3.6	1.2	0.89	0.97	0.93	0.06

As a last part, to see the improvement in the second study compared to the first study, the quantification results of cells and clusters on test images from the first study and the second study are compared, and the results are shown in Table 3.15. Results show that, with the second study, all evaluation matrices have improved. In addition, in the second study, the Mann-Whitney U-test is also used to show statistical relations between manual and automated counting. This time test is completed on all patients' datasets, and a 0.64 Asymp. Sig. value, which is larger than 0.05, is obtained. This means that the hypothesis H_0 and it is concluded that there are no statistical differences between manual counting by experts and automated counting.

Table 3.15 Comparison of quantification results between our previous study and proposed study on test images. Improvement rate is calculated as (Proposed method's result – the previous method's result) / |ideal result - previous result| * 100.

	Precision	Recall	Detection error	F-Measure	Max-Precision
First study	81%	84%	16%	79%	95%
Second study	89%	97%	6%	93%	99%
Improvement Rate	**9.8%**	**15.4%**	**62.5%**	**17.7%**	**4.2%**

3.4 Discussion and Conclusion

Cells are stained for separation and visualization purposes [113]. In addition, immunomagnetic beads are stained and used to detect breast cancer-associated cytokeratin cells [65]. In this study, the biochip design has immunomagnetic beads. Beads are used for inherent cell separation and as a visual marker for cell detection. Stained cell detection from microscopy images is reported with more than 90% accuracy and precision values [68]–[70], but cell staining makes the system easy to detect. In this study, the images have cells, immunomagnetic beads, and square micropads, which makes cell detection more challenging. In the first part of this chapter study, image processing-based methods are developed to detect five object classes in the image. The developed system is simple, fast, and effective on the dataset. Algorithms are tested on two types of datasets: cell cultures and patient samples. Since cells are clearer on cell culture images, proposed methods give a higher precision rate on those images. In addition, for some patient samples, the micropad size is large, which makes it difficult to detect cells. The average run time for simple image processing algorithms is 90 seconds. The developed system incorporated color, size, and shape-based methods to identify objects. A smaller micropad size gives better detection accuracy.

The image objective of this chapter is 20X. Without cell staining, color-size and shape-based methods achieved 85% precision, 86% recall, and an F-measure for the patient dataset. The results show that, even if micropad size and bead intensity change, the proposed system can still find cells. 98% precision and recall are obtained with clearer cell culture images. After detection, objects are counted. The output of the proposed method of counting is compared with that of manual counting. In addition, for patient samples, a statistical test is used to compare the output of the algorithm with the output of the medical expert's evaluation, and the test revealed that the manual and automated counting are statistically in agreement.

Segmentation of cells from microscopy images is reported with 96.1% average accuracy and 92.1 average IoU [114]. However, cells are stained, and there is no unbalanced dataset problem. In the second part of this chapter study, deep learning-based methods are developed to detect five object classes in the image. Proposed methods achieved 98.7% average accuracy, 86.1% average IoU, and 92% average BFScore. During the first part of this study, different methods for each object are implemented, parameters are optimized, and for low-quality images, pre-processing steps are implemented. Those things make the system less fully automated. That's why, on the patient dataset, deep learning-based methods are applied to increase accuracy and make the system fully automated and robust. The developed system uses a hyper-parameter optimized convolutional neural network architecture (U-Net + BN) to segment objects. U-Net performance is compared with other semantic segmentation models. As a result, all architectures show similar results, but U-Net shows the best performance. The unbalanced dataset problem dramatically affects the performance of the network, and in this study, the problem is solved by training the network on a balanced dataset first and then using pre-trained weights while training the unbalanced dataset. Besides, data augmentation techniques were tried in the study to increase the performance of the network, but they didn't increase the performance, so they were not used in the final test results. The system doesn't require further processing after parameter optimization. Segmentation accuracy for each class is more than 85%, and IoU for each class is more than 65%. After segmentation, cells are quantified. Results show that the first part of the study improved with the second part of the study. Improvements are given in Table 3.15. Also, in this part, statistical analysis is completed, and results reveal that deep learning-based cell segmentation and counting can be used as a readout method for the MRD biochip.

Chapter 4

Improving Multi-Class Senescent Cell Segmentation Using Self-Supervised Learning on Bright-Field Microscopy Images

4.1 Introduction

Mesenchymal stem cells (MSCs) are adult stem cells, and they play important roles in self-renewal and multidifferentiation. They have extensive available sources such as cord blood, amniotic fluid, bone marrow, and adipose tissue. Since they are extensively available, they are used for therapeutic purposes and research [114]. In Figure 4.1, the sources of MSCs and their usage are shown. It can be seen that those cells are common in clinical studies. They are used in regenerative medicine, immunological diseases, and cancer treatment [115]. MSCs exert their therapeutic effects largely through their differentiation capacity into multiple cell lineages and paracrine actions. However, they lose their therapeutic effect due to aging and cellular senescence. Cellular senescence is a case where the cell cycle arrests irreversibly. Also, the secretion profile of MSCs is changed by senescence. Senescence-associated secretory phenotype (SASP) is a multicomponent phenotype released in large amounts by senescent MSCs that comprises pro-inflammatory chemicals, immune modulators, and growth factors [116]–[118]. SASP can be a driver of metabolic disease, as shown in Figure 4.2. That’s why an understanding of the MSC aging process is needed to be able to use it in clinical applications. It is crucial to detect and quantify senescent MSC in the population.

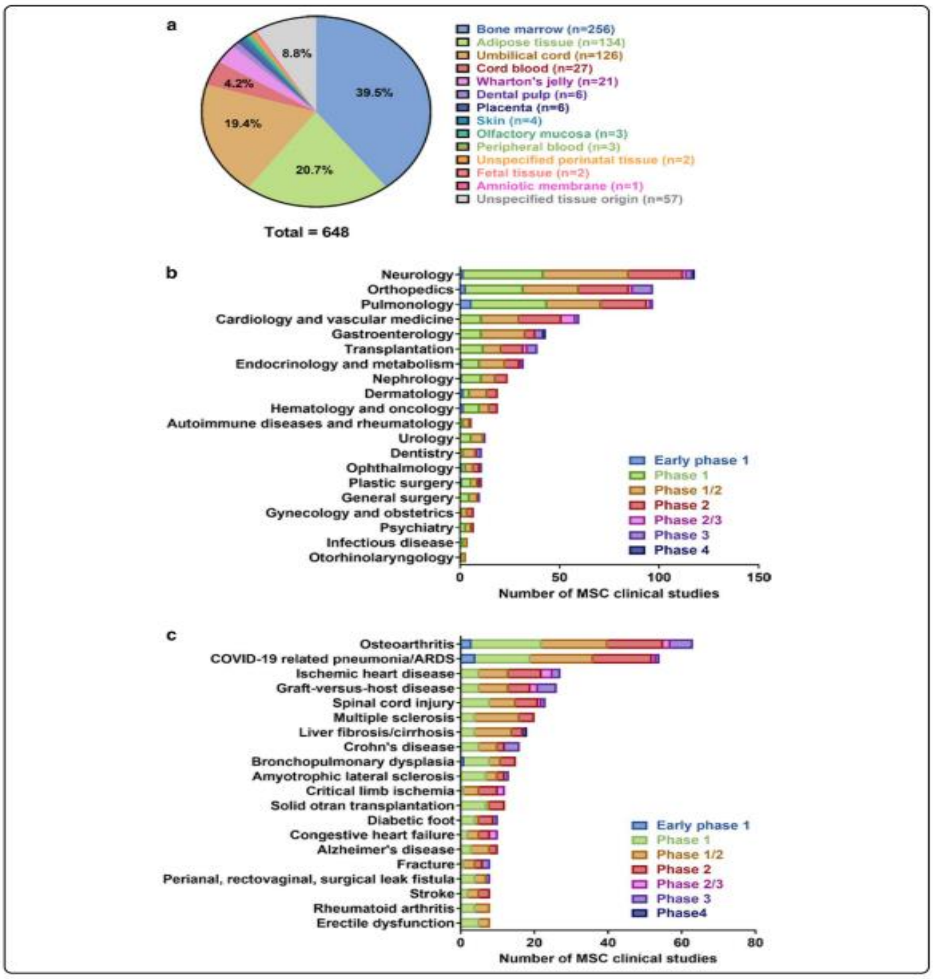


Figure 4.1 MSCs sources and their usage in clinical studies [114].

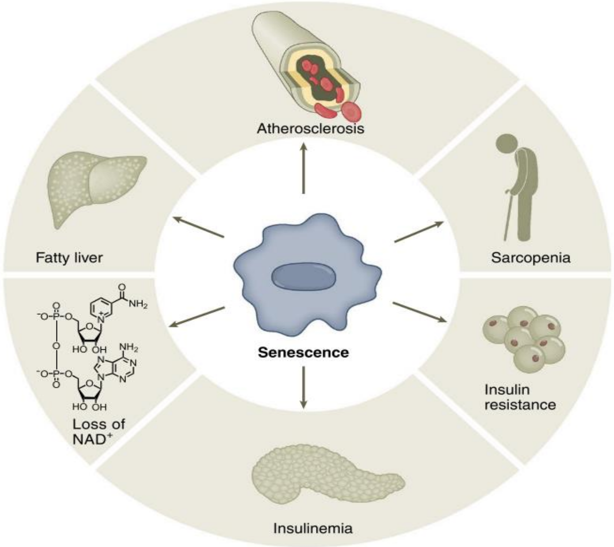


Figure 4.2 Relationships between metabolism and cellular senescence [116].

One way to detect cellular senescence is the beta-galactosidase assay (b-Gal), but it has disadvantages like cell confluency, serum depletion, and operator bias. Operator bias can cause false positives, and when some senescent cells don't show b-Gal activity, this causes false negatives [119]. Moreover, to be able to use MSC in clinical applications, Good Manufacturing

Practices (GMP) must be followed during manufacture. GMP requires a well-organized organization of qualified staff, specialized and controlled equipment, and multiple quality controls [120]. The protocol of b-Gal to identify senescent cells is not easily adaptable to the GMP pipeline and is time consuming. In addition, senescent cell morphology can be different, so image processing-based detection and counting of senescent cells can be easy. In [49], [97], conventional image processing techniques are applied to analyze leukemia cells, and in another study [47], [48], machine learning-based techniques are proposed for cell analysis. The majority of these machine learning techniques require labeled data. Deep learning techniques are also successful for identification and quantification. One example is that label-free senescent cell analysis for endothelial cells is achieved by convolutional neural networks [121]. Moreover, because MSCs have unique cell shapes and morphologies, there is no available image-based analysis method for their detection and quantification.

Deep learning techniques are broadly applied to various fields with high performance [122], and they are applied to biomedical images for segmentation, classification, and object detection purposes. X-ray images, pathology images, radiology images, and cancer images are analyzed by deep learning techniques [94]. Also, deep learning techniques are implemented for cellular image analysis [26]. Cells are segmented using convolutional neural networks [123]–[125]. In another study, two fully convolutional regression networks are implemented on fluorescent microscopy cell images to detect and count cells, and the results of both networks were compared [126]. ImageJ is an open-access software tool modified by adding U-Net architecture for segmentation purposes for non-machine-learning experts. The software lets the user analyze their data either on a local computer or on cloud services [93]. Cellular senescence in cell culture is predicted by a nuclear morphology-based neural network [127]. In [128], Mask R-CNN is applied to segment cell nuclei images that are acquired from microscopy by initializing weights from a pre-trained Mask R-CNN model on the MSCOCO [129] dataset with a 70.54% IoU value. In addition, a pre-trained Mask R-CNN model is used to segment microscopy cell nuclei images, which are stained with hematoxylin and eosin with a 45.02% IoU value [130]. In another study, overlapping cells were segmented by Mask R-CNN based methods [131].

Deep learning techniques require a large amount of well-labeled data. However, labeling data is time-consuming and requires an expert in that field. To overcome the small number of labeled dataset problems in deep learning, transfer learning techniques are applied. Instead of training the network from beginning to end to obtain weights, transfer learning initializes the weights of the network using pre-trained network weights. There are some models that have pre-trained weights like Mask R-CNN, AlexNet, VGG-16, etc., but those weights are obtained by training the models on natural images, which are different than medical images. In addition, it is proven that transfer learning from in-domain data improves network performance compared to transfer learning from out-domain data [132], [133]. That’s why, in this study, a network that is pre-trained by in-domain data with a smaller number of labeled data is implemented and results are obtained. To alleviate the large number of labeled data requirements, a self-supervised learning methodology was developed for the network[134].

During this study, it is needed to segment and count the number of senescent cells, so the Mask R-CNN instance segmentation method is implemented in a self-supervised learning (SSL) manner. SSL on the in-domain dataset increased the mean average precision metric of Mask R-CNN. This study proves that, even if a small number of unlabeled datasets are available, in-domain dataset SSL pre-training exceeds the performance of fine-tuning downstream labeled dataset training tasks.

4.2 Experimental Section

4.2.1 Materials and Methods

For this study, 342 cell culture images were used. Images are captured by a color camera with a 10X objective. The size of the images is 2592 x 1944 pixels. In one image, both young and senescent cells can exist in more than one.

Adipose tissue-derived mesenchymal stem cells (MSCs) were obtained from the American Type Culture Collection (ATCC PCS-500-011). MSCs were grown in DMEM supplied with 10% FBS (Fetal Bovine Serum), 100 U/mL penicillin-streptomycin, 4 mM L-glutamine, and 5 ng/mL bFGF at 37°C in 5% CO_2 environment.

Senescence of the mesenchymal stem cells was induced by the addition of 300 μM H2O2 in PBS for 30 minutes. Then, cells were washed 3 times with PBS and left in a complete growth medium.

Visualization of the senescent and young stem cells achieved after fixation of the cells. MSCs cells were washed with PBS three times. Then, cells were incubated in a 10 mL fixative solution (0.2% glutaraldehyde in PBS) for 15 minutes. Cells were washed with PBS to clean up the remaining glutaraldehyde. Fixed cells stored with the addition of 50% glycerol.

4.2.1.1 Instance Segmentation with Mask R-CNN

In this section, the implementation of segmenting medical images with the convolutional instance segmentation algorithm is explained. The original method's performance is improved by the SSL method, which is explained in the next section.

As architecture, Mask R-CNN is used. Mask R-CNN is an instance segmentation algorithm, and it is implemented on top of Faster R-CNN. The network mainly classifies each pixel into a fixed set of categories and at the same time, precisely segments each instance. In this study, the categories are senescent cell, young cell, and background. A schematic view of the architecture of this study dataset is shown in Figure 4.3.

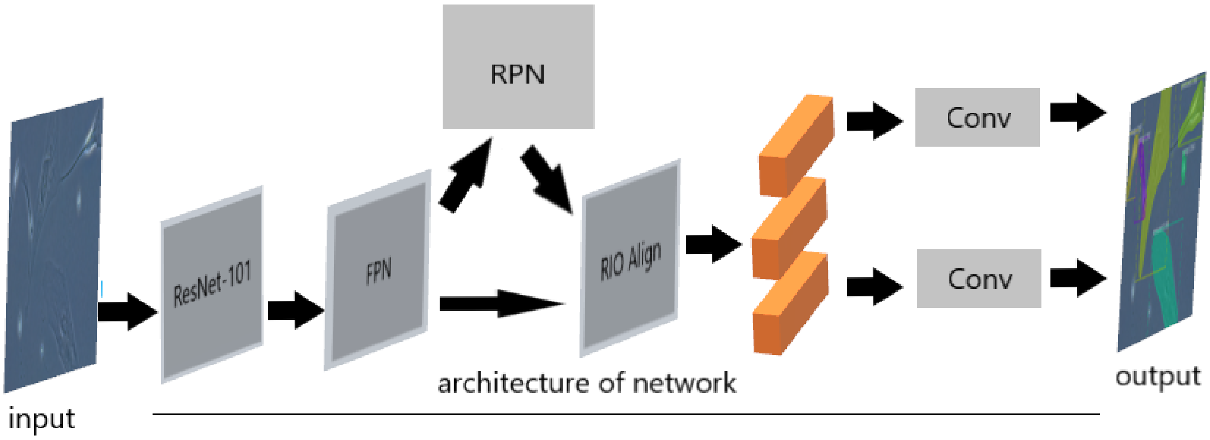


Figure 4.3 General view of Mask R-CNN architecture used on our dataset. Left to right; input image, architecture of network and output image. It takes an input image and returns output image with mask of each instance and their bounding boxes.

Architecture has two stages. While the first stage has a backbone network (ResNet-101, VGG, Inception...) to extract features and a region proposal network, the second stage returns confidence, bounding boxes, and binary masks, which are obtained via regions of interest in the image. The input image is pushed to architecture, and candidate objects with bounding boxes are formed. Then, a fully connected network classifies the proposed regions. The proposed region might be of arbitrary size, but the fully connected classifier layer requires a fixed-size image. To have the same size, networks use RoIAlign methods. The output of the RoIAlign layer goes to the mask head layer, and the layer generates a mask for each RoI with the predicted classes [135].

To be able to estimate cellular senescence, the number of senescent and young cells in one image is needed. To segment and then get the numbers of objects, in bright-field microscopy images, the Mask R-CNN instance segmentation method is used. As a backbone network, ResNet-101 with feature pyramid network (FPN) is implemented. Inc., released under an MIT License implementation [136], which is based on Keras [137] and Tensorflow [138] libraries, is used. In the first phase of the study, instead of training the network end-to-end from the start, a transfer learning technique that initializes the model weights with pre-trained model weights is applied. Training after weight initialization has two approaches: (1) training the network head layers, and (2) training all layers of the network. Data augmentation is applied to increase the dataset during training. The techniques used for data augmentation are vertical and horizontal flipping, rotation, scaling, and adding blur.

4.2.1.2 Instance Segmentation with Mask R-CNN and SSL

Training deep learning models requires a vast amount of high-quality labeled data, and data labeling is a challenging task due to time and cost. Therefore, transfer learning is a simple and effective way to overcome the small number of labeled data problems while training for both natural and medical images. In addition, transfer learning from the same domain data improves performance further. However, medical image labeling is more challenging because of the structure of the objects in the images.

On natural images, it is proven that self-supervised contrastive pre-training has shown improvements in the performance of deep learning models [139], [140]. Despite the success of SSL on natural images, the application of SSL on medical images is limited and needs to be explored. The self-supervised learning technique allows pre-training with unlabeled domain-specific images to learn more pertinent features for that domain [141]. In this study, SSL is applied to train the ResNet-101 backbone network of Mask R-CNN with a domain-specific dataset. A general view of the SSL-based instance segmentation workflow of this study is shown in Figure 4.4. The pre-training technique in this study consists of two approaches: (1) training the network end-to-end from the start; (2) initializing the weights from ImageNet [142], then training the network.

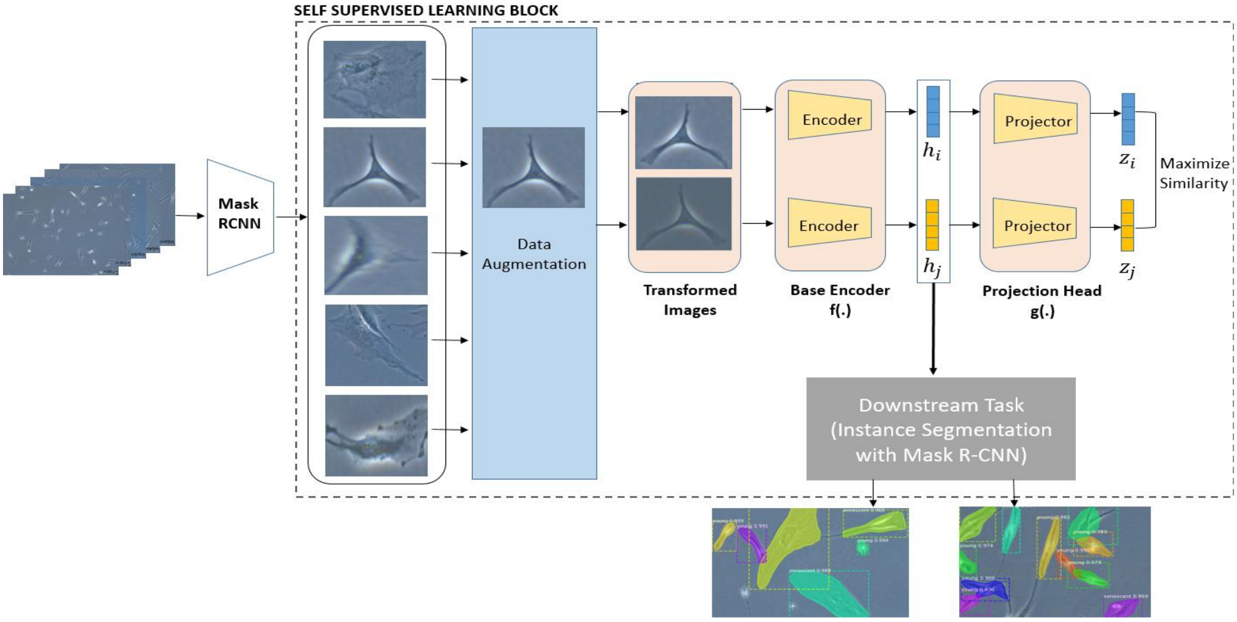


Figure 4.4 The proposed self-supervised instance segmentation on microscopic images workflow. It comprises of three steps. First, applying Mask R-CNN on unlabeled high-resolution images to get single object patches. Second, unsupervised learning to learn general representations of our microscopy image patches. Third, fine-tuning and training on labeled dataset, and then instance segmentation by using Mask R-CNN architectures.

After SSL pre-training, backbone network weights are used as initialization weights for downstream supervised training tasks on labeled data. Later, the results of both supervised pre-training on natural images and fine-tuning on labeled medical datasets and self-supervised pre-training on in-domain medical images and fine-tuning on labeled medical datasets are compared.

There are some SSL methods that are applied to medical images, such as SimCLR [134], MoCo [140], and context restoration [143]. In this study, SimCLR, which is a state-of-the-art method, is implemented. The method takes an image and applies basic augmentations (random cropping, random color distortions, and Gaussian blur) to form two transformed images. Then the method learns general representations by maximizing agreements between the same image transformations and minimizing agreements between different image transformations. As an augmentation random flip, color distortion with strength 1 is implemented in this study.

The number of images and their sizes that are used for SSL and downstream tasks are given in Table 4.1. First of all, 14 high-resolution images are labeled by three medical experts. Then, the labeled images are trained with Mask R-CNN by using the transfer learning approach. After that, unlabeled 242 high-resolution images are used as test images to extract unlabeled single cell patches since Mask R-CNN returns the mask of each instance's bounding boxes. A total of 8719 patches are obtained and used for SSL pre-training. Extracted unlabeled image sizes are different from each other, but a deep learning model needs a fixed dataset size, so each of the 8719 images is resized to 224x224x3. Finally, to show the performance of downstream labeled training after SSL, in addition to 14 labeled images, 86 images are labeled, and in total, 100 high-resolution labeled images are formed for downstream tasks.

Table 4.1 Number and properties of images used for self-supervised learning and fine-tuning.

	Total Number of Images	Image Sizes	Number of Objects	Labeled
Self-supervised Learning Raw Images	242	2592x1944	Multiple	No
Self-supervised Learning Patches	8719	224x224	1	No
Fine Tuning	100	2592x1944	Multiple	Yes

4.2.1.3 Implementation Details of Mask R-CNN with Transfer Learning from MSCOCO Dataset

In this part, the transfer learning approach is achieved by initializing the weights of Mask R-CNN from a pre-trained model, which is obtained after training with the MSCOCO dataset. Transfer learning is used to segment microscopic cell images. Labeled 100 high-resolution images are randomly divided into 50 training, 20 validations, and 30 test datasets. The training dataset is both trained with and without augmentation. The effect of data augmentation on this study dataset is demonstrated in Figure 4.5. It can be seen from the figure that data augmentation dramatically affects the model's performance by decreasing validation loss.

While training, the initial parameters are set as follows: minimum confidence 0.9, steps per epoch 100, images per GPU 1, weight decay 0.0001, and learning rate 0.001. The maximum number of epochs was set to 100, and an early stopping technique was used [106].

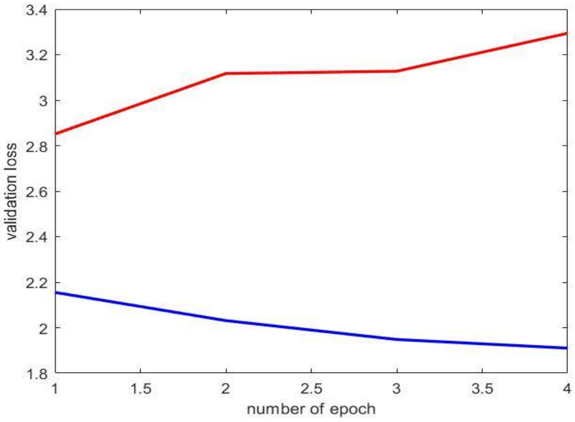


Figure 4.5 Effect of data augmentation on our dataset. Blue color indicates validation loss with augmentation, and red color indicates validation loss without augmentation.

After the training phase, the model is tested on labeled test images. The model returns an image with segmented object masks, object labels with probability values, and bounding box coordinates for each object. The resultant images are shown in Figure 4.6.

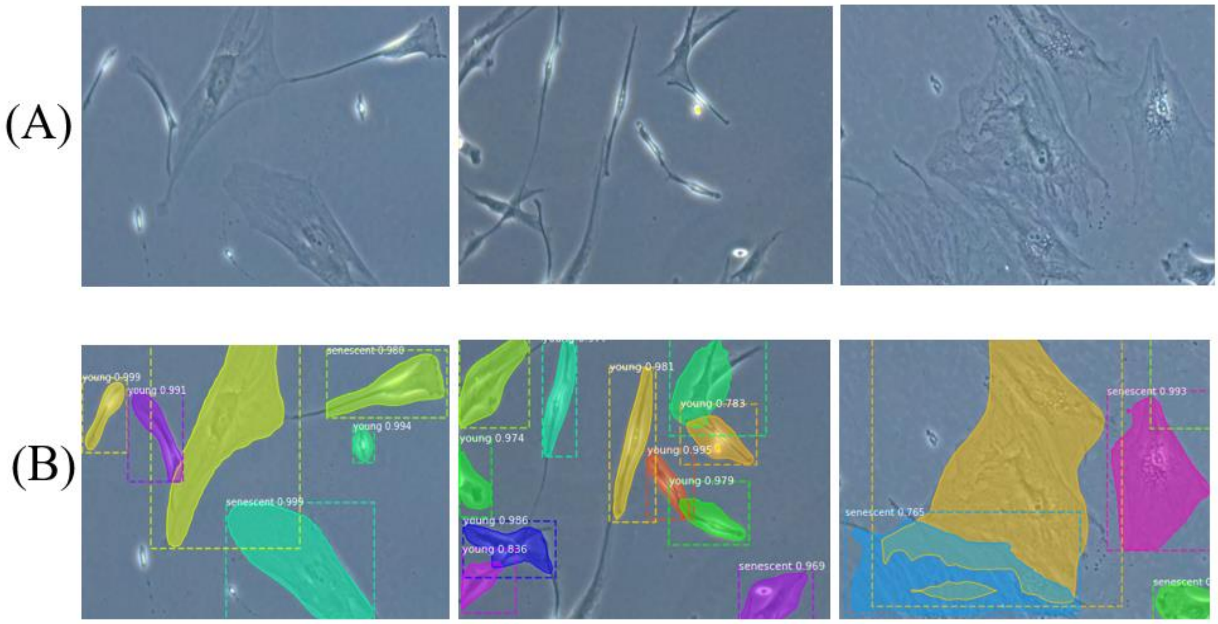


Figure 4.6 Test image samples and their Mask R-CNN without SSL segmentation results. Row (A) shows test images, and row (B) shows results of instance segmentation with masks, object labels, probability values, and bounding boxes.

4.2.1.4 Implementation Details of Mask R-CNN with Self-Supervised Learning and Fine-Tuning

In this part, implementation details of SSL pre-training on unlabeled images and fine-tuning on labeled images for downstream tasks are discussed. While pre-training the ResNet-101 network, followed by a non-linear projection head, is implemented, the pre-training phase is completed with two scenarios: (1) use unlabeled cell images for pre-training without initial weights; and (2) use unlabeled cell images for pre-training with ImageNet weights as initial weights. SSL pre-training is implemented with learning rate in {0.3, 0.1, 0.001, 0.0001}, batch size in {2, 4, 8, 16, 32, 64, 256}, and epoch in {50, 100, 150, 200, 250}. The result of self-supervised training gives the normalized temperature-scaled cross entropy loss (NT-Xent) [134], which computes the loss for positive pairs of two augmented images and contrasts with the distance to the negative pair. Formally expressed as:

$$l_{i,j} = -log \frac{\exp(sim(z_i,z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k\neq i]} \exp(sim((z_i,z_j)/\tau))} \tag{4.1}$$

where sim is cosine similarity and τ is temperature parameter.

LARS optimizer [144] with larger batch sizes is used to stabilize training. The best pre-training model is selected based on downstream task performance, which is trained on a labeled dataset with fine-tuning from SSL pre-training. After SSL on unlabeled data, ResNet-101 as the backbone network without the non-linear projection head is used for the Mask R-CNN architecture in order to train and test labeled datasets. Images are randomly divided into training, validation, and testing with numbers 50, 20, and 30, respectively. While training, initial parameters are set as follows: minimum confidence 0.9, steps per epoch 100, images per GPU 1, weight decay 0.0001, and learning rate 0.001. The maximum number of epochs was set to 800, and the early stopping technique was used.

4.3 Results

To evaluate Mask R-CNN instance segmentation algorithm performance, mean average precision (mAP) and mean average recall (mAR) values are calculated. Precision is the value of correctly predicted bounding boxes out of all bounding boxes found in the image based on an intersection over union (IoU) threshold, which is 0.5 for our case. While the mean of average precision across all images in the dataset is called mAP, the mean of average recall across all images in the dataset is called mAR. mAP is calculated as follows:

$$\boldsymbol{mAP} = \frac{\mathbf{1}}{\boldsymbol{classes}} \sum_{\boldsymbol{c} \in \boldsymbol{classes}} \frac{\boldsymbol{TP_c}}{\boldsymbol{TP_c}+\boldsymbol{FP_c}} \tag{4.2}$$

$$\boldsymbol{mAR} = \frac{\mathbf{1}}{\boldsymbol{classes}} \sum_{\boldsymbol{c} \in \boldsymbol{classes}} \frac{\boldsymbol{TP_c}}{\boldsymbol{TP_c}+\boldsymbol{FN_c}} \tag{4.3}$$

where TP represents true positives, FP represent false positives, FN represents false negatives and classes represents senescent cells or young cells in this problem.

To measure quantification result of Mask R-CNN precision (PR), recall (RE) and dice similarity coefficient (DSC) metric scores were calculated, and they are described as follows;

$$\boldsymbol{PR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FP}} \tag{4.4}$$

$$\boldsymbol{RE} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FN}} \tag{4.5}$$

$$\boldsymbol{DSC} = \frac{\boldsymbol{2TP}}{\boldsymbol{2TP}+\boldsymbol{FP}+\boldsymbol{FN}} \tag{4.6}$$

In Eqs. (4.4)– (4.6), TP (true positive) represents the correctly quantified young or senescent cells, FP (false positive) represents the invalidly quantified young or senescent cells, and FN (false negative) represents the missed quantified young or senescent cells. The overall result of transfer learning from the MSCOCO dataset and transfer learning from an in-domain dataset by SSL is shown in Table 4.2.

Table 4.2 Evaluation results of Mask R-CNN on our microscopy test images with fine-tuning from MSCOCO dataset and self-supervised learning.

Backbone	**Batch Size of SSL**	**SSL from ImageNet weights**	**SSL end-to-end training from start**	$mAP_{0.5}$	$mAR_{0.5}$	$mAP_{0.5-0.95}$	$mAR_{0.5-0.95}$
ResNet-101 + FPN				0.653	0.785	0.351	0.694
ResNet-101 + FPN	16	✓		0.683	0.790	0.341	0.697
ResNet-101 + FPN	32	✓		0.670	0.784	0.344	0.69
ResNet-101 + FPN	64	✓		0.670	0.792	0.331	0.648
ResNet-101 + FPN	256	✓		0.694	**0.821**	0.354	0.707
ResNet-101 + FPN	256		✓	**0.736**	0.819	**0.393**	**0.735**

Mean average precision and mean average recall values are calculated with IoU thresholds of 0.5 and 0.0-0.95. It can be concluded that SSL improves the performance of the segmentation algorithm. The best result for downstream tasks is achieved by SSL with 256 batch sizes. It means that SSL exploits larger batch sizes since larger batch sizes mean a more diverse augmented version of an image, so the network can learn more domain-specific features. The results of different batch sizes and different approaches in SSL during downstream tasks are given in Figure 4.7. It can be seen that the minimum loss in SSL is achieved by 256 batch sizes and training end-to-end from the start. In addition, even smaller batch size pre-training on unlabeled in-domain datasets gives better performance during fine-tuning compared to fine-tuning from MSCOCO dataset weights.

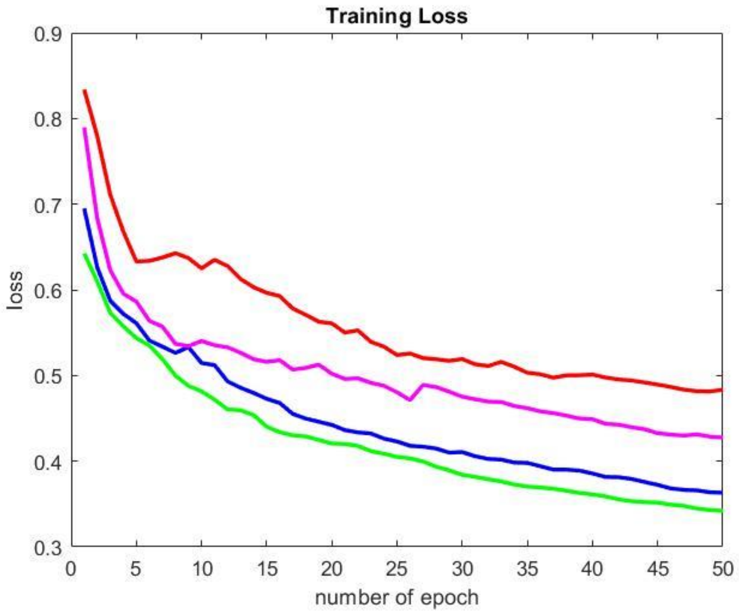
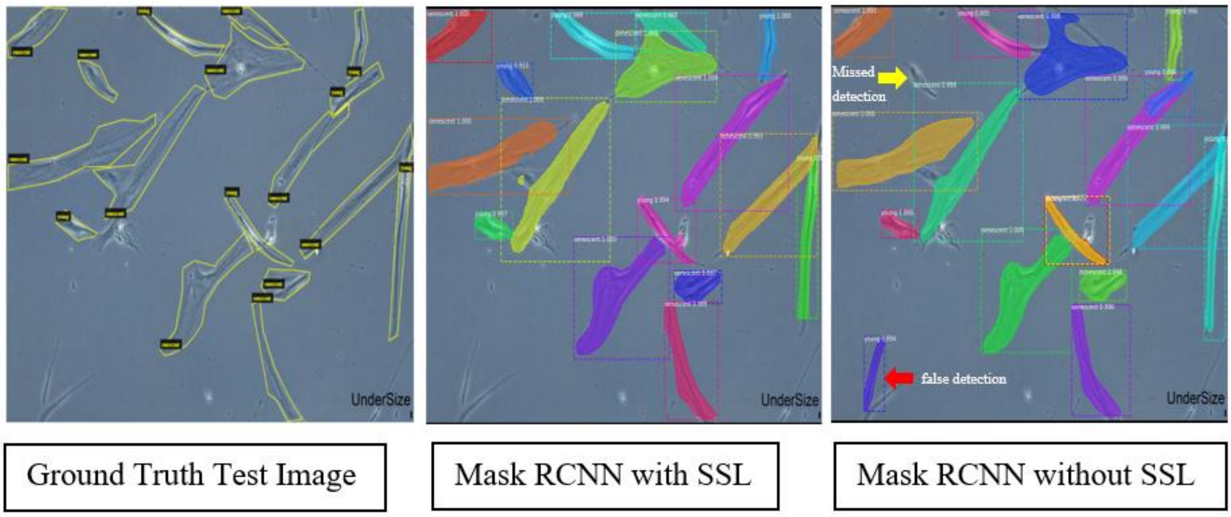


Figure 4.7 Training loss results on labeled dataset. SSL batch 16 indicates ImageNet initialized self-supervised learning with batch size 16 while pre-training, SSL batch 256 indicates ImageNet initialized self-supervised learning with batch size 256 while pre-training, SSL batch 256 end-to-end indicates self-supervised learning without any weights initialization in the proposed method. SSL improves instance segmentation performance.

Besides, from Table 4.2, it can be concluded that self-supervised training end-to-end from the start yields a higher mAP metric for IoU 0.5 and IoU for range 0.5-0.95 than initializing self-supervised training weights from ImageNet. SSL end-to-end from start increased the mAP of Mask R-CNN by %8.3 when IoU is 0.5 and %4.2 when IoU is in the range 0.5-0.95. In addition, SSL end-to-end from start increased the mAR of Mask R-CNN by %3.4 when IoU is 0.5 and %4.1 when IoU is in the range 0.5-0.95. To sum up, it can be seen from Table 4.2 that both SSL approaches increased the evaluation metrics of the original Mask R-CNN.

Self-supervised learning can learn finer details. In the images, there are young cells, senescent cells, and cells that just start aging, which are also taken as senescent cells. While original Mask R-CNN learning just started aging cells as young cells, self-supervised learning can learn those cells as senescent and so segment them correctly. Moreover, Mask R-CNN with self-supervised learning can detect more young cells correctly compared with the original Mask R-CNN. In addition, self-supervised learning can find cell boundaries better when there are overlapping cells. Some samples of instance segmentation using the proposed methods are shown in Figure 4.8. It can be seen that Mask R-CNN is improved by SSL by finding cell classes more accurately and by a low rate of missed detections.



Ground Truth Test Image

Mask RCNN with SSL

Mask RCNN without SSL

Figure 4.8 Test image result with self-supervised learning and without self-supervised learning.

In this study, it is also important to find the number of young and senescent cells to calculate cellular senescence. That’s why the number of detected cells is counted automatically since Mask R-CNN gives the number of each instance. In Table 4.3, the number of senescent and young cells in ground truth images, in original Mask R-CNN, and in Mask R-CNN with SSL is given. As a result, Mask R-CNN with SSL finds a more correct number of cells compared to conventional Mask R-CNN architecture.

Table 4.3 Comparison of the methods by total number of correctly found cells.

	Ground Truth		**Prediction by Mask R-CNN**		**Prediction by SSL + Mask R-CNN**	
	senescent	young	senescent	young	senescent	young
Total Number of cells	334	339	277	239	304	262
Correct Detection ratio (%)			83%	71%	91%	77%
False Negative Ratio (%)			17%	29%	9%	23%

In addition, performance metrics results after automated counting are shown in Table 4.4. To describe TP, FP, and FN values for performance metrics calculation, the results of algorithms are compared with the results of manual counting by experts. It can be seen from the table that Mask R-CNN with SSL increases the automated counting results by 5.04%, 6.71%, and 5.98% in precision, recall, and dice similarity coefficient, respectively.

Table 4.4 Quantitative results of Performance Assessment Procedures (Mean, [min, max]) for Proposed Approach on microscopy cell images.

	Precision%	Recall %	Dice Similarity Coefficient %
Mask R-CNN	83.36 [64.70, 94.11]	79.96 [59.57, 91.66]	81.10 [63.63, 96.96]
Mask R-CNN + SSL	88.40 [73.33, 100]	86.67 [59.57, 100]	87.08 [73.91, 100]

4.4 Conclusion

The main purpose of this study is to develop an automated detection and quantification of young and senescent cells in cell cultured microscopy images to estimate cellular senescence density while only a limited number of labeled training samples are available. When the labeled samples in the training phase are scarce and the pre-training samples are from a different domain (natural images), the transfer learning methods fall short in microscopic image segmentation. This study showed that incorporating the SSL method increases the performance of the model. Without cell staining, the Mask R-CNN instance segmentation algorithm without SSL achieved 0.65 mean average precision (mAP) and 0.78 mean average recall (mAR) on test images with a 0.5 IoU threshold. Mask R-CNN with SSL pre-training achieved 0.736 mean average precision (mAP) and 0.819 mean average recall (mAR) on test images with a 0.5 IoU threshold and 0.393 mean average precision (mAP) and 0.735 mean average recall (mAR) on test images with an IoU threshold in the range of 0.5-0.95. Results show that even if there are a small number of unlabeled datasets, in-domain data self-supervised learning improves the performance of the instance segmentation algorithm. Considering the promising and widespread use of MSCs and the importance of detecting senescent cells in clinical practice, using the deep-learning technique, which can be easily adapted to GMP, is quite advantageous over time-consuming manual methods.

Chapter 5

Conclusions and Future Prospects

5.1 Conclusions

In this thesis study, three studies have been explained. The center of these three studies is the automated detection and counting of objects in medical images, which are acquired from microscopy. In addition, these studies are proposed to produce real-time analysis and results.

In the first study, automated detection and counting of immunomagnetic beads and leukemia cells in bright-field microscopy images were completed. There were two different objective images, like 40X and 20X. During the detection of objects, classical image processing techniques were implemented to have a fast and effective system. However, cells in some images were not easy to detect with only image processing algorithms. That's why machine learning techniques were implemented to detect cells in 20X objective images, and then, after detection of cells, some image processing techniques were used to conclude the quantification process. While for image processing parts, high-resolution images were used, for machine learning parts, small image patches obtained via image division with overlapping were used. The proposed system was planned to be used to quantify cells on a biochip. Without cell staining, color-based image processing methods provided over 90% precision for 40X images. For 20X images, as a machine learning algorithm, a support vector machine (SVM)-based method was used, and the proposed method provided 94% precision for clear images. The results of 20X objective images showed that the image quality has a great impact on the performance of the algorithm.

In the second study, automated detection and counting of immunomagnetic beads, micropads, and leukemia cells in bright-field microscopy images of the MRD biochip were completed. For this study, the image objective was 20X only. During the study, two phases were completed. The first phase only used classical image processing based algorithms to detect and quantify objects. The second phase was to apply deep learning techniques to detect and count objects. For both studies, a patient dataset was used. However, for the first approach, in addition to the patient dataset, cell culture images were used and tested.

In the first phase of the second study, without cell staining, color, size, and shape-based methods achieved 85% precision, 86% recall, and F-measure values for the patient dataset. For some patient images, micropad size or bead density could have been changed, but the results showed that even if micropad size and bead density changed, the proposed methods could still find cells. Besides, micropad sizes may differ, but proposed methods can find them with high accuracy. Another conclusion is that on the image, if the micropad size is small, cells are clearer to see. The highest precision and recall were 98% for the clear cell culture images. Moreover, the proposed methods detected beads and micropads with 0.02 and 0.01 detection errors, respectively. Results were obtained with a comparison of automated counting and manual counting by experts. Finally, for the patient dataset, statistical test results were conducted to see the relationship between automated and manual counting, and the results concluded that the manual counting of the medical expert and automated counting are statistically in good agreement.

In the second phase of the second study, proposed methods were designed to segment objects in the images. Without cell staining, the U-Net + BN network with optimized hype-parameters achieved 98.7% global accuracy, 86.1% mean IoU, 92.2 mean precision, 92.2 mean recall, and 92.2 mean f-1 score on the patient dataset for segmentation. Then the results of the U-Net architecture were compared with those of DeepLabV3 and FCN deep learning networks. Comparison results indicated that U-Net gave the best performance. In addition, the dataset had an unbalanced dataset problem that dramatically affected the performance of the algorithm. To solve this problem, the network is first trained on a balanced dataset problem, and then weights are transferred to the training of an unbalanced dataset. While finding the results of the proposed methods, an annotated dataset was used. After segmentation, automated quantification of objects was done with morphological operations, and results were compared with manual counting results. In this study, annotation of images was completed on high-resolution images, and then images were divided into small patches to have a regular and certain size for the network. This causes some cluster parts to be seen as beads in the patches due to shape and size, so those parts were segmented as beads, and cluster evaluation metrics are lower compared to other classes. Increasing the performance of cluster segmentation can be achieved by dividing images into overlapping windows.

In the third study, automated detection and counting of senescent and young cells were developed in cell-cultured microscopy images to be able to estimate cellular senescence. As a method, a deep learning-based instance segmentation algorithm was implemented. This time, there were a limited number of labeled datasets. However, deep learning techniques benefit from a large number of well-annotated datasets to achieve good performance. To overcome the small number of labeled datasets, a transfer learning approach was applied. Publicly available transfer learning models were pre-trained with natural images, so they are different than medical images. In addition, in the literature, it has been proven that transfer learning from in-domain data exceeds transfer learning from out-domain data. To be able to use transfer learning while training a small number of labeled data, this time pre-training was completed on medical images without labeling. This approach is called as self-supervised learning (SSL) pre-training. ResNet-101, which is the backbone network of the instance segmentation model Mask R-CNN, was first trained on unlabeled senescent and young cell images, and then the weight of the backbone network was used to initialize weights while training with a labeled small number of datasets. Results show that incorporating self-supervised learning into pre-training increased the performance of the segmentation algorithm. Without cell staining, Mask R-CNN with SSL achieved 0.736 mean average precision, 0.819 mean average recall values with a 0.5 IoU value, and 0.393 mean average precision and 0.735 mean average recall with 0.5-0.95 IoU values. In addition, results showed that Mask R-CNN with SSL gave the best performance for the quantification process. Finally, it can be concluded that deep learning techniques can be easily adapted to good manufacturing products to estimate cellular senescence and overcome time-consuming manual methods.

1.3 Societal Impact and Contribution to Global Sustainability

Cancer is the second-leading cause of death in the United States and is expected to surpass heart diseases in the next five years. Early detection of tumor cells and quantifying the tumor cells during therapy are crucial for a successful treatment. There are different treatments for cancer patients, such as surgery, drug therapy (chemotherapy), transplantation, radiation treatment, and immunotherapy. Among those methods, chemotherapy is a widely used first-line treatment for leukemia, but its results differ from patient to patient. While some patients with chemotherapy achieve remission, some patients can have cancer cells that are resistant to treatment, and these cancer cells (blast cells) can cause a relapse of cancer. This case is known as minimal residual disease (MRD). As an alternative to bench-top instruments, novel microfluidics and lab-on-a-chip systems have been demonstrated to capture, isolate, and count tumor cells. To get and analyze the samples in lab-on-a-chip, devices such as optical microscopes for characterizing and counting tumor cells, phase contrast, and immunofluorescence images have been used. To analyze the images, image processing techniques were implemented. In addition, to be able to estimate cellular senescence on images, image processing techniques were implemented. For both studies, our main purpose was to get information from microscopy images with high accuracy to reduce experts' workloads. According to the UNDP Sustainable Development Goals, our purpose meets the fifth goal, which is related to health among all 17 goals. There are some techniques to detect MRD, like immunological assays and polymerase chain reaction (PCR)-based molecular assays. However, those techniques are expensive, take a long time to produce results, and require an expert to be able to use them, whereas the MRD biochip technique, which has image processing embedded in it in order to analyze images, is easy to use and produces results in a short time without any field expertise. On the other hand, one way to detect cellular senescence is the beta-galactosidase assay (b-Gal), but it has disadvantages like cell confluency, serum depletion, and operator bias. Besides, the protocol of b-Gal to identify senescent cells is not easily adaptable to the GMP pipeline and is time consuming. In addition, senescent cell morphology can be different, so image processing-based detection and counting of senescent cells is effective. This thesis is mainly about image processing-based analysis of microscopy images to estimate both the density of cancer cells and senescence cells to reduce time and cost, increase efficiency, and reduce experts' workloads.

1.4 Future Prospects

We previously introduced ultra-low-cost mobile cell phone microscopy to quantify immunomagnetic bead amounts on micro contact printed lines [145] without using any secondary labels. It is possible to combine machine-learning-based image processing algorithms with cell phone microscopy to analyze immunomagnetic-captured cells in biochips for portable on-site analyses, which is our next goal.

The statistical analysis revealed that deep learning-based cell segmentation and counting can be used as a readout method for the MRD Biochip. The current results are promising, and for future work, we are planning to perform more clinical trials and increase the sample size for testing the whole system. We anticipate that the developments in the microfluidic system and the iterative optimizations will further improve the performance of the deep learning algorithm. We observed that cells from patient samples have more variations in terms of size and appearance under the microscope compared to the cells from the cell cultures. In order to improve the reliability of the system, more patient samples should be tested. In addition, we realized that, when we divide the high-resolution images into smaller sizes, some cluster parts are segmented as beads, and this causes less true positives for clusters and more false positives for beads, and so less precision value for clusters, beads, and overall performance. In order to increase precision, the model structure can be changed to analyze high-resolution images by using image pyramid techniques in future work. Moreover, before the segmentation algorithm, some pre-processing methods can be applied to noisy images to increase the precision value.

BIBLIOGRAPHY

[1] M. Hauwel and T. Matthes, “Minimal residual disease monitoring: The new standard for treatment evaluation of haematological malignancies?,” *Swiss Med. Wkly.*, vol. 144, 2014, doi: 10.4414/smw.2014.13907.

[2] “Leukemia - OrthoInfo - AAOS.” https://orthoinfo.aaos.org/en/diseases--conditions/leukemia/ (accessed May 30, 2023).

[3] M. Hauwel and T. Matthes, “Minimal residual disease monitoring: The new standard for treatment evaluation of haematological malignancies?,” *Swiss Med. Wkly.*, vol. 144, 2014, doi: 10.4414/smw.2014.13907.

[4] B. Luan Khoo *et al.*, “Ultra-fast, label-free isolation of circulating tumor cells from blood using spiral microfluidics,” *Nat. Protoc.*, 2015, doi: 10.1038/nprot.2016.003.

[5] S. Halldorsson, E. Lucumi, R. Gómez-Sjöberg, and R. M. T. Fleming, “Advantages and challenges of microfluidic cell culture in polydimethylsiloxane devices,” *Biosens. Bioelectron.*, vol. 63, pp. 218–231, 2015, doi: 10.1016/j.bios.2014.07.029.

[6] S. Zheng *et al.*, “Membrane microfilter device for selective capture, electrolysis and genomic analysis of human circulating tumor cells,” *J. Chromatogr. A*, vol. 1162, no. 2 SPEC. ISS., pp. 154–161, 2007, doi: 10.1016/j.chroma.2007.05.064.

[7] C. L. Chang *et al.*, “Circulating tumor cell detection using a parallel flow micro-aperture chip system,” *Lab Chip*, vol. 15, no. 7, pp. 1677–1688, 2015, doi: 10.1039/c5lc00100e.

[8] S. Nagrath *et al.*, “Isolation of rare circulating tumour cells in cancer patients by microchip technology,” *Nature*, vol. 450, no. 7173, pp. 1235–1239, Dec. 2007, doi: 10.1038/nature06385.

[9] A. L. Campaña *et al.*, “Enzyme-based electrochemical biosensors for microfluidic platforms to detect pharmaceutical residues in wastewater,” *Biosensors*, vol. 9, no. 1, 2019, doi: 10.3390/bios9010041.

[10] C. L. Chang *et al.*, “Circulating tumor cell detection using a parallel flow micro-aperture chip system,” *Lab Chip*, vol. 15, no. 7, pp. 1677–1688, Mar. 2015, doi: 10.1039/C5LC00100E.

[11] S. R. Editor Michael Neuman, J. Jin, J. V Hajnal, D. Hill, and D. J. Hawkes, “Biomedical Image Analysis,” Dec. 2004, doi: 10.1201/9780203492543.

[12] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.

[13] P. Soille, “Morphological Image Analysis,” *Morphol. Image Anal.*, 2004, doi: 10.1007/978-3-662-05088-0.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1076432

[15] “Digital Image Processing for Medical Applications - Geoff Dougherty - Google Kitaplar.”

[16] C. Frank *et al.*, “Collagen Fibril Diameters in the Healing Adult Rabbit Medial Collateral Ligament,” *http://dx.doi.org/10.3109/03008209209007000*, vol. 27, no. 4, pp. 251–263, 2009, doi: 10.3109/03008209209007000.

[17] I. Castiglioni *et al.*, “AI applications to medical images: From machine learning to deep learning,” *Phys. Medica*, vol. 83, no. November 2020, pp. 9–24, 2021, doi: 10.1016/j.ejmp.2021.02.006.

[18] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, “Machine Learning for Medical Imaging,” *RadioGraphics*, vol. 37, no. 2, pp. 505–515, Mar. 2017, doi: 10.1148/rg.2017160130.

[19] “Model Fit: Underfitting vs. Overfitting - Amazon Machine Learning.”

https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html (accessed Jun. 28, 2022).

[20] O. S. Kareem, A. K. AL-Sulaifanie, D. A. Hasan, and D. M. Ahmed, “Segmenting and Classifiying the Brain Tumor from MRI Medical Images Based on Machine Learning Algorithms: A Review,” *Asian J. Res. Comput. Sci.*, no. July, pp. 50–61, 2021, doi: 10.9734/ajrcos/2021/v10i230239.

[21] J. Carrasquilla and R. G. Melko, “Machine learning phases of matter,” *Nat. Phys.*, vol. 13, no. 5, pp. 431–434, 2017, doi: 10.1038/nphys4035.

[22] B. Ramsundar *et al.*, “Massively Multitask Networks for Drug Discovery,” 2015, Accessed: Jun. 28, 2022. [Online]. Available: https://tripod.

[23] N. Coudray *et al.*, “Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning,” *Nat. Med.*, vol. 24, no. 10, pp. 1559–1567, 2018, doi: 10.1038/s41591-018-0177-5.

[24] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nat. Publ. Gr.*, 2017, doi: 10.1038/nature21056.

[25] R. Poplin *et al.*, “Creating a universal SNP and small indel variant caller with deep neural networks,” *bioRxiv*, pp. 1–24, 2016, [Online]. Available: http://biorxiv.org/content/early/2016/12/19/092890.abstract

[26] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, “Deep learning for cellular image analysis,” *Nat. Methods*, vol. 16, no. 12, pp. 1233–1246, 2019, doi: 10.1038/s41592-019-0403-1.

[27] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Dec. 2014, doi: 10.48550/arxiv.1412.6980.

[28] J. Duchi JDUCHI and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization * Elad Hazan,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[29] “On the importance of initialization and momentum in deep learning.” http://proceedings.mlr.press/v28/sutskever13.html (accessed Jun. 29, 2022).

[30] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How Does Batch Normalization Help Optimization?,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, pp. 2483–2493, May 2018, doi: 10.48550/arxiv.1805.11604.

[31] A. S. Daniel Kermany, M. Goldbaum, W. Cai, M. Anthony Lewis, H. Xia, and K. Zhang Correspondence, “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” *Cell*, vol. 172, 2018, doi: 10.1016/j.cell.2018.02.010.

[32] G. Zhu, B. Jiang, L. Tong, Y. Xie, G. Zaharchuk, and M. Wintermark, “Applications of Deep Learning to Neuro-Imaging Techniques,” *Front. Neurol.*, vol. 10, p. 869, Aug. 2019, doi: 10.3389/fneur.2019.00869.

[33] L. Keren *et al.*, “A Structured Tumor-Immune Microenvironment in Triple Negative Breast Cancer Revealed by Multiplexed Ion Beam Imaging,” *Cell*, vol. 174, no. 6, pp. 1373-1387.e19, Sep. 2018, doi: 10.1016/J.CELL.2018.08.039.

[34] “SimCLR — A Simple Framework For Contrastive Learning Of Visual Representation - AI Summary.” https://www.ai-summary.com/summary-simclr-a-simple-framework-for-contrastive-learning-of-visual-representat/ (accessed Jun. 30, 2022).

[35] C. E. DeSantis *et al.*, “Cancer treatment and survivorship statistics, 2014,” *CA. Cancer J. Clin.*, vol. 64, no. 4, pp. 252–271, 2014, doi: 10.3322/caac.21235.

[36] K. Pantel, R. H. Brakenhoff, and B. Brandt, “Detection, clinical relevance and specific biological properties of disseminating tumour cells,” *Nat. Rev. Cancer*, vol. 8, no. 5, pp. 329–340, 2008, doi: 10.1038/nrc2375.

[37] N. Max and U. Keilholz, “Minimal residual disease in melanoma,” *Semin. Surg. Oncol.*, vol. 20, no. 4, pp. 319–328, 2001, doi: 10.1002/ssu.1050.

[38] S. Nagrath *et al.*, “Isolation of rare circulating tumour cells in cancer patients by

microchip technology," *Nat. 2007 4507173*, vol. 450, no. 7173, pp. 1235–1239, Dec. 2007, doi: 10.1038/nature06385.

[39] S. Zheng *et al.*, "3D microfilter device for viable circulating tumor cell (CTC) enrichment from blood," *Biomed. Microdevices*, vol. 13, no. 1, pp. 203–213, Feb. 2011, doi: 10.1007/S10544-010-9485-3/FIGURES/9.

[40] K. İçöz, Ü. Akar, and E. Ünal, "Microfluidic Chip based direct triple antibody immunoassay for monitoring patient comparative response to leukemia treatment," *Biomed. Microdevices*, vol. 22, no. 3, p. 48, Sep. 2020, doi: 10.1007/s10544-020-00503-6.

[41] F. Xing and L. Yang, "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review," *IEEE Rev. Biomed. Eng.*, vol. 9, pp. 234–263, 2016, doi: 10.1109/RBME.2016.2515127.

[42] G. Mata *et al.*, "Automated Neuron Detection in High-Content Fluorescence Microscopy Images Using Machine Learning," *Neuroinformatics*, vol. 17, no. 2, pp. 253–269, 2019, doi: 10.1007/s12021-018-9399-4.

[43] N. Hazwani, A. Halim, M. Y. Mashor, and R. Hassan, "Automatic Blasts Counting for Acute Leukemia Based on Blood," *Int. J. Res. Rev. Comput. Sci.*, vol. 2, no. 4, pp. 971–976, 2011.

[44] L. Putzu, G. Caocci, and C. Di Ruberto, "Leucocyte classification for leukaemia detection using image processing techniques," *Artif. Intell. Med.*, vol. 62, no. 3, pp. 179–191, 2014, doi: 10.1016/j.artmed.2014.09.002.

[45] A. Georgantzoglou *et al.*, "Automatic cell detection in bright-field microscopy for microbeam irradiation studies," *Phys. Med. Biol.*, vol. 60, no. 16, pp. 6289–6303, 2015, doi: 10.1088/0031-9155/60/16/6289.

[46] F. Buggenthin *et al.*, "An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy," 2013.

[47] X. Long, L. Cleveland, and Y. Lawrence Yao, "Automatic detection of unstained viable cells in bright field images using a support vector machine with an improved training procedure," *Comput. Biol. Med.*, vol. 36, no. 4, pp. 339–362, 2006, doi: 10.1016/j.compbiomed.2004.12.002.

[48] F. Mualla, S. Scholl, B. Sommerfeldt, A. Maier, and J. Hornegger, "Automatic cell detection in bright-field microscope images using sift, random forests, and hierarchical clustering," *IEEE Trans. Med. Imaging*, vol. 32, no. 12, pp. 2274–2286, 2013, doi: 10.1109/TMI.2013.2280380.

[49] F. Uslu, K. Icoz, K. Tasdemir, and B. Yilmaz, "Automated quantification of immunomagnetic beads and leukemia cells from optical microscope images," *Biomed. Signal Process. Control*, 2019, doi: 10.1016/j.bspc.2019.01.002.

[50] J. Xu, K. Mahajan, W. Xue, J. O. Winter, M. Zborowski, and J. J. Chalmers, "Simultaneous, single particle, magnetization and size measurements of micron sized, magnetic particles," *J. Magn. Magn. Mater.*, vol. 324, no. 24, pp. 4189–4199, Dec. 2012, doi: 10.1016/J.JMMM.2012.07.039.

[51] M. Rizon *et al.*, "Object Detection using Circular Hough Transform," *Am. J. Appl. Sci.*, vol. 2, no. 12, pp. 1606–1609, 2005.

[52] R. M. Haralick, I. Dinstein, and K. Shanmugam, "Textural Features for Image Classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973, doi: 10.1109/TSMC.1973.4309314.

[53] O. L. Junior, D. Delgado, V. Gonçalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 432–437, 2009, doi: 10.1109/ITSC.2009.5309700.

[54] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.

[55] L. K. Soh and C. Tsatsoulis, “Texture analysis of sar sea ice imagery using gray level co-occurrence matrices,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 2 I, pp. 780–795, 1999, doi: 10.1109/36.752194.

[56] “Properties of gray-level co-occurrence matrix - MATLAB graycoprops.” https://www.mathworks.com/help/images/ref/graycoprops.html (accessed Jul. 14, 2022).

[57] D. A. Clausi, “An analysis of co-occurrence texture statistics as a function of grey level quantization,” *Can. J. Remote Sens.*, vol. 28, no. 1, pp. 45–62, 2002, doi: 10.5589/m02-004.

[58] R. Akbani, S. Kwek, and N. Japkowicz, “Applying support vector machines to imbalanced datasets,” *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 3201, pp. 39–50, 2004, doi: 10.1007/978-3-540-30115-8_7/COVER/.

[59] L. Shamir, J. D. Delaney, N. Orlov, D. M. Eckley, and I. G. Goldberg, “Pattern recognition software and techniques for biological image analysis,” *PLoS Comput. Biol.*, vol. 6, no. 11, 2010, doi: 10.1371/journal.pcbi.1000974.

[60] A. Sopharak *et al.*, “Machine learning approach to automatic exudate detection in retinal images from diabetic patients,” *J. Mod. Opt.*, vol. 57, no. 2, pp. 124–135, 2010, doi: 10.1080/09500340903118517.

[61] G. Braz Junior, S. V da Rocha, J. D. S de Almeida, A. C. de Paiva, A. C. Silva, and M. Gattass, “Breast cancer detection in mammography using spatial diversity, geostatistics, and concave geometry,” *Multimed Tools Appl*, vol. 78, pp. 13005–13031, 2019, doi: 10.1007/s11042-018-6259-z.

[62] Y. C. Chang, C. Y. Yang, R. L. Sun, Y. F. Cheng, W. C. Kao, and P. C. Yang, “Rapid single cell detection of Staphylococcus aureus by aptamer-conjugated gold nanoparticles,” *Sci. Rep.*, vol. 3, pp. 1–7, 2013, doi: 10.1038/srep01863.

[63] J. Lojk, U. Čibej, D. Karlaš, L. Šajn, and M. Pavlin, “Comparison of two automatic cell-counting solutions for fluorescent microscopic images,” *J. Microsc.*, vol. 260, no. 1, pp. 107–116, 2015, doi: 10.1111/jmi.12272.

[64] D. A. Van Valen *et al.*, “Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments,” *PLoS Comput. Biol.*, vol. 12, no. 11, pp. 1–24, 2016, doi: 10.1371/journal.pcbi.1005177.

[65] T. E. Witzig *et al.*, “Detection of circulating cytokeratin-positive cells in the blood of breast cancer patients using immunomagnetic enrichment and digital microscopy,” *Clin. Cancer Res.*, vol. 8, no. 5, pp. 1085–1091, 2002.

[66] J. Selinummi *et al.*, “Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images,” *PLoS One*, vol. 4, no. 10, 2009, doi: 10.1371/journal.pone.0007497.

[67] F. Buggenthin *et al.*, “An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy,” *BMC Bioinformatics*, vol. 14, p. 297, 2013, doi: 10.1186/1471-2105-14-297.

[68] N. Ghane, A. Vard, A. Talebi, and P. Nematollahy, “Segmentation of White Blood Cells from Microscopic Images Using a Novel Combination of K-Means Clustering and Modified Watershed Algorithm,” *J. Med. Signals Sens.*, vol. 7, no. 2, pp. 92–101, 2017, doi: 10.4103/2228-7477.205503.

[69] S. Mohapatra, D. Patra, and S. Satpathi, “Image analysis of blood microscopic images for acute leukemia detection,” *2010 Int. Conf. Ind. Electron. Control Robot. IECR 2010*, pp. 215–219, 2010, doi: 10.1109/IECR.2010.5720171.

[70] N. Patel and A. Mishra, “Automated Leukaemia Detection Using Microscopic Images,” *Procedia Comput. Sci.*, vol. 58, pp. 635–642, 2015, doi: 10.1016/j.procs.2015.08.082.

[71] D. Lin, L. Sun, K. A. Toh, J. B. Zhang, and Z. Lin, “Biomedical image classification based on a cascade of an SVM with a reject option and subspace analysis,” *Comput. Biol. Med.*, vol. 96, no. December 2017, pp. 128–140, 2018, doi: 10.1016/j.compbiomed.2018.03.005.

[72] G. Wang *et al.*, “Morphological background detection and illumination normalization of text image with poor lighting,” *PLoS One*, vol. 9, no. 11, pp. 1–22, 2014, doi: 10.1371/journal.pone.0110991.

[73] L. Yang *et al.*, “An adaptive non-local means filter for denoising live-cell images and improving particle detection,” *J. Struct. Biol.*, vol. 172, no. 3, pp. 233–243, 2010, doi: 10.1016/j.jsb.2010.06.019.

[74] Y. Xue, Z. R. Li, C. W. Yap, L. Z. Sun, X. Chen, and Y. Z. Chen, “Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents,” *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 5, pp. 1630–1638, 2004, doi: 10.1021/ci049869h.

[75] C. L. Huang and C. J. Wang, “A GA-based feature selection and parameters optimizationfor support vector machines,” *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006, doi: 10.1016/j.eswa.2005.09.024.

[76] B. Heisele, T. Serre, S. Prentice, and T. Poggio, “Hierarchical classification and feature reduction for fast face detection with support vector machines,” *Pattern Recognit.*, vol. 36, no. 9, pp. 2007–2017, 2003, doi: 10.1016/S0031-3203(03)00062-1.

[77] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines,” *Mach. Learn. 2002 461*, vol. 46, no. 1, pp. 389–422, 2002, doi: 10.1023/A:1012487302797.

[78] B. Gao, H. J. Klumpen, and H. Gurney, “Dose calculation of anticancer drugs,” *Expert Opin. Drug Metab. Toxicol.*, vol. 4, no. 10, pp. 1307–1319, 2008, doi: 10.1517/17425255.4.10.1307.

[79] R. Sánchez, R. Ayala, and J. Martínez-López, “Minimal residual disease monitoring with next- generation sequencing methodologies in hematological malignancies,” *International Journal of Molecular Sciences*. 2019. doi: 10.3390/ijms20112832.

[80] G. a M. Neale *et al.*, “Comparative analysis of flow cytometry and polymerase chain reaction for the detection of minimal residual disease in childhood acute lymphoblastic leukemia.,” *Leukemia*, vol. 18, no. 5, pp. 934–8, 2004, doi: 10.1038/sj.leu.2403348.

[81] S. Böttcher *et al.*, “Minimal residual disease detection in mantle cell lymphoma: Methods and significance of four-color flow cytometry compared to consensus IGH-polymerase chain reaction at initial staging and for follow-up examinations,” *Haematologica*, vol. 93, no. 4, pp. 551–559, 2008, doi: 10.3324/haematol.11267.

[82] J. M. Jackson *et al.*, “Microfluidics for the detection of minimal residual disease in acute myeloid leukemia patients using circulating leukemic cells selected from blood,” *Analyst*, vol. 141, no. 2, pp. 640–651, 2016, doi: 10.1039/c5an01836f.

[83] S. Faley *et al.*, “Microfluidic platform for real-time signaling analysis of multiple single T cells in parallel,” *Lab Chip*, 2008, doi: 10.1039/b719799c.

[84] Y. Alapan, K. Icoz, and U. A. Gurkan, “Micro- and nanodevices integrated with biomolecular probes,” *Biotechnol. Adv.*, vol. 33, no. 8, pp. 1727–1743, 2015, doi: 10.1016/j.biotechadv.2015.09.001.

[85] J. Schindelin *et al.*, “Fiji: An open-source platform for biological-image analysis,” *Nat. Methods*, vol. 9, no. 7, pp. 676–682, 2012, doi: 10.1038/nmeth.2019.

[86] F. De Chaumont *et al.*, “Icy: An open bioimage informatics platform for extended reproducible research,” *Nat. Methods*, vol. 9, no. 7, pp. 690–696, 2012, doi: 10.1038/nmeth.2075.

[87] M. Georg *et al.*, “Development of image analysis software for quantification of viable cells in microchips,” *PLoS One*, vol. 13, no. 3, pp. 1–15, 2018, doi: 10.1371/journal.pone.0193605.

[88] J. C. Caicedo *et al.*, “Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images,” *Cytom. Part A*, vol. 95, no. 9, pp. 952–965, 2019, doi: 10.1002/cyto.a.23863.

[89] O. Z. Kraus *et al.*, “Automated analysis of high-content microscopy data with deep

learning," *Mol. Syst. Biol.*, vol. 13, no. 4, p. 924, 2017, doi: 10.15252/msb.20177551.

[90] M. Shahzad, A. I. Umar, M. A. Khan, S. H. Shirazi, Z. Khan, and W. Yousaf, "Robust Method for Semantic Segmentation of Whole-Slide Blood Cell Microscopic Images," *Comput. Math. Methods Med.*, vol. 2020, 2020, doi: 10.1155/2020/4015323.

[91] N. Meng, E. Y. Lam, K. K. Tsia, and H. K. H. So, "Large-Scale Multi-Class Image-Based Cell Classification with Deep Learning," *IEEE J. Biomed. Heal. Informatics*, vol. 23, no. 5, pp. 2091–2098, 2019, doi: 10.1109/JBHI.2018.2878878.

[92] B. N. Narayanan, R. A. Ali, and R. C. Hardie, "Performance analysis of machine learning and deep learning architectures for malaria detection on cell images," no. September 2019, p. 29, 2019, doi: 10.1117/12.2524681.

[93] T. Falk *et al.*, "U-Net: deep learning for cell counting, detection, and morphometry," *Nat. Methods*, vol. 16, no. 1, pp. 67–70, 2019, doi: 10.1038/s41592-018-0261-2.

[94] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," *Lect. Notes Comput. Vis. Biomech.*, vol. 26, pp. 323–350, 2018, doi: 10.1007/978-3-319-65981-7_12.

[95] M. Loey, M. Naman, and H. Zayed, "Deep transfer learning in diagnosing leukemia in blood cells," *Computers*, vol. 9, no. 2, 2020, doi: 10.3390/computers9020029.

[96] K. K. Anilkumar, V. J. Manoj, and T. M. Sagi, "Automated Detection of B Cell and T Cell Acute Lymphoblastic Leukaemia Using Deep Learning," *Irbm*, vol. 1, pp. 1–9, 2021, doi: 10.1016/j.irbm.2021.05.005.

[97] F. Uslu, K. Icoz, K. Tasdemir, R. S. Doğan, and B. Yilmaz, "Image-analysis based readout method for biochip: Automated quantification of immunomagnetic beads, micropads and patient leukemia cell," *Micron*, vol. 133, no. January, 2020, doi: 10.1016/j.micron.2020.102863.

[98] P. E. Danielsson, "Euclidean distance mapping," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 227–248, 1980, doi: 10.1016/0146-664X(80)90054-4.

[99] F. Çelebi, K. Tasdemir, and K. Icoz, "Deep learning based semantic segmentation and quantification for MRD biochip images," *Biomed. Signal Process. Control*, vol. 77, no. April, p. 103783, 2022, doi: 10.1016/j.bspc.2022.103783.

[100] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, vol. 37, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[101] C. Chen *et al.*, "Improving the Generalizability of Convolutional Neural Network-Based Segmentation on CMR Images," *Front. Cardiovasc. Med.*, vol. 7, no. June, pp. 1–17, 2020, doi: 10.3389/fcvm.2020.00105.

[102] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 234–241.

[103] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017, doi: 10.1016/j.eswa.2016.12.035.

[104] N. Praljak *et al.*, "Integrating deep learning with microfluidics for biophysical classification of sickle red blood cells adhered to laminin," *PLoS Comput. Biol.*, vol. 17, no. 11, pp. 1–24, 2021, doi: 10.1371/journal.pcbi.1008946.

[105] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 2951–2959, 2012.

[106] L. Liu, F. X. Wu, Y. P. Wang, and J. Wang, "Multi-receptive-field CNN for semantic segmentation of medical images," *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 11, pp. 3215–3225, 2020, doi: 10.1109/JBHI.2020.3016306.

[107] P. J. Chiang, S. M. Wu, M. J. Tseng, and P. J. Huang, "Automated Bright Field Segmentation of Cells and Vacuoles Using Image Processing Technique," *Cytom. Part*

A, vol. 93, no. 10, pp. 1004–1018, 2018, doi: 10.1002/cyto.a.23595.

[108] R. Kumar, R. Srivastava, and S. Srivastava, "Detection and Classification of Cancer from Microscopic Biopsy Images Using Clinically Significant and Biologically Interpretable Features," *J. Med. Eng.*, vol. 2015, pp. 1–14, 2015, doi: 10.1155/2015/457906.

[109] C. Nagant, D. Casula, A. Janssens, V. T. P. Nguyen, and B. Cantinieaux, "Easy discrimination of hematogones from lymphoblasts in B-cell progenitor acute lymphoblastic leukemia patients using CD81/CD58 expression ratio," *Int. J. Lab. Hematol.*, vol. 40, no. 6, pp. 734–739, 2018, doi: 10.1111/ijlh.12912.

[110] E. Kasuya, "Mann-Whitney U test when variances are unequal," *Anim. Behav.*, vol. 61, no. 6, pp. 1247–1249, 2001, doi: 10.1006/anbe.2001.1691.

[111] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10072 LNCS, pp. 234–244, 2016, doi: 10.1007/978-3-319-50835-1_22.

[112] G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation?," *BMVC 2013 - Electron. Proc. Br. Mach. Vis. Conf. 2013*, 2013, doi: 10.5244/C.27.32.

[113] C. Ounkomol, S. Seshamani, M. M. Maleckar, F. Collman, and G. R. Johnson, "Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy," *Nat. Methods*, vol. 15, no. 11, pp. 917–920, 2018, doi: 10.1038/s41592-018-0111-2.

[114] W. Z. Zhuang *et al.*, "Mesenchymal stem/stromal cell-based therapy: mechanism, systemic safety and biodistribution for precision clinical applications," *J. Biomed. Sci.*, vol. 28, no. 1, p. 28, Dec. 2021, doi: 10.1186/S12929-021-00725-7.

[115] X. Wei, X. Yang, Z. P. Han, F. F. Qu, L. Shao, and Y. F. Shi, "Mesenchymal stem cells: A new trend for cell therapy," *Acta Pharmacologica Sinica*, vol. 34, no. 6. pp. 747–754, Jun. 2013. doi: 10.1038/aps.2013.50.

[116] C. D. Wiley and J. Campisi, "The metabolic roots of senescence: mechanisms and opportunities for intervention," *Nat. Metab. 2021 310*, vol. 3, no. 10, pp. 1290–1301, Oct. 2021, doi: 10.1038/s42255-021-00483-8.

[117] N. Basisty *et al.*, "A proteomic atlas of senescence-associated secretomes for aging biomarker development," *PLOS Biol.*, vol. 18, no. 1, p. e3000599, Jan. 2020, doi: 10.1371/JOURNAL.PBIO.3000599.

[118] S. Özcan *et al.*, "Unbiased analysis of senescence associated secretory phenotype (SASP) to identify common components following different genotoxic stresses," *Aging (Albany NY)*, vol. 8, no. 7, p. 1316, Jul. 2016, doi: 10.18632/AGING.100971.

[119] A. Bertolo, M. Baur, J. Guerrero, T. Pötzel, and J. Stoyanov, "Autofluorescence is a Reliable in vitro Marker of Cellular Senescence in Human Mesenchymal Stromal Cells," *Sci. Reports 2019 91*, vol. 9, no. 1, pp. 1–15, Feb. 2019, doi: 10.1038/s41598-019-38546-2.

[120] C. Lechanteur, A. Briquet, V. Bettonville, E. Baudoux, and Y. Beguin, "Msc manufacturing for academic clinical trials: From a clinical-grade to a full gmp-compliant process," *Cells*, vol. 10, no. 6, Jun. 2021, doi: 10.3390/cells10061320.

[121] D. Kusumoto *et al.*, "Anti-senescent drug screening by deep learning-based morphology senescence scoring," *Nat. Commun.*, vol. 12, no. 1, 2021, doi: 10.1038/s41467-020-20213-0.

[122] K. M. A. Muhammad Hammad Saleem, Sapna Khanchi, Johan Potgieter, "Image-Based Plant Disease Identification by Deep," *Plants*, vol. 9, pp. 1–23, 2020.

[123] W. Wang *et al.*, "Learn to segment single cells with deep distance estimator and deep cell detector," *Comput. Biol. Med.*, vol. 108, no. April, pp. 133–141, 2019, doi: 10.1016/j.compbiomed.2019.04.006.

[124] S. K. Sadanandan, P. Ranefall, S. Le Guyader, and C. Wählby, "Automated Training of

Deep Convolutional Neural Networks for Cell Segmentation," *Sci. Rep.*, vol. 7, no. 1, pp. 1–7, 2017, doi: 10.1038/s41598-017-07599-6.

[125] Y. Al-Kofahi, A. Zaltsman, R. Graves, W. Marshall, and M. Rusu, "A deep learning-based algorithm for 2-D cell segmentation in microscopy images," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–11, 2018, doi: 10.1186/s12859-018-2375-z.

[126] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 6, no. 3, pp. 283–292, 2018, doi: 10.1080/21681163.2016.1149104.

[127] I. Heckenbach *et al.*, "Deep Learning Shows Cellular Senescence Is a Barrier to Cancer Development", doi: 10.1101/2021.03.18.435987.

[128] J. W. Johnson, "Adapting Mask-RCNN for Automatic Nucleus Segmentation," pp. 1–7, 2018, doi: 10.1007/978-3-030-17798-0.

[129] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1_48.

[130] S. Zhou, Q. Wang, and N. Kumar, "Mask-RCNN for Cell Instance Segmentation Mask-RCNN for Cell Instance Segmentation," no. April, 2020.

[131] J. Chen and B. Zhang, "Segmentation of Overlapping Cervical Cells with Mask Region Convolutional Neural Network," *Comput. Math. Methods Med.*, vol. 2021, 2021, doi: 10.1155/2021/3890988.

[132] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Comput. Methods Programs Biomed.*, vol. 187, p. 104964, 2020, doi: 10.1016/j.cmpb.2019.06.023.

[133] M. Heker and H. Greenspan, "Joint Liver Lesion Segmentation and Classification via Transfer Learning," no. 1, pp. 1–5, 2020, [Online]. Available: http://arxiv.org/abs/2004.12352

[134] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *37th Int. Conf. Mach. Learn. ICML 2020*, vol. PartF16814, no. Figure 1, pp. 1575–1585, 2020.

[135] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[136] "GitHub - matterport/Mask_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow." https://github.com/matterport/Mask_RCNN (accessed Dec. 10, 2021).

[137] "Keras: the Python deep learning API." https://keras.io/ (accessed Dec. 09, 2021).

[138] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.04467

[139] P. Khosla *et al.*, "Supervised contrastive learning," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, no. NeurIPS, pp. 1–23, 2020.

[140] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9726–9735, 2020, doi: 10.1109/CVPR42600.2020.00975.

[141] S. Azizi *et al.*, "Big Self-Supervised Models Advance Medical Image Classification," no. 1, 2021, [Online]. Available: http://arxiv.org/abs/2101.05224

[142] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," pp. 248–255, 2009, doi: 10.1109/cvprw.2009.5206848.

[143] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Med. Image Anal.*, vol. 58, p. 101539, 2019, doi: 10.1016/j.media.2019.101539.

[144] Y. You, I. Gitman, and B. Ginsburg, "Large Batch Training of Convolutional Networks,"

Aug. 2017, Accessed: Apr. 04, 2022. [Online]. Available: https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet

[145] K. Içöz, “Image processing and cell phone microscopy to analyze the immunomagnetic beads on micro-contact printed gratings,” *Appl. Sci.*, vol. 6, no. 10, 2016, doi: 10.3390/app6100279.

CURRICULUM VITAE

2011 – 2016	B.Sc., Computer Engineering, Melikşah University, Kayseri, TÜRKİYE
2016 – 2023	Ph.D., Electrical and Computer Engineering, Abdullah Gül University, Kayseri, TÜRKİYE
2019 – present	Research Assistant, Computer Engineering, Abdullah Gül University, Kayseri, TÜRKİYE

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) Uslu, F., Icoz, K., Tasdemir, K., & Yilmaz, B. (2019). Automated quantification of immunomagnetic beads and leukemia cells from optical microscope images, published in Biomedical Signal Processing and Control, DOI: 10.1016/j.bspc.2019.01.002

J2) Uslu, F., Icoz, K., Tasdemir, K., Doğan, R. S., & Yilmaz, B. (2020). Image-analysis based readout method for biochip: Automated quantification of immunomagnetic beads, micropads and patient leukemia cell, published in Micron, DOI: 10.1016/j.micron.2020.102863

J3) Çelebi, F., Tasdemir, K., & Icoz, K. (2022). Deep learning based semantic segmentation and quantification for MRD biochip images, published in Biomedical Signal Processing and Control, DOI: 10.1016/j.bspc.2022.103783

C1) Uslu, F., İçöz, K., & Taşdemir, K. (2019, April). Image-Processing Based Signal Readout Method for MRD Biochip. In 2019 27th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.