Deep Learning Based Diagnosis of Acute Lymphoblastic Leukemia (B-ALL) using Image Flow Cytometry

Muhammad Asim Latif

(Registration No: 00000330486)

Department of Engineering

School of Interdisciplinary Engineering & Science (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(September, 2024)

Deep Learning Based Diagnosis of Acute Lymphoblastic Leukemia (B-ALL) using Image Flow Cytometry

By

Muhammad Asim Latif

Registration No: 00000330486

A thesis submitted to the National University of Sciences and Technology, Islamabad, in partial fulfillment of the requirements for the degree of

Master of Science

in

Computational Science & Engineering

Supervisor: **Dr. Mian Ilyas Ahmad**

Co Supervisor: **Dr. Ishrat Jabeen**

School of Interdisciplinary Engineering & Science (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(September, 2024)

Annex A to NUST Letter No.
0972/102/Exams/Thesis-Cert
dated 23 Dec 16.

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr **Muhammad Asim Latif** Registration No. **00000330486** of **SINES** has been vetted by undersigned, found complete in all aspects as per NUST Statutes/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature with stamp: ____________

Name of Supervisor: Dr. Mian Ilyas Ahmad

Date: 20/09/2024

Dr. Mian Ilyas Ahmad
HoD Engineering
Professor
SINES - NUST, Sector H-12
Islamabad

Signature of HoD with stamp: ____________

Date: 20/09/2024

Dr. Mian Ilyas Ahmad
HoD Engineering
Professor
SINES - NUST, Sector H-12
Islamabad

Countersign by

Signature (Dean/Principal): ____________

Date: 20/9/24

Dr. SYED IRTIZA ALI SHAH
Principal & Dean
SINES - NUST, Sector H-12
Islamabad.

AUTHOR'S DECLARATION

I, ***Muhammad Asim Latif*** hereby state that my MS thesis titled "**Deep Learning based Diagnosis of Acute Lymphoblastic Leukemia (B-ALL) using Image Flow Cytometry"** is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature:_____ M Asim Latif

Name:________Muhammad Asim Latif

Date: ____________________19-09-2024

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled **"Deep Learning based Diagnosis of Acute Lymphoblastic Leukemia (B-ALL) using Image Flow Cytometry"** is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and the National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even after the award of MS degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and NUST, Islamabad have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature:____ M Asim Latif

Name:________Muhammad Asim Latif

Date: ___________________19-09-2024

DEDICATION

This thesis is dedicated to *my beloved Parents & Siblings*.

ACKNOWLEDGEMENTS

"Read. Read in the name of thy Lord who created; [He] created the human being from a blood clot. Read in the name of thy Lord who taught by the pen: [He] taught the human being what he did not know."

The importance of education lies in the first Revelation of the Quran and this proved to be my motivation. All praises to **ALLAH**, who bestowed me with the knowledge, patience, health, and ability to complete this thesis.

First and foremost, I would like to sincerely thank my research supervisor Dr. Mian Ilyas Ahmad, and co-supervisor Dr. Ishrat Jabeen for their complete support and encouragement throughout my research, particularly in the dubious phases. Their constructive advice and suggestions have allowed this research to move in the direction it moved and their support has also instilled confidence in me to become a self-dependent researcher. I would also like to acknowledge my GEC members Dr. Muhammad Tariq Saeed and Dr. Absar ul Jabbar for their advice and moral support.

I would like to offer my special thanks to my friends for their untimely discussions and advice which helped me get through the difficult phases of my research. The faculty and management of SINES, NUST proved to be extremely resourceful as well and have helped create a supportive and productive environment within the department.

Last but not the least, my deepest gratitude goes to my beloved parents who are the reason behind all my successes, and also to my siblings for their endless prayers and love.

Contents

Abstract **X**

1 Introduction **1**

1.1 Background 1

1.2 Deep Learning in Medical Diagnostics 2

1.3 Image Flow Cytometry 4

1.4 Problem Statement 6

1.5 Research Objectives 6

1.6 Overview 7

2 Literature Review **8**

2.1 Acute Lymphoblastic Leukemia (ALL) 8

2.2 Traditional Methods of diagnosis of B-ALL 12

2.2.1 Cytogenetic Analysis 12

2.2.2 Immunophenotyping 13

2.2.3 Integration of Prognostic Factors 14

2.3 Image Flow Cytometry in Hematology 14

2.3.1 Applications in Hematology 15

2.3.2 Image Analysis in Hematology 16
2.3.3 B-Cell Acute Lymphoblastic Leukemia (B-ALL) 16
2.4 Deep Learning Techniques in Medical Imaging 17
2.4.1 Principles of Deep Learning 17
2.4.2 Most widely used models in Medical Imaging 20
2.4.3 Case Studies in Deep Learning for Medical Imaging 25
2.5 Methodologies of Integration of Deep Learning with IFC 26
2.5.1 Image Processing 27
2.5.2 Deep Learning Archietecture 29
2.5.3 Training and Validation 31
2.5.4 Applications in Hematology 34
2.5.5 Key Studies 37
2.5.6 Challenges 38
2.6 Overview 40

3 Methodology **42**

3.1 Data Collection 42
3.1.1 Ethical Considerations 43
3.1.2 Image Flow Cytometry 43
3.1.3 Data Quality and Storage 44
3.1.4 Data Collection Challenges 45
3.2 Data Preprocessing 46
3.2.1 Data Cleaning 46
3.2.2 Data Normalization 47

3.2.3 Data Augmentation 48

3.2.4 Feature Extraction 48

3.2.5 Data Splitting 49

3.3 Proposed Pipeline 49

3.3.1 Detection of Viable and Non-Viable Images 50

3.3.2 Cell Detection 52

3.3.3 Cell Classification 53

3.3.4 White Blood Cell Classification 55

3.3.5 Healthy & Non-Healthy Lymphocytes Classification 58

3.3.6 Classification of ALL 61

4 Results and Discussion 63

4.1 Results 63

4.1.1 Data 63

4.1.2 Viable & Non-Viable Detection 64

4.1.3 Cell Detection 66

4.1.4 Cell Classification 69

4.1.5 WBC Classification 71

4.1.6 Healthy & Non-Healthy Lymphocytes 72

4.1.7 Classification of ALL 75

4.1.8 Gradio Interface 77

4.2 Summary of Results 78

4.3 Comparison of Results 78

5 Conclusion **80**

5.1 Conclusion 80

5.2 Future Aspects 81

References **82**

A Code **93**

List of Tables

2.1 Summary of studies on deep learning approaches for ALL diagnostics 38

4.1 Performance metrics for Viable and Non-Viable cases 64

4.2 Performance metrics for Red blood cells, White blood cells, and Platelets . . 69

4.3 Training, Validation, and Test Metrics of Vision Transformer model. 71

4.4 Training and Validation Metrics of ResNet101 demostrating a strong learning curve with positive results. 73

4.5 Performance metrics for Malignant and Non-malignant cases. 73

4.6 Training and Validation Metrics of DenseNet121 75

4.7 Key steps of the pipeline, models, and their accuracies. 78

4.8 Comparison of Accuracies with Our Proposed Pipeline 79

List of Figures

2.1 Basic CNN Architecture [48] 21

2.2 The UNet Architecture [11] 21

2.3 The ResNet Architecture [69] 22

2.4 The ViT base model Architecture [51] 23

2.5 The DenseNet121 Architecture [70] 24

2.6 The YOLOv5 Architecture [78] 25

3.1 LabelImg software is used for annotating cells on blood smear images. 45

3.2 The Overview of Proposed Pipeline 50

3.3 The ResNet101 Architecture [40] 59

4.1 Precision Recall curve of YOLOv5 66

4.2 F1 Confidence curve of YOLOv5 67

4.3 Results summary of YOLOv5 67

4.4 Loss-Accuracy Plot shows the accuracy and loss of the model during training and validation 70

4.5 Performance metric plots showing the Precision, Recall and F1 plot of UNet-ResNet hybrid model. 70

4.6 Loss curve of ResNet101 showing an overall decresing indent demonstrates a positive loss in subsequent epochs. 74

4.7 Training and Validation plot of DenseNet121 demonstrating increase in accuracy showing the positive learning of model. 76

List of Abbreviations

ALL	Acute Lymphoblastic Leukemia
B-ALL	B-cell Acute Lymphoblastic Leukemia
WBC	White Blood Cells
RBC	Red Blood Cells
CNN	Convolutional Neural Network
IFC	Imaging Flow Cytometry
UNet	U-Net (Convolutional Neural Network for Biomedical Image Segmentation)
CRF	Conditional Random Field
YOLO	You Only Look Once (Real-Time Object Detection)
ViT	Vision Transformer
MLL	Mixed Lineage Leukemia
MRD	Minimal Residual Disease
HSCT	Hematopoietic Stem Cell Transplant

Abstract

Acute Lymphoblastic Leukemia (ALL) is among the world's most fatal diseases which results in the death of thousands of people suffering from ALL annually. Most of the time, procedural delay and delays in diagnosing ALL have a higher chance of an increase in suffering and pain and sometimes, even death can occur. A rapid and effective system of diagnosing ALL is used in this study to help doctors in the early and quick management of patients. The pipeline is composed of different steps and each step uses a deep learning model to classify the blood smears taken from Image Flow Cytometry (IFC). A high-definition image taken from IFC has more details as compared to normally available microscopic images and hence the detection of change in cells can be monitored efficiently. After learning from high-definition images, models like ResNet, a combination of U-Net and ResNet, Vision Transformer (ViT), YOLOv5, and DenseNet have shown exceptional accuracies over a dataset of 3242 images with an overall accuracy of 96.8% is shown during the test run of the pipeline. It also takes a quick runtime ranging from 9 seconds to 50 seconds depending upon the information a cell has to provide. Achieved results show the efficiency of the pipeline as compared to the previous studies done by using either a single algorithm or two algorithms. Furthermore, this pipeline can be trained for other blood diseases as well as for the genomic and phenomic data of cells as It will aid in complementing the results generated by the pipeline.

Keywords: deep learning, vision transformers, Acute Lymphoblastic Leukemia (ALL), Imaging Flow Cytometry, pipeline

Chapter 1

Introduction

1.1 Background

Acute Lymphoblastic Leukemia (ALL) is one of the cancer types affecting human body and it involves the uncontrolled growth of immature lymphoid cells in the bone marrow and peripheral circulation. This malignancy inhibits the formation of normal blood cells and therefore there is reduced production of these cells, hence the manifestations of anemia, frequent incidences of infections, and increased bleeding. There are many subtypes as ALL and B-ALL; B-ALL is the most prevalent, additionally accounting for a large percent of leukemia cases, specifically in children. Primary, B-cell ALL is one of the commonly found neoplastic diseases in kids and adolescents; thus, it is of great significance in pediatric hematology/oncology [66],[80].

Over the years, the diagnosis for B-ALL has enhanced dramatically due to enhanced treatment protocol as well as early diagnosis techniques. diagnosis is considerably improved by the use of intensive chemotherapy regimens, targeted anthracycline, and mini–nonanthracycline–containing chemotherapy regimens, as well as by the application of hematopoietic stem cell transplantation; pediatric patients have better outcomes. The success of B-ALL management depends on early diagnosis to facilitate early intervention, hence the achievement of favorable results. However, the rates of survival still remain less consistent, and depend on the age, genetic attributes differences between patients, and regional differences. It reveals the necessity of subsequent studies in refining diagnostics and diagnosis methods and providing equal outcomes for all potential patients [66],[80].

Although these procedures are effective, they are bound to be time-consuming more specifically they are manual intensive and need a lot of specialization. Sample preparation, data collection, and analysis are time-consuming and can be costly processes, which results in a delay in diagnosis and treatment. They have underlined the necessity of better diagnostic methods, which can offer quick, accurate, and self-sufficient analyses to optimize the outcomes of a patient's management [66],[80].

Convolutional neural networks (CNNs),in particular, are deep learning algorithms that have shown astonishing results in recent years, particularly when it comes to learning and detecting intricate patterns in vast datasets. The accuracy and reliability of these algorithms have been demonstrated to be excellent, and they frequently outperform human specialists in terms of performance. In the realm of medical imaging, Convolutional neural networks (CNNs) have been effectively applied to a variety of applications. These techniques include cytometry, histological analysis, and radiological imaging. As a result of their capacity to deliver objective and reproducible analysis, the amount of time necessary for diagnostics has been greatly cut down, which has increased both efficiency and accuracy [41],[44].

1.2 Deep Learning in Medical Diagnostics

Machine learning is a branch of artificial intelligence that works with algorithms capable of learning from and making predictions on datasets deeply embedded with complex patterns, and deep learning is a branch of machine learning that uses artificial neural networks to model and analyze these complex patterns. This has applied to several fields, especially the medical one as Deep learning algorithms have time and again proved to outcompete physicians in image recognition exercises. When such networks are trained on huge amounts of data, deep learning models can differentiate between images based on features of higher complexity and, therefore, improve the reliability of the diagnostic results [14],[55].

It is now clear that deep learning applications in medical imaging are far and wide across the field ranging from radiology, histopathology, and cytometry among others. In radiology, deep learning algorithms utilize information achieved by X-ray, CT scan, and MRI and perceive and diagnose growth, for example, tumors, cracks, or other pathological changes with stunning accuracy. Histopathology expertise is provided by deep learning as the microscopic examination of tissues is enhanced by deep learning-assisted histopathological analysis of tissue samples to detect malignant cells and other histopathological alterations. In the same way, in cytometry, deep learning is applied to improve the investigation of cell properties with regard to the manifestation of various disease states [14],[55].

More precisely, the application of deep learning with regard to the diagnosis of B-cell Acute Lymphoblastic Leukemia (B-ALL) by applying Image Flow Cytometry also caught academics' attention. Image flow cytometry is a factor of higher dimensionality capturing brightfield and/or fluorescence images of suspended cells. The analysis of this data has been done traditionally and it is usually associated with many shortcomings such as being time-consuming and involving a lot of manual inputs. However, if those high-dimensional databases are to be processed, it often requires lots of pre-processing but deep learning can efficiently pick up features out of these big databases. It minimizes the misdiagnosis and variability in observation by different people and thus provides a more accurate and consistent chance of diagnosis [41],[44].

Deep learning with image flow cytometry continues with the enhancement of new algorithms that are fit for processing data from this modern technology. These algorithms are normally recurrent with convolutional neural networks (CNNs) since the information they handle is mostly in the form of images. It's possible to achieve promising prognostication by training CNNs to distinguish particular features or characteristics from labeled datasets of image flow cytometry data derived from B-ALL cases. This automated analysis not only accelerates the diagnosis but also increases the accuracy of the diagnostic process and offers clinicians a better understanding of the specific disease [41],[44].

Another advantage of using deep learning in the diagnosis of B-ALL is that it limits inter-observer variation. Thus, the processes of diagnosis in traditional diagnostic methods depend on the clinician's interpretation of the data obtained, which in turn contributes to inter-clinician variability in the model. It also removes variability and a situation where some patients have their data analyzed to a higher standard than others is eradicated since deep learning algorithms work on structures and apply them to all forms of data. However, deep learning is an automatic method of training and classification; it can thus handle a significant quantity of examples in a short period, which is vital in environments like clinics since the speed of diagnosis greatly affects the patients' outcomes [41],[44].

1.3 Image Flow Cytometry

Image flow cytometry is a highly developed amalgamation of the quantitative characteristics of flow cytometry along with highly descriptive microscopy methods. It can get bright-field and/or fluorescence images of suspended cells at very high magnification. Conventional flow cytometry has high sensitivity to quantify the cellular entities and to detect a certain type of cells employing fluorescent tags. However, it is usually not capable of giving specific information regarding the manners in which morphological characteristics are exhibited. On the other hand, the microscopy technique lacks quantitative data but it provides detailed images. Notification of these gaps is made by image flow cytometry since it integrates the strengths of both approaches for a versatile tool in cellular analysis [3],[18].

In the case of hematological malignancies, like B-ALL, image flow cytometry has most of its applications since it provides an integrated approach to cancer diagnostics. However, these diseases' minor structural changes and the virtue of unique receptors are vital when differentiating the diseases and even in diagnosis. The combination of qualitative and quantitative paternal analysis allows for greater definition in the study of the cells' properties. For example, in the case of image flow cytometry, two populations of cells that have different sizes, shapes and fluorescence intensities are easily discriminated and relate to different disease states, malignant or benign. This dual functionality increases diagnostic accuracy and examines disease etiology in greater detail [3],[18].

Applying deep learning to image flow cytometry calls for the designing of algorithms to be able to feed and analyze large data produced by this technology. These algorithms, which usually rely on CNN, are employed to motivate multidimensional data and recognize various patterns that possibly may not be detectable by the human eye. Such models are trained by inputting large sets of annotations of images that facilitate the algorithms to identify characteristics related to various prognostic markers. For instance, one can train deep learning models to identify different stages of disease severity or to find biomarkers that correspond to patients' diagnosis [41],[79].

Thus, deep learning in this context has several following important benefits. Firstly, the use of the software increases the reliability of data analysis by minimizing errors and observers' variation. Unlike humans it does not have preconceptions and the same criteria that apply to the training dataset will be used on the test datasets. Secondly, these models can handle large amounts of data, and take less time than conventional approaches to come up with results. This speed is important particularly within clinical environments because time is of the essence in matters concerning diagnosis and treatment or even the diagnosis of the patient's condition [79].

Additionally, the future combination of deep learning with the method of image flow cytometry can open great opportunities in the field of individualized medicine. It is only logical that through the stratification of patients with their respective cellular characteristics, treatment can now be offered more specifically to patients. For example, patients with higher risk will be detected at the initial stage of the disease and recommended for a higher-risk treatment plan unlike patients with lower risk will be recommended for low-risk skills all in order to reduce side effects. The idea is to provide the patient with a more individualized treatment that can enhance the therapeutic outcomes and increase the survival rates of B-ALL patients [79].

1.4 Problem Statement

Acute Lymphoblastic Leukemia (ALL) is the second most common acute leukemia in grown-ups, with an occurrence of 6500 cases each year in the US alone. An automated framework for the diagnosis of Acute Lymphoblastic Leukemia (ALL) based on deep learning models is therefore essential that can take images of blood samples and classify the disease, accurately and precisely within seconds. It will aid doctors in the diagnosis of ALL without waiting for detailed test descriptions of different flow cytometry. This quick way of diagnostics of ALL will save time as kids need urgent medication to stop the spread of malignancy. It will also describe how future intelligent flow cytometry may be implemented using upcoming machine learning methods for more diseases of the blood.

1.5 Research Objectives

The following objectives have been designed for conducting the proposed research work:

- Collection and pre-processing of image-based data, including labeling and transformation.
- Development of a deep learning-based framework for the diagnosis of Acute Lymphatic Leukemia (ALL)
- Comparison of this developed framework with existing deep learning models.
- Web/Software-based Application for the proposed framework/pipeline.

1.6 Overview

After the discussion of the significance of the diagnosis for Acute Lymphoblastic Leukemia (ALL) and the role of deep learning for precise diagnosis in the previous sections of this thesis, this stage of the work will provide a review of the existing literature in Chapter 2. This chapter will elaborate on the current approaches in the diagnosis of adult lymphoblastic leukemia and its consequences: manifestations of image flow cytometry and the existing traditional ones, their development steps, and possible problems. In doing so, it paves the way for the methodological proposal detailed in Chapter 3 which presents the data acquisition tools coupled with preprocessing techniques, as well as the proposed deep learning-based framework for ALL diagnosis. Lastly, Chapter 4 describes the evaluations of the framework and employs the results to discuss the relative merits of our approach to previous approaches, followed by a brief explanation of its practical use and further development in Chap. 5.

Chapter 2

Literature Review

In this chapter, we briefly discuss the ALL disease and the contributions in literature related to its diagnosis. The literature includes different traditional diagnostic tools like cytogenetic analysis, immunophenotyping, and molecular testing. The application of flow cytometry in diagnostics and the use of deep learning in the field of diagnostics, using images, are also discussed.

2.1 Acute Lymphoblastic Leukemia (ALL)

Acute lymphoblastic leukemia, ALL, is a type of hematological cancer that originates from the precursor lymphoid cells in the bone marrow. This disease is a significant public health concern all over the world, particularly among children and young adults. According to [54], it is among the most prevalent types of leukemia in kids and adolescents, accounting for around 25 % of all child cancer cells. The heterogeneity of ALL, the variety of professional discourses, and the unpredictable therapy reactions continue to present challenges, despite the tremendous advancements that have been made in the treatment modalities.

ALL is caused by the malignant lymphoid progenitor cells, which results in the uncontrolled spread of the disease as well as the infiltration of disease-causing cells into the bone marrow, the bloodstream, and the extra-medullary sites. According to [31], these lethal outbreaks are unable to differentiate into fully formed lymphocytes, which fails bone marrow and compromised immune system function in the body. Exhaustion, high temperature, dullness, simple wounding, petechial, and bone ache are some of the signs and symptoms that may be associated with ALL. The professional symptoms of ALL vary greatly and may include a variety of indications and complications. Central nerve system (CNS) participation indicated by signs and symptoms such as migraines, throwing up, plus cranial nerve friends, is additionally typical in ALL plus stands for a major danger aspect for negative results [49].

The formation of premature B-cell blasts is the defining characteristic of B-ALL, characterized by the generation of precursor B-cells in the bone marrow that is responsible for the disease's development. The genetic landscape of B-ALL is complex, with recurrent chromosomal abnormalities and molecular alterations contributing to the etiology of the disorder as well as professional outcomes [59]. Typical hereditary abnormalities in B-ALL consist of chromosomal translocations of immunoglobulin hefty chain (IGH) locus such as t(12; 21) ETV6-RUNX1 or t(9; 22) BCR-ABL1 in addition to rearrangements of mixed-lineage leukemia (MLL) genetics [59]. The disruption of critical signaling pathways linked with B-cell growth and differentiation also drives leukemogenesis and the development of disease. These genetic changes are responsible for the dysregulation of these pathways.

Outer blood and bone marrow examinations, in addition to immunophenotypic analysis, are the components that are used in the medical identification of all-encompassing lymphoma (ALL). The presence of increased outbursts, which may indicate specific roles such as a high nuclear-cytoplasmic proportion, limited cytoplasm, and penalty chromatin pattern, is revealed by morphological study of blood and bone marrow patches [46]. According to [46], immunophenotyping, which can be performed using circulation cytometry or immunohistochemistry, is a method that identifies leukemia cells based on their surface area antigen expression accounts. The majority of ALL cases are of the B-cell family tree, while the T-cell family tree remains the dominant family tree. According to [59], cytogenetic and molecular studies continue to identify ALL below types based on inherited disorders such as chromosomal translocations, removals, and anomalies. These chromosomal abnormalities have a significant impact on diagnosis and also encompass a range of treatment options.

The treatment options available for ALL have undergone tremendous advancements over the last few years, which led to major improvements in survival rates of pediatric patients in particular. Contemporary treatment practices typically involve complex multi-agent radiation treatment programs that are administered in phases. These stages include induction, combination, and maintenance treatment [54]. Treatment of the central nervous system (CNS) is an essential component of all-embracing lung cancer (ALL) treatment, which can be accomplished using intrathecal radiation therapy and/or cranial irradiation in order to prevent relapse of the CNS [49]. High-risk people, such as those with persistent conditions or relapsed ALL, may be advised to consider undergoing an allogeneic hematopoietic stem cell transplant (HSCT). This procedure offers the potential for long-term disease control as well as treatment [10].

Regardless of the heterogeneity of hereditary irregularities that have been observed in B-ALL targeted treatments, such as tyrosine kinase inhibitors (TKIs) and monoclonal antibodies versus B-cell surface area pens, there has been demonstrated assurance in improving therapy outcomes, particularly in individuals who have high-risk conditions or B-cell acute lymphoblastic leukemia that has relapsed or is resistant to treatment [19]. According to Shanon and Theodore [43], the introduction of chimeric antigen receptor (AUTOMOBILE) T-cell treatment has transformed the situation of therapy for relapsed/refractory B-ALL, that targets CD19-expressing B-cell malignancies. This treatment has been observed to have high rates of complete remission as well as long-lasting effects in professional tests. Nonetheless, obstacles continue to be in recognizing efficient targeted treatments for particular hereditary subtypes of B-ALL together with handling treatment-related toxicities highlighting the requirement for additional research study plus development for B-cell Acute Lymphoblastic Leukemia.

Despite the significant progress that has been made in ALL therapy, there are still limitations that exist, particularly in certain individual subgroups that have high-risk condition functions or resistance to therapy. Relapsed or refractory ALL is a significant professional challenge caused by insufficient outcomes and a limited range of treatment options [19]. Furthermore, the long-term adverse consequences of prolonged radiation treatment programs, which include cardiotoxicity, neurotoxicity, infertility, and new hatreds, pose significant challenges for survivors of pediatric acute lymphoblastic leukemia [30]. Because of this, there is an immediate need for innovative restorative procedures in addition to tailored therapy techniques in order to improve outcomes and bring down the amount of toxic drugs that are associated with treatment in ALL.

Recent advances in genomic profiling and molecular characterization have offered valuable information about the etiology of ALL and identified specific therapeutic targets. These advancements have also led to the identification of targeted therapeutic interventions. By recognizing recurrent hereditary changes, such as chromosomal translocations involving the BCR-ABL1 combination genetics and also abnormalities of the MLL genetics, targeted treatments consisting of tyrosine kinase inhibitors (TKIs) and epigenetic modifiers have been developed [59]. These treatments have been extremely effective in treating multiple myeloma. With high feedback rates and also resilient compensations observed in professional tests, immunotherapy has become a promising therapy modality for relapsed/refractory ALL [43]. Specifically, chimeric antigen receptor (CAR) T-cell treatment, that targets CD19-expressing B-cell malignancies, has emerged as a promising target for immunotherapy. According to Sabina and Robin [6], preclinical and clinical studies are currently examining unique agents that target critical signaling pathways and immunological checkpoints. These agents are providing new options for accurate medication and tailored treatment of acute lymphoblastic leukemia (ALL).

ALL continues to be a significant medical challenge, even though there have been advancements in treatment methods and encouraging treatment activities. In light of the variability of the illness as well as the diverse responses to treatment, there is a pressing need for creative and individualized therapeutic approaches. Recent developments in genomic profiling, targeted therapy, and immunotherapy have provided individuals with ALL new expected boosting results as well as lifestyle improvements. These developments highlight the importance of multidisciplinary collaboration and translational research study in the progression of leukemia treatment.

2.2 Traditional Methods of diagnosis of B-ALL

The B-ALL, B-cell acute lymphoblastic leukemia, is a disease that is characterized by various inherited issues. Because of this, it is necessary to have accurate analysis tools in order to guide treatment decisions and to estimate individual outcomes. In the case of B-ALL, the typical methods of diagnosis involve a combination of professional characteristics, cytogenetic analysis, immunophenotyping, and molecular screening. The purpose of these methods is to categorize individuals into different threat groups and then provide the treatment as per requirement [47]. Although these classic analysis features provide valuable insights into the disease as well as the response to treatment, they have certain limitations when it comes to accurately predicting individual outcomes, particularly in the context of accurate medication and tailored treatment.

2.2.1 Cytogenetic Analysis

Cytogenetic analysis is a crucial component in the process of diagnosing B-ALL. It is responsible for identifying recurrent chromosomal abnormalities, as well as hereditary changes that are associated with the etiology of the disease and professional outcomes [47]. Typical cytogenetic abnormalities in B-ALL include chromosomal translocations such as t(9; 22) BCR-ABL1, t(12; 21) ETV6-RUNX1, and t(1; 19) TCF3-PBX1, in addition to mathematical irregularities, deletions, and amplifications [59]. As a consequence of these genetic changes, critical signaling pathways linked with B-cell proliferation and differentiation are disrupted, which leads to the dysregulated multiplication and survival of leukemic blasts. When it comes to B-ALL, cytogenetic abnormalities play a significant role in determining the eventual outcome, with certain hereditary groupings being associated with either favorable or unfavorable outcomes. As an illustration, the Philadelphia chromosome (t(9; 22) BCR-ABL1) presence is associated to an incorrect diagnosis as well as resistance to conventional radiation therapy. On the other hand, hyperdiploid, in conjunction with the absence of high-risk cytogenetic irregularities, is associated with favorable outcomes [59]. Cytogenetic analysis has limitations when it comes to finding low-frequency subclonal abnormalities and intratumoral heterogeneity, both of which have the potential to influence the response to treatment as well as the progression of both conditions.

2.2.2 Immunophenotyping

According to [46], immunophenotyping, which can be performed by circulation cytometry or immunohistochemistry, is a valuable tool for detecting and also sub-classifying B-ALL based on the expression of lineage-specific antigens as well as uneven surface area pens. The B-cell-associated antigens, such as CD19, CD20, CD22, and CD79a, and specific immunoglobulins, presence is a defining characteristic of B-ALL. Additionally, the absence or diminished expression of T-cell and myeloid antigens is also a characteristic of B-ALL, as stated by [46]. According to [49], irregular antigen expression patterns, such as the deletion of CD10 or CD20, may be indicative of a more aggressive disease genotype, which may also be associated with a more concerning diagnosis. Immunophenotyping is also useful in determining very minor recurring disease (MRD), which is the presence of recurring leukemic cells at reduced levels (below 0.01%) after the completion of treatment. This is a reliable indicator of regression and survival in patients with B-ALL, as stated by [49]. Despite this, immunophenotyping is not without its limitations when it comes to identifying atypical subpopulations of leukemic cells, and additionally, it is susceptible to the quality of the samples.

2.2.3 Integration of Prognostic Factors

In the case of B-ALL, the combination of multiple analytic aspects, which include professional, cytogenetic, immunophenotypic, and molecular specifications, permits even more precise threat categorization in conjunction with individualized treatment methods [47]. Numerous risk stratification systems have been developed to categorize individuals with B-ALL into various risk groups. These risk groups are determined by the presence of high-risk factors, which include age, leukocyte count, cytogenetic abnormalities, and treatment reaction [47]. These threat stratification systems, such as the National Cancer Institute (NCI) standards for St. Jude Total XV and XVI procedures, as well as the European Treatment of Acute Lymphoblastic Leukemia (EORTC) risk rating, assist in evaluating the efficacy and the duration of treatment, maximizing the effectiveness of treatment while simultaneously minimizing the risk of treatment-related poisoning [47]. In addition, the utilization of MRD surveillance throughout and after treatment makes it possible for the early detection of disease regression and rapid treatment, which in turn increases the survival rates of patients with B-ALL [49]. Nonetheless, the perfect incorporation of analysis elements combined with the growth of risk-adapted therapy formulas continues to be the location of dynamic research study as well as expert assessment in B-ALL.

2.3 Image Flow Cytometry in Hematology

Image Flow Cytometry is a technique that combines the quantitative capabilities of flow cytometry and the precise imaging of microscopy. This apparatus is considered one of the most potential apparatuses in hematology. Large populations of cells can simultaneously be analyzed with the capture of detailed images with this apparatus, and not only the numerical data given but also the visual confirmation of the cellular properties is also possible. Our development of knowledge of hematopoietic diseases was improved as a result of its numerous uses in clinical diagnosis and research of many blood diseases. Examples of principles that are used in image flow cytometry include microscopy and flow cytometry, among others. The most commonly used flow cytometry is used in the first moving step, in which the cells are suspended in a fluid and are passed individually through a laser beam. However, an image is captured for each better than only monitoring light scatter and fluorescence in order to conduct an analysis. This dual technique allows for a morphological analysis to be conducted in addition to the study of traditional flow cytometric data ([23]).

2.3.1 Applications in Hematology

Diagnosing Hematologic Malignancies

Image flow cytometry is a very helpful technique for diagnosing hematologic diseases such as lymphomas and leukemias. It goes an extra mile to look at the shape of the cell. Classical flow cytometry, on the other hand, would detect the cell's abnormal populations depending on marker expression. Proceedings of a study conducted by [77] stated that this particular amalgam could identify various types of leukemia, as a result of which appropriate diagnosis could be done, which in turn could help in planning the treatment protocol.

Analyzing Cell Morphology

Image flow cytometry provides a significant increase in the morphological examination of cells. In hematology, proper knowledge of blood cell morphology is required for the proper diagnosis of a very large number of diseases, such as anemia, myelodysplastic syndromes, and other blood disorders. Image flow cytometry provides an abundance of precise morphological data about every cell, which can readily be combined with data from standard flow cytometry, again with a significant gain regarding diagnosis accuracies, as per the results obtained by [38].

Monitoring Treatment Response

More to this, another important application of image flow cytometry involves the monitoring of the treatment of individuals with hematologic diseases. For instance, image flow cytometry is important for the tracing of alterations in the morphological characteristics of the malignant cells and expression markers in diagnosing leukemia patients. It is through this that the physician can determine how effective the treatment is working and make any necessary adjustments if need be [77]

2.3.2 Image Analysis in Hematology

It is important that the ability to apply sophisticated imaging analysis techniques should be present to ensure that the image flow cytometry works properly. Such techniques would involve algorithms that can identify and quantify cellular features such as size, shape, and texture. These are all qualities that can help to distinguish between the many types of cells and diagnose problems that originate from them. Furthermore, [26] in their article state that the application of machine learning strategies to image analysis has become increasingly prevalent. More generally, the amalgamation of these systems leads to interpretations of the data that are more accurate and complete in terms of automation.

2.3.3 B-Cell Acute Lymphoblastic Leukemia (B-ALL)

The commonly found type of leukemia is B-cell acute lymphoblastic leukemia (B-ALL), and primarily it is prevalent among children. During the diagnosis and monitoring of B-ALL image flow cytometry was found to have some good results. Image flow cytometry was used during the study by [73] while analyzing blood samples drawn from people diagnosed with B-ALL. Specific morphologic features and marker expressions that would be able to separate B-ALL cells from normal lymphocytes were the main targets of the study project. Image flow cytometry appeared to be able to detect MRD at a more sensitive level in comparison to more conventional approaches. This had the effect of improving early diagnosis of relapse and guided changes in treatment.

2.4 Deep Learning Techniques in Medical Imaging

Deep Learning is a specific field of Machine Learning that has been doing exceptional work in various industries, with the medical imaging field being no exception. This technology, using multiple layers of neural networks for learning and extraction of features from data, will allow for the development of robust modeling techniques to be used for image analysis. The same will also develop models that can be put into place. The deep learning applied in the field of medical imaging has made terrific improvements in diseases, diagnosis of diseases, and remedies. All these resulted from deep learning. The present manuscript evaluates the basics, architectures, applications, and associated problems linked with various medical imaging techniques by deep learning of medical imaging, these techniques are used in medical imaging.

2.4.1 Principles of Deep Learning

Neural Networks and Deep Learning

Deep learning models are based on artificial neural networks, which are inspired by how the human brain is structured and operates. An artificial neural network is a network of connected neurons or nodes that receive information supplied to the system to be learned from and processed within several layers. Each neuron in the preceding layer receives input from the previous layer and processes the input into some function, which it then communicates to the neuron in the subsequent layer. This way, the model can simulate receiving information in a way that resembles humans receiving information in their neurons to process and dispatch.

A deep learning model is a model of artificial neural networks that are quite different from standard networks. This large difference is in the depth of the model, and the term "deep" comes from the idea that deep learning models are a magnitude of hidden layers as compared to traditional neural networks which usually consist of only one or two hidden layers in practice. Deep models can learn features, patterns, and representations that are quite complicated from the data, hence they are powerful enough to be able to output some perspective towards quite complex problems. Deep learning models are good at finding complex structures that lie within huge and sophisticated data sets, are robust, and conduct themselves in pertinent ways for that task.

The development of efficient training algorithms on large-scale datasets allows the training of deep neural networks to be followed by increased computation power. For example, back-propagation and optimization algorithms, such as stochastic gradient descent, are important for their very recent purpose of effectively training deep learning networks. Furthermore, large-scale datasets and the availability of powerful hardware, in particular GPUs, have reduced the training time to a great extent and, in return, enabled the application of deep learning across many fields, including medical imaging, natural language processing, and autonomous systems.

Convolutional Neural Networks (CNNs)

One of the most widely used deep learning architectures in the domain of medical imaging is convolutional neural networks (CNNs). These CNNs are, in fact, one of the most useful kinds of architectures for image analysis, too, because CNNs can automatically learn spatial hierarchies from the raw pixel data. CNNs acquire a high level of effectiveness as a consequence of this. Convolving filters with the input image to extract features, such as edges, textures, and forms, is done with the aid of convolutional layers. A chain of convolutional layers accomplishes this. Pooling layers further reduce the dimensionality of these feature maps, which helps in making the network computationally more efficient with reduced overfitting. Ultimately, these properties are unraveled by fully connected layers so that one can make predictions or classifications — in other words, properties deciphered by the fully connected layers result in predicting a class. In the words of [37].

This makes the hierarchical learning structure of CNNs powerful for image identification assignments. In pictorial information, the lower layers will gather simple features like edges, and the deeper layers will capture sophisticated information such as objects or pieces of objects. For this reason, CNNs are highly effective in the segmentation of pictures, identification of objects, and categorization of images. In recent years, convolutional neural networks have been widely applied in medical imaging for disease detection, anatomical feature segmentation, and image quality enhancement. Modern medical imaging for disease detection, segmentation of anatomical features improving image quality, among others have seen convolutional neural networks as one of the key players in modern medical imaging. These models have shown promising results and have great potential for current healthcare practices, particularly modern imaging and diagnosis practices [41].

Other Deep Learning Architectures

Deep learning architectures other than CNNs are obviously used in medical imaging. More deep learning designs are included: Recurrent Neural Networks, Generative Adversarial Networks, and Autoencoders. For example, RNNs can be effectively used to evaluate time-series data or sequences, such as electrocardiogram (ECG) signals or patient records over time. This ability is primarily because an RNN is designed to be used with sequential data and, with respect to this, it proves to be highly useful. The design of the RNN has the capability to trace and identify temporal relationships and patterns that show in time. Therefore, RNNs have been considered an appropriate candidate for predicting the course of a disease or its consequences for patients.

GANs and Autoencoders, on the other hand, have been used to carry out the generation and enhancement tasks of images. To proceed, artificial generalized adversarial networks are defined to have two neural networks that are learned simultaneously in a competitive process: a generator and a discriminator. The generator will be the one to produce the realistic images, then the discriminator will contrast it against the real photographs. As this process continues, the images produced through this process are enhanced and refined for a better-quality image. Such is essential in synthesizing images in medical imaging to be able to use the created synthetic images in expanding datasets, enhancing image resolution, or even anonymizing data. Autoencoders may be used for both noise reduction and image denoising and in feature extraction through unsupervised learning. Autoencoders take advantage of the learning process of image compression and reconstruction for the identification of significant features and patterns in medical images, hence developing diagnostic tools and systems using such specifications. Deep learning may thus change the game in medical imaging, leading to a difference in diagnostic accuracy and results in treatment. These are very different architectures but indicate the adaptability and potential of deep learning in this sense.

2.4.2 Most widely used models in Medical Imaging

Simple CNN Architecture

A Simple CNN can be used for the classification of images into viable and non-viable ones. This architecture for classification applies the following layers: convolutional layer, batch normalization, activation layer, pooling layer, fully connected layer, and dropout layer. Figure 2.1 shows a simplified version of CNN architecture. ReLU is the activation function used to introduce linearity and it ranges from [0, 1].

UNet Model

UNet is originally used for biomedical image segmentation since it provides a high performance in segments and localizes cells in any image accordingly. The UNet model is based upon the encoder-decoder structure combined with the skip connections. The contracting path or the encoder includes convolutional and pooling layers to extract contextual features and the

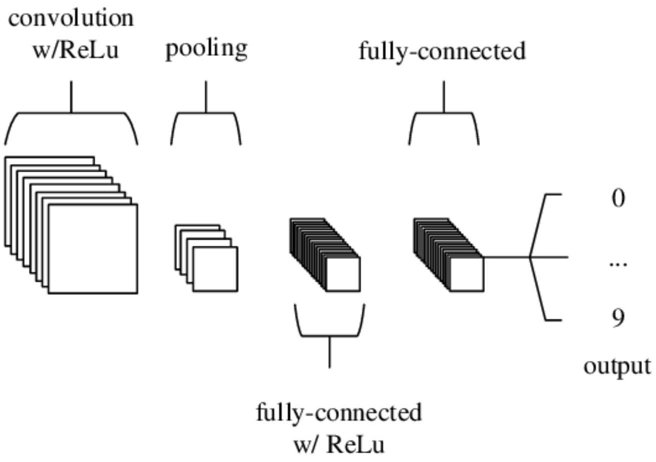

Figure 2.1: Basic CNN Architecture [48]

expanding path or the decoder generates the segmented image of the same resolution as the input image. Short connections between corresponding layers in the encoder and decoder paths are omitted to maintain the spatial information which results in better segmentation. The architectural demonstration is given in figure 2.2.

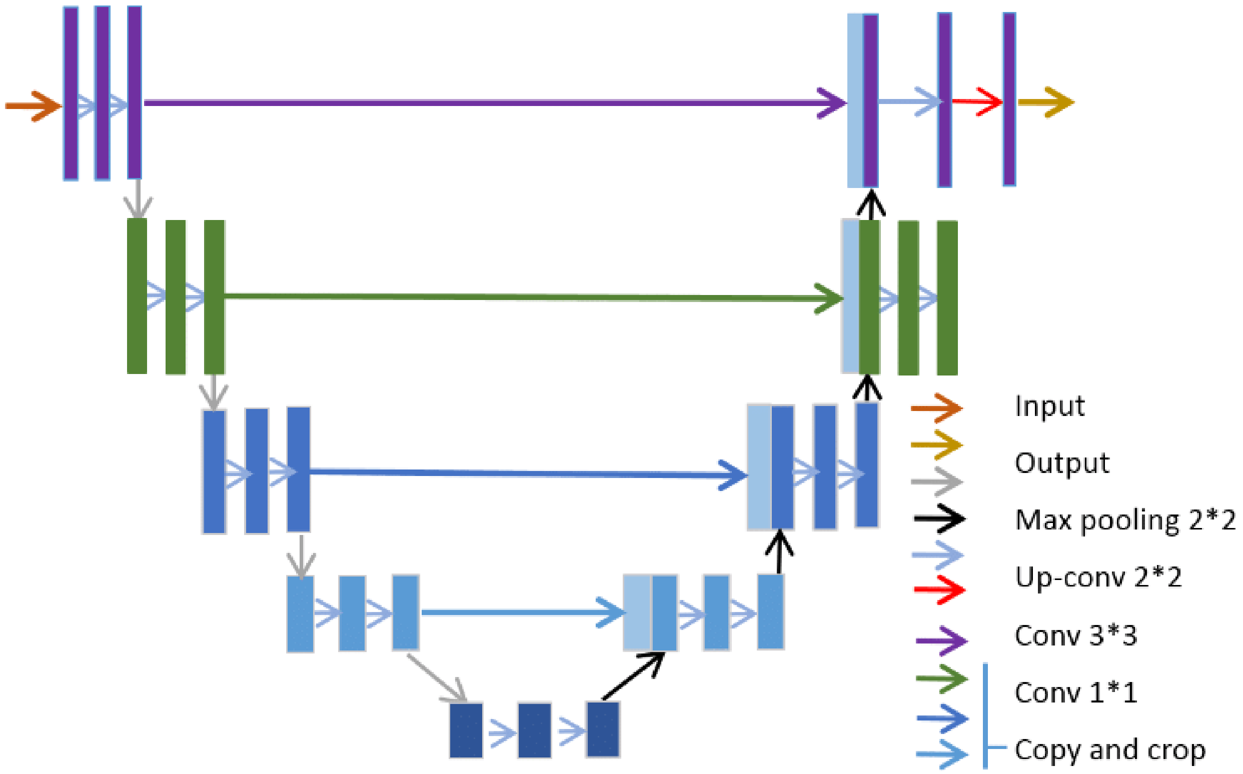

Figure 2.2: The UNet Architecture [11]

ResNet Model

ResNet is also known as Residual Network, widely used in deep learning and the application of residual connections that reduce the vanishing gradient problem, enabling the establishment of very deep neural networks. Here, ResNet34 is used as the feature extractor within the general model. After that, ResNet takes these segments from UNet and further transforms them into higher-level features required to describe cells adequately. The residual connections in ResNet are very deep, hence capable of preserving features in the cells' morphology patterns, which is fundamental for identifying RBCs, WBCs and Platelets. Its architecture is shown in figure 2.3.

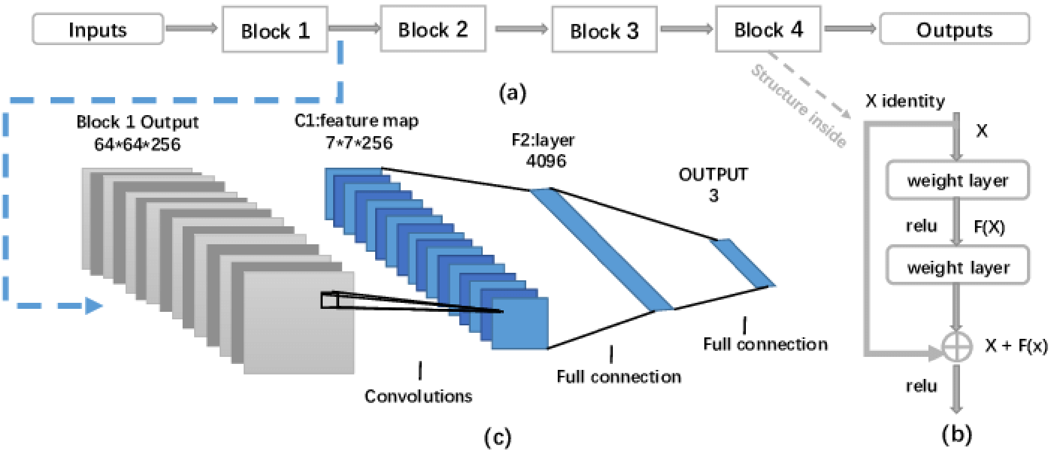

Figure 2.3: The ResNet Architecture [69]

Vision Transformer (ViT) Model

The Vision Transformer model works by splitting an input image into patches of spatial resolution and linearly projecting the patches into the proper dimension to be processed through a series of transformer steps. Each patch is considered as a token in a sequence, so the model can learn the context and long-range dependencies over the whole image. ViT-Base-Patch16 is even more particular since it splits images into patches of 16X16 that are processed by the transformer encoder. Figure 2.4.

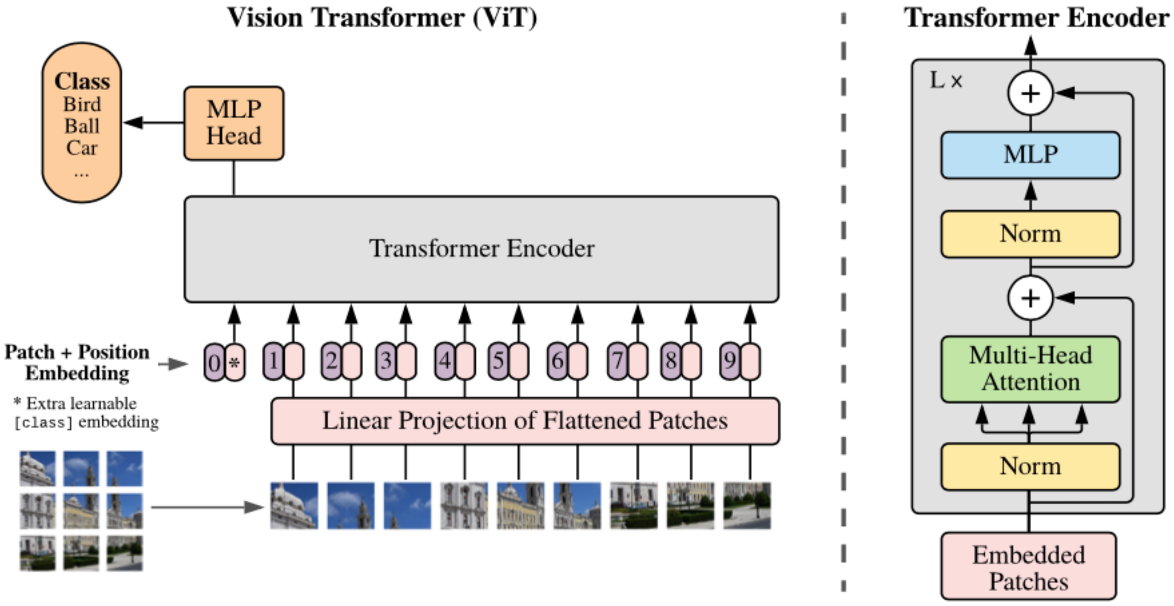

Figure 2.4: The ViT base model Architecture [51]

DenseNet121

DenseNet121 also known as Dense Convolutional Network is designed to push as much information as possible up and down the density connection of the network. All the layers in DenseNet121 directly receive feature vectors from all the earlier layers, and in turn, feed feature maps to all the subsequent layers [29]. The formula for the output of a dense layer can be written as:

$$\mathbf{y}_l = H_l([\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{l-1}])$$

where $\mathbf{y}_l$ is the output of the l-th layer, H_l is the composite function of batch normalization, ReLU, and convolution, and $[\cdot]$ denotes concatenation.

This higher level of connectivity also improves the feature extraction ability of the model since it can focus on minute details of the morphology of specific cell types like the lymphocytes and the details regarding their nucleus and chromatin. A simple architecture of DenseNet is given in Figure 2.5.

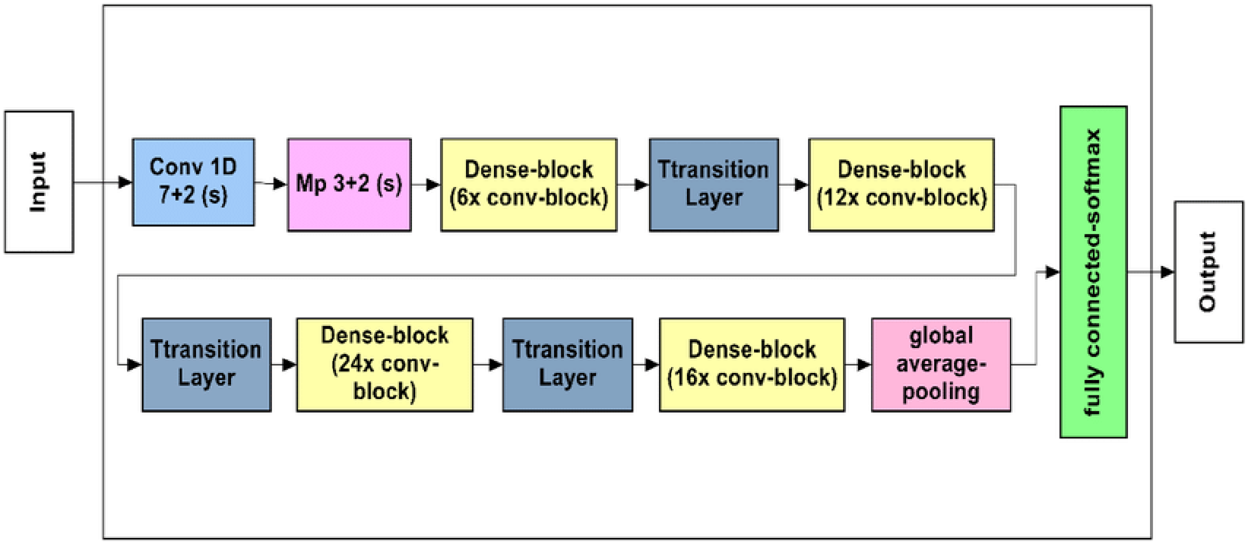

Figure 2.5: The DenseNet121 Architecture [70]

YOLOv5 Model

YOLOv5 works in a way that it takes the input image and then divides the input image into multiple grids and then predicts multiple bounding boxes and class probabilities for each cell of the grid. This made YOLOv5 efficient in predicting numerous bounding boxes and classifying these boxes all at the same time. The use of convolutional layers, the use of anchor boxes, and non-maximum suppression make the model capable of handling the highly dense and complex image of blood samples. The Yolov5 network architecture is divided into three sections: Yolo Layer, (2) Neck, and (3) Head. CSPDarknet is the Backbone. The information is supplied into PANet for feature fusion after being first input into CSPDarknet for feature extraction. Yolo Layer finally outputs the detection results (size, location, score, and class). Figure 2.6 shows an architectural demonstration of YOLO.

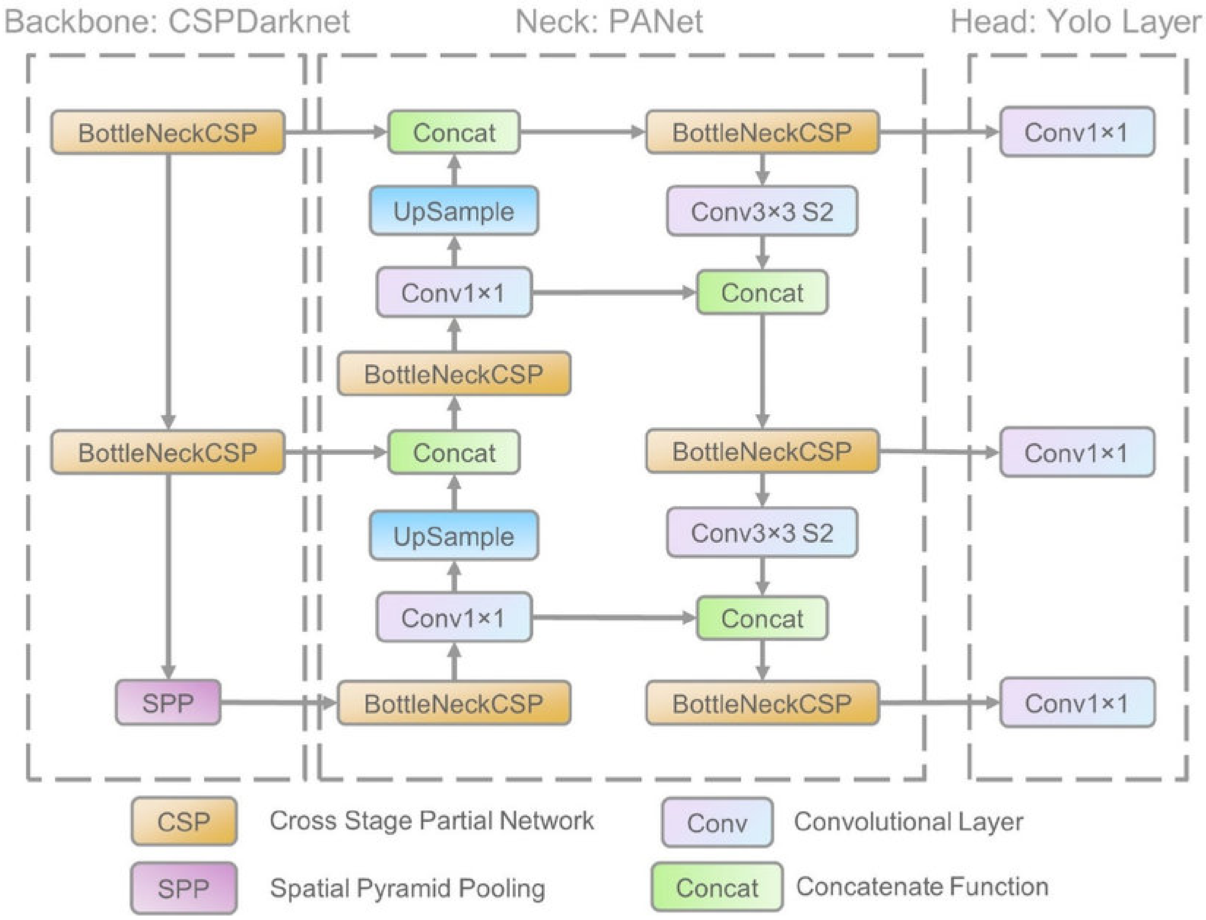

Figure 2.6: The YOLOv5 Architecture [78]

2.4.3 Case Studies in Deep Learning for Medical Imaging

Case Study: Brain Tumor Segmentation

Brain tumors are very heterogeneous in both structure and location, they lead to several important issues in terms of diagnosis and therapy. During surgery and planning radiotherapy, an accurate segmentation of the parts of a tumor that need to be scanned using MRI is required. [34], developed a deep learning model known as DeepMedic for segmenting brain tumors in MRI.

The MRIs were sourced from a 3D CNN architecture to extract local and global contextual levels called DeepMedic. For achieving the segmentation process of the different parts of brain tumors like the core of the tumor, the augmenting tumor, and peritumoral edema, this model achieved training from the myriad of pictures on the brain MRIs, which involve a few differing kinds of brain cancers and imaging irregularities. However, this model achieved the correct and nearly accurate performance in the segmentation process of different parts of brain tumors, including the core of the tumor, the augmenting tumor, and peritumoral edema. The segmentation process is vital in determining the tumor limits used in creating or formulating disease-specific treatment strategies.

The comparison was made with the performance of the model and with the expert manual segmentations and produced measures that were almost the same; some even meant that the model outperformed his human performance. Said model can guide neurosurgeons and radiologists to better decision-making, which will eventually translate to better patient outcomes. This is one highly promising model. The clinical implementation of DeepMedic, among others analogous to it, into the aspiration of brain tumor management through effective and precise ways, basically improves the amount of equal precision in both therapeutic ways-surgical and non-surgical, according to [32].

2.5 Methodologies of Integration of Deep Learning with IFC

IFC is the abbreviation of Image Flow Cytometry. It merges quantitative power, classically in flow cytometry, with the detailed imaging of classical microscopy. Integration of deep learning into IFC can unravel the potential of changing this field through automation of the image analysis process; it can improve accuracy and detect subtle patterns that are usually missed by human observers. This section elaborates on the integration of deep learning in Image Flow Cytometry, based on the methodology, applications, benefits, and issues directly connected with the integration.

2.5.1 Image Processing

1. Data Normalization:

   Normalization is very essential precursor to preparing image flow cytometry data to feed into deep learning model development. This is done by scaling the pixel values of images to a standard range in order for the neural network to handle the data well. This should reduce the impact of the numerous conditions of light and imaging settings, both of which can pull out differences in the data. Normalization seeks to make the performance of the models more certain and stable across many datasets [20]. This is done through normalizing the data which is usable with the pixels of the images.

   These techniques further use normalization techniques such as z-score normalization and min-max scaling for the intensity values of the photos. This is performed so as to introduce each image having an equal contribution while testing the model with images. It is important for the reason that the process hinders the model from a partiality toward the images having higher intensity values and thus contributes to bringing about an improvement in the overall accuracy and robustness of models used in IFC. This process is based on the learning process from the data, which makes the model capable enough to recognize the objects or features in an image.

2. Data Augmentation

   Data augmentation is performed through operations: rotation, scaling, flipping, and cropping, to mitigate difficulty due to scarce datasets. For example, these methods fight overfitting and artificially inflate the dataset, in this or that form, enhancing the generalization capacity. According to [52], data augmentation is the process of creating variations for the current images. This way, a model learns better from a larger variety of examples and generalizes better on new data that it hasn't seen before.

   Furthermore, it includes other methods of augmentation, such as random erasing and elastic transformations, to add even more variation to the training data. These methods make the model robust in dealing with these kinds of variations when applied through the simulation of real-world fluctuations and artifacts that can appear in IFC images. According to [65], this technique not only enhances the model's performance on the training data but also enhances performance over a wide range of complex datasets such models come across in real-world applications.

3. Image Segmentation

   Image segmentation is the process of partitioning an image to give separate meaningful sections. Through the segmenting algorithms for separating the gathered individual cells from the background, the deep learning models could be implemented more correctly through IFC. Accurate segmentation is absolutely necessary here so that the cells can be rightly quantified and classified on the basis of the features that are picked up by these models relevant to the cells being targeted [60]. Through this, the quantification and classification of cells will be precise.

   In terms of the applications of IFC image segmentation, deep learning models like U-Net and its various modes are inclined toward the mainstream. All these models are easily able to differentiate cell boundaries and accurately segment overlapping cells, two major challenges one might face in any IFC image. These advanced segmentation methods yield the ability to acquire very high recognition and analysis of individual cells. As told by [7], they will provide a more reliable and in-depth cellular investigation.

4. Applications

   The combination of deep learning and IFC has a huge number of potential applications both in the domain of research and in the domain of application in therapeutic practice. This technology enables an all-encompassing morphological study, identification of unusual cell populations, and automatic identification with classification of varied types of cells [16].

For example, deep learning IFC can be optimized to help diagnose circulating tumor cells in the blood. This offers incredibly fundamental information that can be integrated and used in the diagnosis and monitoring of cancer. This technique can also be utilized in immunology to have a better characterization of the immune cell subsets, which will aid in the understanding and treatment of immunological-related diseases. [41] have suggested that automation and high-throughput capability in this technology boost up, at a great speed, scientific and clinical diagnostics, creating all new ways for personalized medicine and targeted therapeutics.

2.5.2 Deep Learning Archietecture

1. Convolutional Neural Networks (CNNs)

CNNs build the bottom level of deep learning in image analysis because CNNs can learn spatial hierarchies from the use of convolutional layers. For that reason, CNNs are capable of learning spatial features automatically and adaptively from the input images, such as edges, textures, and patterns. Therefore, these are very important features in tasks that involve the recognition and classification of images. These layers enable CNNs to automate these features. For example, [39] demonstrated that CNNs can learn to extract information from images on complex information like cell morphology and fluorescence intensity, hence being able to do the task of precise classification and detection. IFC uses a convolutional neural network.

The common architecture of CNNs displays that the network consists of a large number of convolutional layers followed by pooling layers. In processing the important properties of data from a smaller number of dimensions of the input data, these layers reduce the spatial dimensions gradually. In employing the process of immunofluorescence (IFC) technology, this facility of hierarchical feature extraction is of extreme importance to be employed in processing important high-resolution images that distinguish between cell types and the states of cells. Therefore, the identification of the cellular capability, which plays an important role in the diagnosis of different disease conditions and also in tracking the responsiveness of treatment, could be automated by researchers using the convolutional neural networks of feature extraction. This automation of the network, according to [41], not only enhances the precision of analysis but is also considerably faster. Meanwhile, it processes large-scale datasets by raising the capability of processing data in the field of biomedical analysis.

2. Transfer Learning

Transfer learning is a tool affecting the improved performance of restricted volumes of annotated data in a particular domain by advancing pretraining with relatively large general domain datasets. This goes a long way in proving useful in the domain of international finance, where annotated data is scarce, and the cost of annotation is high. The pre-trained models include VGG, Res-Net, and Inception, which have been trained with large datasets of images, such as ImageNet. In this way, a wide range of visual properties have been captured, which are also very useful in the domain of hematology and other medical domains [50].

By fine-tuning these pre-trained models on IFC datasets, performance on the IFC of researchers' models can be improved significantly from only a few samples bearing labels. This method also helps to be generalized for the capability that makes it resistant to the fluctuations in data. This becomes very useful in the field of medical imaging, which has not been possible to make huge datasets whose data has been relatively well annotated. The application of transfer learning in IFC makes it possible to effectively use information collected beforehand, thus reducing the time and effort involved in the collection of significant data and its annotation.

3. Autoencoders

Two potential applications can be done with autoencoders: unsupervised feature learning and picture reconstruction of a certain kind of artificial neural network. This means having an encoder that compresses the input into a representation resulting from a few dimensions of compression and a decoder that reconstructs the input according to the representation resulting from the compression. [27] say that IFC applications with autoencoders can be used for filtering noise in pictures, the identification of anomalies, and data compression which results in keeping data more compact in storage and fast in transmission.

One of the main benefits of using autoencoders within IFC is their capability to produce a reduced representation of the information. Several tasks can be realized with this compact representation in the application further downstream. For example, a denoising autoencoder can be used to improve the quality of images in IFC by suppressing artifacts and noise, leading to more accurately analyzed cells. Autoencoders could also be used for anomaly detection, which, in a place like learning to recognize the usual patterns of images of cells, quite easily detect deviations from these patterns as likely to be abnormal. The ability to recognize abnormality in general is a foundational skill, according to [82], in the ability to find deviant or rare cells that could be indicative of disease.

Autoencoders are most helpful in managing the very huge data volumes IFC generates through data compression. Compressing the data, autoencoders reduce the volume of space for storage and improve data transmission. This makes it easy to share and analyze enormous datasets across many research institutes. Maintaining such high efficiency is very important for the success of joint research endeavors, explained [76].

2.5.3 Training and Validation

- Data Splitting

Most of the time, the dataset is split into training, validation, and test pieces to ensure that the model is trained and evaluated appropriately. The model is trained using the training set, which helps it learn from the data and adjust its parameters. The validation set is used for making decisions about tuning hyperparameters, model design, and optimization tactics. This will avoid the problem not only of overfitting the model but also protect the generalization power for new data. The test set finally measures the performance of the model on the observation of data that were not used until that point. It gives an objective estimate of the correctness and efficiency of the model in applications with a ground truth in the real world [20].

Proper splitting of data ensures that every piece of data will represent the overall data set, hence capturing all forms of diversity as well as complex patterns lying beneath. Most techniques, such as stratified sampling, will at least maintain a similar class distribution between the training, validation, and test sets. This is important, especially in a medical imaging dataset, as some classes in the dataset are liable to be underrepresented. Consequently, making model evaluation fair and ensures that the model performances are constant between a range of data subsets [35].

- Cross-Validation

Generalization capability can be measured with cross-validation methodologies such as k-fold cross-validation. The k number of folds in the validation process are created from the dataset in such a manner that they have equal sizes. Of the k number of total folds, k-1 folds will be used to train the model and one fold will be used to validate the model. The cycle takes place k times so that each time, one fold will be the validation set. The results are averaged to give the final estimation of model performance. It greatly increases the accurate estimation of the model's generalization to new data and reduces the threat of overfitting the data [22].

Cross-validation is very important in IFC, especially where labeled data is sparse. It is, therefore, a vital technique. Cross-validation ensures more than one subset of the data is used in the two procedures of the data: training and the other for validation. This can achieve the goal mentioned above by exploiting the highest available data. It is, therefore, not only more reliable in enabling one to know the performance of the model but also helps in knowing the best configuration of the model to be adopted. More advanced cross-validation techniques, according to [74], further help enhance the robustness of the process of model evaluation, which may be used in layered cross-validation. These, in turn, help in hyperparameter tuning and selection of models.

- Performance Metrics

Full performance metrics are required in the assessment of the deep learning models built for the IFC. These metrics are required to elaborate on how good a model is in the classification and identification of cells. A few examples of such standard metrics include accuracy, precision-recall, F1 score, and many others. An accuracy measure is the share of the total examples classified correctly out of the entire number of instances, though this might be very misleading when used with datasets that are imbalanced, where one class is much more abundant than the rest. However, it shows a general impression of the model.

1. Positive predictive value, precision, is a measure that establishes the ratio of true positive predictions out of all the positive predictions—defining how accurate the positive class predictions are.
2. Recall, sensitivity, or true positive rate, is the proportion of actual positive cases that got predicted correctly. This means the model can detect all the positive cases.
3. The F1 score is the harmonic mean between precision and recall, thus giving a single statistic that balances between these two concerns that are energetic. This is particularly helpful in scenarios where datasets are uneven.

Such indicators, taken in their totality, give an overall performance analysis of the model and can, therefore, identify its strengths in addition to giving avenues for optimization. These metrics, as given by [67], come in handy in fine-tuning the model for maximum performance with IFC, whose most critical aspect of categorizing the cells precisely is crucial in ensuring that images of cells are analyzed in the most reliable, accurate manner.

2.5.4 Applications in Hematology

- Leukemia Diagnosis

Leukemic cells can easily be detected using deep learning models in combination with an image flow cytometer. These algorithms analyze morphological features of the cells and corresponding fluorescence indicators. For instance, the use of CNNs can detect free cells and at the same time, differentiate between normal and affected cells, hence diagnosing leukemia at a much earlier time and with great precision. The CNN models have utilized the vast visual data compiled through IFC to detect, with precision, the tiny changes in morphological attributes of the cells that indicate leukemia. According to [21], such minute differences cannot be detected using the conventional methods.

MRD detection is of much significance for the purpose of determining the efficacy of leukemia treatment and predicting relapse. An advanced IFC integrated with deep learning can identify rare leukemic cells from an overwhelming number of normal cells. In turn, the MRD assessment is both sensitive and highly specific compared with current methods. High sensitivity is required so that even minute numbers of cancer cells are detected. These cell numbers, if not detected, remain in the body and stealthily or aggressively predispose the patient to relapse, respectively. This is of paramount importance for the best possible outcome of treating and managing such tumors [77].

- Disease Diagnosis

Deep learning models can be created for the analysis of longitudinal IFC data and prediction of the possible disease trajectory and patient diagnosis. With this approach, patterns and biomarkers indicative of the diagnosis of a disease are revealed from learned past trends for the linear planning of treatment. Predictive modeling in this respect reveals early markers in the development of illnesses, supporting an understanding of how different persons respond to treatment across time [15].

For instance, the mass and fluorescence intensity changes in cell populations can be done at various time points to predict the likelihood of relapse for a patient. Such prediction acts as a roadmap for clinicians to refine treatment plans for a patient, which may be very well proven towards longevity and enhancing the quality of life. It can be felt that predictive modeling in IFC is a tremendous new development in the field of personalized medicine; it permits the selection of such interventions that are much more precise and efficient, according to [9].

It is the product of deep learning and IFC on research findings that permits the drawing of the statistical models, which are both stratified and associated with respect to patients' cellular profiles from whom they were drawn. This stratification helps overall in boosting the general treatment efficacy and assists in adapting treatment programs to the specific cases in question. If the profile for the patient is high-risk, then closer follow-up can be done using therapy more aggressively, while the ones with low-risk profiles may be availed with less intense interventions [2]. Accurate stratification is most important in diseases like leukemia because the responses to treatment can vary dramatically from patient to patient.

- Enhanced Accuracy and Efficiency

For being capable of identifying minuscule patterns and anomalies in IFC images, deep learning models can analyze gargantuan datasets with precision and speed above the human level. This enhances IFC's diagnostic capabilities and even predictive accuracy. Automated image analysis has been reported to process data within a very short period of time. The frequency with which manual review is done is thereby minimized, thereby enhancing throughput.

Another benefit of deep learning is that it reduces the risk of human error by many degrees, therefore resulting in infinitely more reliable and consistent outputs. This means a lot in clinical conditions, as an erroneous diagnosis would mean inappropriate management of patients. As documented by Litjens [41], the better accuracy guaranteed by deep learning algorithms ensures that important decisions are taken based on the most reliable information available at present, and with time, this invariably results in better patient outcomes.

- Automation and Scalability

Since the image analysis through the combined mechanism of deep learning and IFC is almost instant, concerning the fact that less human labor, on the pathologists' part, is employed in analyses. Large amounts of data are handled with ease, thereby making the technology more scalable for use in clinical setups. According to [41], handling the high demand for diagnostic services can be done using automated systems, especially for situations with a resource base that is limited or in cases where the number of patients to be handled in one setting exceeds expectations.

The IFC, enhanced by deep learning, is especially significant for its scalability in the model, able to be applied in the most diverse ranges of healthcare settings. Such modalities help health professionals channel their efforts toward more challenging patients and research by taking routine procedures off their hands. Overall effectiveness and efficiency of the health system are increased enormously from this. This capability to scale out is necessary to meet the ever-increasing requirements of modern-day healthcare systems [71].

- Discovery of New Biomarkers

The new biomarker discovery from the IFC data would be unreachable through more conventional techniques of analysis. The discovery of biomarkers can, further, lead to new mechanistic insights into the disease together with probable treatment target discovery. Deep learning can tap patterns and relationships that conventional methods may tend to overlook through the analysis of large and complex datasets. These deep learning models would help in new biomarker discovery from the IFC data. Discovery of such biomarkers may further lead to new mechanistic insights into the disease along with probable treatment target discovery.

With the help of deep learning, patterns, and relationships that conventional methods might tend to miss would be tapped through the analysis of large and complex datasets. Such biomarkers, newly identified, have the potential to revolutionize the diagnosis and treatment of illness by making medicines more precise and specific. For example, the identification of specific cellular changes that are reflected by responses to treatments could very well be a most helpful guide in designing individualized regimens of treatments, that are more effective and less toxic. This is of great potential benefit in the prospect of an extended field of hematology and better care for patients, as the work is being done incessantly and further developed in this area.

2.5.5 Key Studies

Table 2.1 shows the key findings of studies done in the past which add significant findings on the application of deep learning in diagnosing of B-ALL using IFC.

Study	**Objective**	**Methodology**	**Accuracy**	**Implications**
Jing Sun, Lan Wang [68]	Used deep learning with light scattering microfluidic cytometry for leukemia classification	Combined deep learning with light scattering techniques to classify acute lymphocytic leukemia (ALL) cells.	Accuracy is 91.5% in label-free classification	Provides a non-invasive diagnostic tool, potentially reducing the need for traditional staining methods.

Maila Claro [8]	Developed CNN models for diagnosing acute leukemia	Applied CNN models on image data for the diagnosis of leukemia, focusing on differentiating between types of leukemia.	Achieved high accuracy of 94.7% in classifying leukemia types.	Highlights the utility of CNNs in automating leukemia diagnosis, aiding in quick and accurate clinical decision-making.
Ibrahim Abdulrab Ahmed [1]	Explored hybrid techniques for ALL diagnosis using CNN feature fusion	Merged features from different CNN architectures to improve ALL classification accuracy.	Achieved an accuracy of 95.2%.	Suggests that hybrid models can leverage the strengths of different architectures for more robust diagnostics.
Niranjana Sampathila [64]	Developed a customized deep learning classifier for detecting ALL using blood smear images	Created a tailored deep learning model specifically for blood smear image analysis in ALL detection.	Reported an overall accuracy of 93.8%.	Demonstrates the potential for specialized deep learning models to improve diagnostic accuracy in hematological diseases.
Adnan Saeed [61]	Implemented a deep learning-based approach for ALL diagnosis	Utilized deep learning algorithms to classify ALL cells, focusing on improving diagnostic accuracy and speed.	Achieved an accuracy of 96.1%.	Suggests deep learning as a powerful tool for enhancing diagnostic processes in ALL, potentially improving patient outcomes.

Table 2.1: Summary of studies on deep learning approaches for ALL diagnostics

This table shows the research objective, key findings, and implications of each study, providing a brief and comprehensive overview of each study done in the past on the topic of application of Deep Learning in diagnosing Acute Lymphoblastic Leukemia using Image Flow Cytometry.

2.5.6 Challenges

1. Data Quality and Quantity

To train powerful deep learning models, only high-quality datasets that are labeled can yield good results. Due to heterogeneity in sample preparation, staining methods, and imaging conditions, it can be challenging to obtain large datasets that are well-annotated using image flow cytometry (IFC). According to [41], the standardization of a dataset capable of accurately reflecting all possible cases presented at hospitals is complicated precisely because it contains those components that make any data prone to certain inconsistencies and biases.

To address these challenges, multiple solutions — such as data augmentation, transfer learning, and artificial production of new synthetic data, among others—are being experimented with by researchers. Note that the dataset can be artificially inflated through the use of data augmentation techniques. All of these techniques try to apply various transformations like rotation, scaling, and flipping, which help the model learn something that eventually increases its generalization power. The best part about them is transfer learning enables one to take models that have already been trained on big, generic datasets and then fine-tune them for specific tasks using small datasets pertaining to the same domain. [65] mention that synthetic data generation with techniques like GANs seems a promising solution as it can produce realistic images. Moreover, these realistic enhancements to the training dataset may provide more diversity in samples from which to cheer labels.

2. Model Interpretability

One often-heard concern about deep learning is that because these are “black box” models, it’s almost impossible to understand how they actually arrive at their conclusions. This lack of transparency presents a major barrier for clinical applications (where we need to know why something was diagnosed), [63]. It is important that, especially when these decisions can directly affect patient care, clinicians trust and understand the judgments made by artificial intelligence systems.

Researchers are developing strategies to create more interpretable findings, as they identify the specific portions of a picture that contributed most to the decision-making process of their model. Visual explanations using techniques like Grad-CAM (where a heatmap is superimposed on the input photos to point out which areas on the image contributed the most to the prediction) are one approach. [58] have proposed several methods (e.g., SHAP (Shapley Additive Explanations) as well as wider applicability solutions like LIME (Local Interpretable Model-agnostic Explanations) to gain information concerning feature value and model response attitudes; this provides more transparent output that is easier for clinicians to understand.

3. Computational Resources

Developing deep learning models requires significant computational overhead, including immense GPUs and huge memory size. However, this could be a constraint for organizations as they may not have the required infrastructure in many healthcare settings and research institutes [39]. This is challenging due to the hefty monetary constraints of cutting-edge computational power, continuous upkeep, and updates.

To overcome these problems, high-performance computing clusters and cloud-based solutions are increasingly being used. These platforms allow institutions to train and deploy deep learning models without requiring a huge upfront investment by offering scalable and affordable access to powerful computational resources. The computational load is under control, and deep learning is made more approachable through optimization techniques, including model pruning, quantization, and efficient neural network topologies, such as MobileNets [28]. All of these methods are coming to market.

2.6 Overview

In this chapter, we have endeavored a critical analysis of the current literature on ALL disease profiles, and the traditional diagnostic tools designed that comprise of cytogenetic analysis, immunophenotyping, and molecular testing. We also discussed the updates and subsequent uses of image flow cytometry in hematology with an emphasis on its capability to better differentiate between the cellular structures. Also, we covered how DL is combined with medical imaging and how these kinds of methods enhance the ability to detect diseases, diagnose them, and plan treatment.

Although there have been numerous developments done to diagnose ALL and the future outlook of deep learning is quite optimistic in this research domain, the following gaps are observed: Convention methods on the other hand are efficient, however; they take lots of time and require the services of specialized individuals, thus, leading to inconsistencies in diagnosis. Furthermore, the blend of deep learning with image flow cytometry is relatively in its infancy and thus needs further data to implement the method effectively in clinical practice.

To fill these gaps, Chapter 3 of this dissertation proposal will describe the research approach of our study, explaining the procedure that would be adopted to design and implement a deep learning-based diagnostic framework for ALL utilizing image flow cytometry. This involves the process of data accumulation, initial data preparation methods, and the creation of our deep learning platform. Thus, the subsequent chapter, constructed from this literature review, is designed to offer a detailed view of the presented approach and its possible beneficial influence on the process of ALL identification.

Chapter 3

Methodology

This chapter will describe the steps of data collection, data preprocessing, and methodology adopted in this research, which is building an automated pipeline for the diagnosis of Acute Lymphoblastic Leukemia using image flow cytometry data. The pipeline includes six significant steps, each designed to process and analyze images of blood cells with different deep-learning models and techniques. The objective is the efficient classification of blood cells and to identify all their apparent characteristics so that the respective subtype is categorized, which again is critical for an effective diagnosis regarding diagnoses and treatment planning.

3.1 Data Collection

Regarding building a deep learning model for the diagnosis of B-cell Acute Lymphoblastic Leukemia (B-ALL), data acquisition is a crucial stage that must be performed. As for the type of data, high-resolution images of cells that were collected using image flow cytometry served as the primary data source of this particular study. To ensure that the dataset only includes complete data and that it is representative of the treatment of hematological cancers, these photos were collected from a reputable healthcare organization treating these diseases.

3.1.1 Ethical Considerations

Increasing the availability of the patient selection criteria is evidence that the analyzed work aimed to include a diverse population of B-ALL patients, but its structure was designed to guarantee such goals. The selection criteria involved the participants being adults of various ages, who were at different stages of the manifestation of the disease, had different reactions to the treatment. For the purpose of training a model that is competent to improve its performances drastically in clinical situations of distinct types, this element of diversity is inevitable. Each process of the data collection procedure was handled while following the major ethical concerns in the process. There was also consent from the Hospital's Institutional Review Board (IRB) that ensured that all the treatments provided met the set ethical virtue. This study followed the principles of informed consent from patients or their legal representatives and data anonymization to protect patient's identity.

3.1.2 Image Flow Cytometry

To enrich the resolution images of leukemic cells from the bone marrow and peripheral blood samples the image flow cytometry was used. This method not only makes use of the morphological and the fluorescent features of cells, but also makes use of both the quantitative characteristics of the standard flow cytometry as well as the pictures. The first procedure carried out in this procedure involved getting informed consent from the patients to take their bone marrow and blood samples. Specific fluorescent antibodies for a number of cell surface markers that seem to be associated with B-ALL were employed to the treatment and staining of these samples before the analysis was conducted. Next, the stained samples were processed in image flow cytometry to produce image maps for the cells in the samples.

3.1.3 Data Quality and Storage

Important to mention that during the process of collecting data, one of the crucial points was to guarantee the high quality of the images that were collected. Image flow cytometry, as a technique, is used to obtain tremendous data with quantitative fluorescent measurements and digital image data acquired from the procedure. A number of actions were performed in order to preserve the quality of the data:A number of actions were performed in order to preserve the quality of the data:

- Performing Calibration and Standardization: The image flow cytometer was calibrated regularly to ensure that the fluorescence intensity was being measured properly and there was a good quality of the pictures that were being taken and they were in some certain extent identical. While taking the sample and staining, it was ensured that various procedures were taking place in a very controlled manner so that variation could be minimized.

- Quality Control: To filter out low-quality pictures, measures of automated quality control were employed. It is crucial to note that images that had most of the artifacts, poor focusing, or images that had not much of the fluorescence signal were not included in the set.

- Data annotation: To get ground truth labels for the training and validation of the deep learning model, a small proportion of photos was reviewed by a group of hematological experts. Regarding these annotations, information concerning the leukemic cells, normal cells, as well as other suitable and relevant morphological features were noted down and labeled on a platform named LabelImg (figure 3.1).

Each data point that was gathered was saved in the central database of a hospital. Security was managed at this database and adequate measures were taken in order to contain access and ensure proper security for the data. Thus, in a case where data was at the risk of being lost, the solution involved making frequent backups.

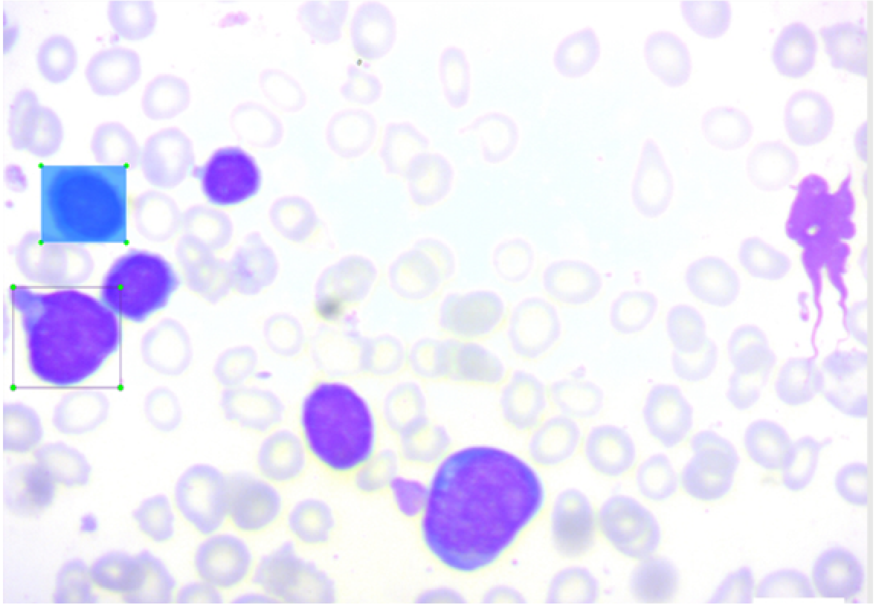

Figure 3.1: LabelImg software is used for annotating cells on blood smear images.

3.1.4 Data Collection Challenges

There was a need to undertake a lot of planning and problem-solving to reduce the barriers that were met while collecting information. Some of these difficulties included: Some of these difficulties included:

- Variability in Sample Quality: The processes of sample collection and preparation may vary and this may in turn affect image quality. This issue was somewhat resolved with regard to standardizing methods and offering training to the lab staff carrying out the tests.

- Ethical and Logistical Constraints: Due to the ethical and practical issues, for patients and medical staff's agreement and compliance with the requirements to organize it conveniently, proper coordination was required. There were essentially two important categories: Firstly, the implementation of clearly understandable general rules and requirements; and Secondly, the exchange of information with the patient and/or the patient's family.

- Technical Limitations: The process of using flow cytometry is technical in nature hence technical restrictions are evident it is embedded with a lot of technology that requires frequent tuning. Therefore, while interacting with other experts and manufacturers of the equipment, we could always guarantee that the device was optimal in its functioning.

This section of research, which was aimed at the collection of data, was performed with the intention of obtaining high-quality Image Flow Cytometry data from a hospital that treats children with B-ALL. The ethical issue was solved properly, the quality of the data was controlled, and the logistical issues were successfully solved, which allowed it possible to obtain a large and rather reliable dataset. This paved way for the subsequent data preparation and the build-up of deep learning models.

3.2 Data Preprocessing

The preprocessing stage is an important activity as it leads to the preparation of the image flow cytometry that has been attained to formulate the deep learning model. In this phase, a number of activities are performed with the aim of preparing the data for the cleaning, transformation, and reshaping into a suitable format for training and testing the model. Regarding the noise issue, it is concluded that the change in the quality of the data set and, accordingly, the change in the effectiveness of the deep learning algorithm can be achieved through effective preprocessing.

3.2.1 Data Cleaning

Data cleaning is the first task of the preprocessing step where any abnormality in the raw image data is removed. In this process, the set that is being used to train the model will be as reliable and dependable as one can possibly make it. Images produced during image flow cytometry includes that are blurred, images that stained incorrectly, or images that were in some way affected by technical problems that were recognized and excluded. The automated method was employed here as it is aimed at eliminating photographs that had significant levels of artifacts. This is done to ensure that only the best images would be stored for future reference [41]. The methods of noise reduction are used with the aim of reducing the level of image noise that is incorporated. The randomness of the image can also be attributed to factors such as electronic interferences and/or some mistakes made during sample preparation. Lecun [39] noted that in a bid to enhance the outlook of prominent cellular features while preserving the contents of such features, filters of Gaussian and median kinds were used to smoothen the images.

3.2.2 Data Normalization

Normalization is an essential step in the preprocessing process that entails bringing the range of pixel values included in the photos to a scale that is constant throughout. Taking this step contributes to the enhancement of the convergence rate of the deep learning models while they are being trained. The actual numerical values of the pixel in the fluorescence intensity images were normalized to 1 or over a fixed range which is between 0 and 1 calculated by:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

This procedure is known as intensity normalization. In this procedure, an attempt is made to color balance the pixel intensity to prevent inconsistency among the photos thereby enhancing the performance of the model [14]. To increase the visibility of cellular characteristics, contrast enhancement approach like the enhancement of histograms was used. In the opinion of Krizhevsky [37], this stage aids the deep learning model in the process of feature extraction since it stimulates an enhancement in the levels of discrimination of seminal morphological attributes in the images.

3.2.3 Data Augmentation

There is a technique that is employed in the attempts to enlarge the volume of training datasets artificially, and this is referred to as data augmentation. This method concerns applying numerous modifications to the images that are already included. In DNN, they provide a way of reducing the overfitting and improving the model's capability to generalize. Some of the geometric transformations that have been used on the images included; rotation, flipping, scaling, and translation among others. These changes were among the geometric transformations. There were other transformations but these are said to be the geometric transformations. Shorten and Khoshgoftaar [65] noted that these alterations create new original images' versions, which, in turn, are used to offer the model a range of typical instances in which it can enhance its stability. Adjustments were made to compensate for imaging situations of the brightness, contrast, and saturation of the images. These changes were called intensity variations. For this reason, during the augmentation stage, the model is capable of handling variations in the quality of the image and lighting which are quite common occurrences in real-life situations [52]. It ensures that the model can accommodate all these changes as a result of the previous step.

3.2.4 Feature Extraction

Feature extraction is an important process of data preprocessing in which useful features are selected as well as extracted from the images. It entails the conversion of the raw image data into a set of features that is most suitable to be ingested by a deep learning model. Other morphological characteristics like cell size, shape and texture of the image sections were then studied. These features are especially significant for subclassifying many cell types and identifying leukemic cells, particularly in B-ALL [60]. Similar to the intensity values of the nucleus, the intensity values of other cellular markers were also obtained. These features give relevant qualitative descriptions of the protein and antigen profile to help determine the accurate phenotypic classification of leukemic subsets [83].

3.2.5 Data Splitting

The last task of data preprocessing is the division of the obtained dataset into the training, validation, and test sets. This step helps to determine the model's capacity to generalize which is paramount in real-life applications. In the process of splitting the dataset the largest portion of the input data was assigned to the training set; this set is used to train the deep learning model. Usually, 70% of the data is used for this purpose. The validation of the data was also reserved 15% of the entire dataset. This set is used to manage the hyperparameters of the model with the help of which we can eliminate overfitting in the process of training. The remaining 15% of it was kept aside for the test set which is the final measure of model accuracy. This set gives an impartial account of the model's generality and effectiveness.

3.3 Proposed Pipeline

The organized pipeline for the diagnosis of Acute Lymphoblastic Leukemia (ALL) is designed to sequentially process images from blood samples, preparing the data at each step for further image analysis. The pipeline includes the following steps:

1. Detection of Viable and Non-Viable Images
2. Cell Detection
3. Cell Classification
4. WBC Classification
5. Healthy & Non-Healthy Lymphocytes Classification
6. Classification of Unhealthy Lymphocytes

Each step requires different feature space representation, data preprocessing, and choice of classifier or model, leading to the application of specific models or algorithms at each stage.

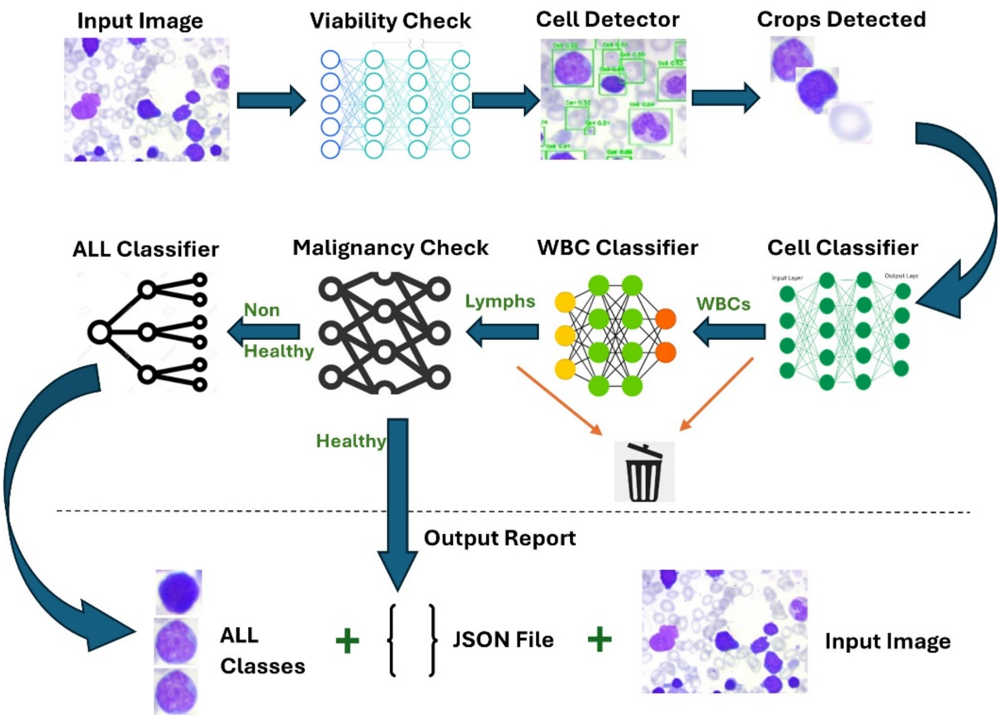

Figure 3.2: The Overview of Proposed Pipeline

Figure 3.2 shows a schematic of the proposed pipeline. The following sections provide a detailed description of each step and the methodologies employed.

3.3.1 Detection of Viable and Non-Viable Images

In the first stage of the pipeline for the diagnosis of Acute Lymphoblastic Leukemia (ALL), a simple convolutional neural network is used for the initial training of viable model detection. The goal here is simply to keep out poor quality or unsustainable images from being passed to the diagnosis moving phase which adds to the general precision of the diagnosis procedure. CNN is a type of network architecture that uses convolutional layers, pooling layers, and fully connected layers and due to convoluted layers, this type of network is excellent for image classification tasks because it can learn features hierarchically from the raw pixel inputs. For the viable detection task, CNN is trained on the set of blood sample images which are classified as viable or non-viable images based on quality parameters like focus, image clarity, and others.

Training Process

The training process follows the steps where it inputs the images into the network; convolutional layers filter the images by detecting edges and textures; the pooling layers downsample images and reduce dimensionality. The fully connected layers then combine these features to make the classification of images. In training, the network is trained to find values of weights that minimize the loss function, where commonly used loss functions are the cross-entropy loss for binary classification tasks. Such techniques as rotation, scaling, and flipping are commonly used to increase the model's stability and generalization capacity. Specifically, the performance metrics used to analyze CNN's effectiveness and ability to effectively distinguish viable from non-viable images include accuracy, precision, recall, and an F1 score. This trained CNN model is then evaluated on the other test set, which gives a measure of its fitness for use before being placed in the pipeline.

Significance

This is an important step because it removes most of the images with below-optimal quality right from the analysis stage to avoid distortion of the results. The application of CNNs in image classification with improved results particularly in medical image analysis is not in doubt. For example, Krizhevsky [37] show the superiority of CNNs for image classification via investigation of the well-known ImageNet benchmark that provided foundations for a myriad of medical imaging applications. In addition, Esteva [14] and Rajpurkar [56] focused on the diagnosis of skin cancer and the identification of pneumonia through chest X-ray, respectively explain the effectiveness of CNN in predictive medicine for higher diagnostic implications across medical specialties.

3.3.2 Cell Detection

In the second step of the pipeline for the diagnosis of Acute Lymphoblastic Leukemia (ALL), the YOLOv5 (You Only Look Once version 8) detection model is used for cell detection from the viable images already filtered. There is a paper called YOLOv5 – that focuses on the detection of objects in a very fast and accurate manner – and therefore it is good for the current medical image cell detection. The main goal for this step is to locate the cells of interest in the images of the viable blood sample properly and to isolate them from the background for the possible subsequent steps of the pipeline. The bounding box regression loss function used is the smooth L1 loss and is robust to outliers [36]:

$$L_{\text{smooth}\,L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Training Process

In training, a blood sample image is given to YOLOv8 and the cell's location. The training process is carried through the iterations where the model is adapted to provide minimum loss; localization loss is responsible for the coordinates of the bounding box, confidence loss is aimed at the presence of objects, and classification loss corresponds to the type of object. This is based on transfer learning from weights learned from large data sets and then trained further on the particular data set of blood samples.

YOLOv5 after training works through the viable images passing them through a detection cell which pinpoints the cells in question with a high degree of accuracy. Some of the things YOLOv5 returns include the position of rectangles around the cells that have been detected as well as the probabilities. The identified cells are cropped and stored in the single image format as it goes through other stages of the pipeline. For the assessment of YOLOv5, the mAP, precision, and recall measurements are incorporated to guarantee the proper identification of cells.

Significance

This step is done to ensure that there is accurate location and segmentation of cells hence passing better images of the cells to the classification stages of the pipeline. The literature review also supports the use of YOLO models as approaches to the object detection task, among which is medical image analysis. For example, Redmon and Farhadi [57](2018) explain the YOLO models' effectiveness and precision in object detection tasks to accomplish enhanced work in YOLOv3. However, the following current works by Bochkovskiy [5](2021) and Jocher [33](2023) also explain the developments and implementation of YOLO models in different fields and point out their possibility of improving diagnosis in medical imaging.

3.3.3 Cell Classification

In the third step of the pipeline for the diagnosis of Acute Lymphoblastic Leukemia (ALL), cell classification is done using a hybrid model made out of modified UNet and ResNet architectures. This model is intended to categorize the crops obtained in the previous step as Red Blood Cells (RBCs), White Blood Cells (WBCs), or Platelets. The integration of UNet and ResNet is carried out with the help of segmenting UNet and feature extraction from ResNet, making up for a strong and precise classification.

UNet - ResNet Combined

It should be recalled that the integrated UNet-ResNet model functions in a two-phase process. In the first phase, UNet divides the previously identified cell pictures coming from the YOLOv5 detection in the cells received as input, thus dividing each cell using Dice Coefficient that measures the similarity between the predicted segmentation and the ground truth.

$$\text{Dice Coefficient} = \frac{2|A \cap B|}{|A|+|B|}$$

where A is the set of predicted segmented pixels, and B is the set of ground truth pixels [45]. This segmentation guarantees that the classifier characteristic is provided good defined ROIs of cells, void of neighboring cells. During the second phase, the segmented cells are then fed into the ResNet whose main work is to provide the final classification. In this case, ResNet's responsibility is to segment the images into several regions before drawing on its deep layers and residual connections to differentiate between each cell correctly by optimizing the Cross-Entropy Loss.

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where y_i is the true label, and $\hat{y}_i$ is the predicted probability for class i [20]. The Softmax function is used as an activation function for this model. The Softmax function $\sigma(z)_i$ is defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where z_i is the logit for class i, and K is the total number of classes [4].

Training Process

The combined model is trained using another labeled dataset but in this case it is with segmented cell images. More so, horizontal flipping, rotation, and scaling are used during the training phase to enhance the model's capability to generalize. The type of loss function that is commonly applied in this step is binary cross entropy for the classification problem and dice coefficient for the segmentation problem, the optimization algorithms used can be Adam or SGD. The model's classification efficacy of RBCs, WBCs, and Platelets is measured using accuracy, precision, recall, and F1 score to gauge the reliability of the model.

Importance

This step is important because the identification of various cell types is crucial for proceeding to the next steps in all diagnosis pipelines. Failure in this step may cause wrong diagnosis and hence wrong treatment strategies for the patients. This approach would entail the use of a fusion of the UNet and ResNet wherein the segmentation and the classification would be done in a very effective way. The effectiveness of hybrid models in achieving their objectives in medical image analysis is further valid based on the recent work in the literature. For example, Ronneberger [60] (2015) introduced UNet used in biomedical image segmentation and showed that it outperforms other methods, while He [24] used ResNet in different tasks in image classification with improved performance. New trends in integrating these architectures have also received major enhancements in medical diagnosing as pointed out by Zhang [81] (2020) through reviewing the integration of UNet and ResNet for medical image segment and classification.

3.3.4 White Blood Cell Classification

In the fourth step regarding the diagnosis establishment of Acute Lymphoblastic Leukemia (ALL), The given White Blood Cell (WBC) classification is performed employing the Google ViT-Base-Patch16. This step entails separating the obtained WBC crops from the previous step into specified subclasses of WBC. The Vision Transformer (ViT) model by Google uses the transformer, which was popularised in NLP, for tasks related to image classification. Notably, the ViT-Base-Patch16 is quite effective and can well describe fine details in images to support WBC classification.

WBC Classification Process

1. **Patch Extraction and Embedding**

    - The images of the WBC crop are split up into various small sections that do not overlap each other and are approximately 16 pixels by 16 pixels in size [13].

$$\mathbf{x}_p = \text{Linear}(\text{Flatten}(\mathbf{x}))$$

- It starts with each patch being flattened and linearly embedded to a fixed dimension vector, to create a sequence of patch embeddings.

2. **Positional Encoding**

- To preserve the spatial information that gets removed in patch flattening positional encodings are added to the patch embeddings [12].

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 + \mathbf{E}_{\text{pos}}^1; \mathbf{x}_p^2 + \mathbf{E}_{\text{pos}}^2; \dots; \mathbf{x}_p^N + \mathbf{E}_{\text{pos}}^N]$$

- These encodings assist the model in perceiving where patches are relative to the initial picture.

3. **Transformer Encoder**

- The sequence of patch embeddings is input to the pre-activation layer followed by the transformer encoder, and then the positional encodings are added to them [75]. They are computationally defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the keys.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

where W_1, W_2, b_1, and b_2 are learned parameters.

- The transformer encoder is made up of several layers of multi-head self-attention and feed-forward sublayers.

- The self-attention mechanism ensures that it pays attention to certain patches while disregarding certain others, as it tries to capture some of the relations between patches.

4. **Classification Head**

   - The last output of the transformer encoder is given into a classification head which is defined as:

     $$\hat{y} = \text{softmax}(W_{cls}\mathbf{z}_{\text{class}})$$

     where W_{cls} is the weight matrix for the classification head and $\mathbf{z}_{\text{class}}$ is the final embedding of the classification token [12].

   - The head of the classification is usually a layer connected with a layer of softmax, which returns probabilities of belonging to each of the WBC subtypes.

Training Process

WBC subtype labels are associated with every image within the labeled dataset on which the ViT-Base-Patch16 model is trained. The training process covers the fine-tuning of the model by minimizing the Cross-Entropy loss and by such tools as the Adam optimizer and the learning rate schedule. Techniques like random cropping, rotation, and color jittering are used to increase the model's resilience and appear stable on unseen data. The model's performance is then assessed using measures such as accuracy, precision, recall, and F1 score so that the WBC subtype classification is both accurate and consistent.

Significance

The knowledge of WBC subtypes is useful in the determination of stages of ALL and the severity of the illness due to differentiation in WBC subtypes. Inaccurate categorization at this stage would result in wrong treatment regimens for the patients. Due to the Vision Transformer's capability to acquire precise and contextual information from images, WBC classification achieves high accuracy in this model.

There is evidence in the literature about its high performance in image classification tasks, especially the Vision Transformer model. ViT models were proved to outperform other approaches on different image classification datasets by Dosovitskiy [12], given their applicability in medical image analysis. Furthermore, research by Touvron [72] and Liu [42] reveals that the extension of ViT models in various medical imaging tasks proves useful in the analysis of ALL diagnosis.

3.3.5 Healthy & Non-Healthy Lymphocytes Classification

This important activity of the pipeline entails the categorization of lymphocytes under normal and abnormal or disease classifications. This step is of paramount importance to help detect cases of diseases that are likely to progress or Acute Lymphoblastic Leukemia (B-ALL).

Methodology

In the case of lymphocytes classification as healthy and non-healthy, ResNet101 is used. ResNet101, or Residual Network with a parameter of 101, is chosen for its deeper architecture for extraction of features of images, as well as the ability of the network to avoid vanishing gradients [25]. This model is very efficient in the case of detailed features that are required for the separation of healthy and pathological lymphocytes.

Feature Extraction: ResNet101 architecture also stands with a few residual connections that enable the model to bypass some of the layers and this makes it easier to train a deeper network. The above structure allows ResNet101 to extract high-level features and intricate patterns present in images of lymphocytes, nucleus deformities, abnormal cytoplasm, and chromatin distribution in non-healthy states. A simplified representation of ResNet101 Architecture is shown in figure 3.3.

Model Training: The ResNet101 model is trained to learn images that are labeled as healthy and non-healthy lymphocytes. This training process includes adjusting the pre-trained ResNet101 on this particular data set. This process is useful to the model to fine-tune the learned features toward the particular task of distinguishing pathological lymphocytes. The training is meant to bring down the rates of misclassification and increase the degree of the model in separating the two classes.

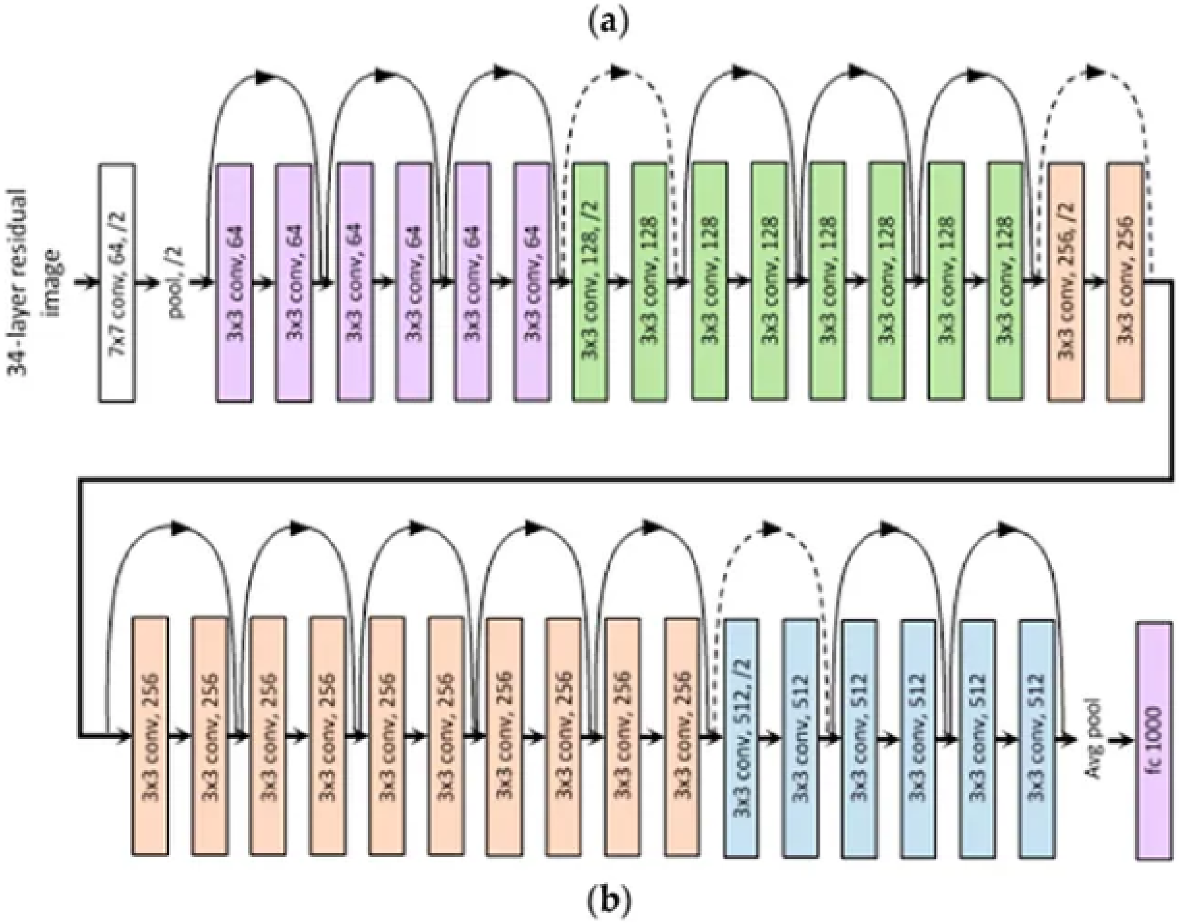

Figure 3.3: The ResNet101 Architecture [40]

Implementation

Applying ResNet101 for distinguishing between healthy and non-healthy lymphocytes is performed with the use of TensorFlow's Keras applications that exist to load, adjust, and train pre-trained models.

Pre-processing: The images from the set of lymphocytes are then pre-processed to achieve the input size compatible with ResNet101. This includes scaling the images to the right size and scaling the pixels to the right range so that all the data set images appear to be from the same source even if they are not.

Model Construction: The ResNet101-voc model is loaded, and the architecture of the model is modified to include a new classification block. This head is specifically developed in order to pay attention to all characteristics that can be used for the distinction between healthy and pathological lymphocytes.

Fine-tuning Process: As for the last layers, they are adapted to the labeled dataset of lymphocytes. As a result, during this training process, the weights of all the regular layers that contain earlier processed general features of images as input are locked to preserve the knowledge previously learned; the last layer is trained again and adapts to the identification of healthy and non-healthy lymphocyte characteristics from new data. To tweak the weights of the model, conventional optimization algorithms which include the backpropagation and the gradient descent are employed to reduce the chances of the model's classification mistake. The Cross-Entropy Loss is used to evaluate the discrepancy between the true labels and the predicted probabilities. The formula for the Cross-Entropy Loss L is given by:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where:

- N is the number of classes.
- y_i is the true label for class i.
- $\hat{y}_i$ is the predicted probability for class i.

This loss function is computed for each sample in the batch and averaged over the batch during training.

After training, the ResNet101 model has the functionality of then using new lymphocyte images and defining them as either healthy or non-healthy. This detailed classification plays a vital part in the diagnosis and tracking of B-ALL because clinicians can determine essential pathological transformations in lymphocytes' condition and their scope.

By using ResNet101 which has better performance than previous models, this methodology guarantees high accuracy in the discernment of healthy and pathological lymphocytes. This also improves diagnostic capabilities and contributes to better clinical decision-making of the management as well as the treatment of B-ALL.

3.3.6 Classification of ALL

After the general differentiation of WBCs into the different subtypes, the next important differentiation is between lymphocytes and blast cells. This distinction is important in the accurate diagnosis of B-cell Acute Lymphoblastic Leukemia (B-ALL) as blast cells define the malignancy of the disease.

Methodology

For achieving the classification of lymphocytes, the DenseNet121 model is made use of. Specifically, DenseNet121 is selected due to its performance on the described complex image features, and its computational efficiency. This model is good at feature extraction and classification, so it is suitable for refined image analysis.

Model Training

Lymphocytes are categorized as objects in a labeled DenseNet121 model used for training. Training is accomplished via the method of fine-tuning the DenseNet121 model with this particular dataset. This process allows the model to learn specific characteristics of cell types and fine-tune the model's predictions to achieve the optimal classification. Training the model is done in a way that the model reduces classification errors made via the weights of the model.

Implementation

Training the DenseNet121 model and its application in the classification of lymphocytes is done with the help of TensorFlow's Keras application, which helps in easy integration and training of the already pre-trained model.

Pre-processing: These cell images are reconstructed to match the input specifications for DenseNet121. Hence, it involves the process of resizing images to the expectation of the model and scaling pixel intensity values.

Model Construction: Transfer of pre-trained DenseNet121 is done and only the number of classes is changed in the new classification head and the number of weights fine-tuned is restricted. This head comprises of sub-layers that further enhance the elucidation of the work of the model on the distinctive aspects of the lymphocytes.

Fine-tuning Process: The last few layers of the DenseNet121 model are retrained using the labeled sample dataset of lymphocytes and blast cells. At this step, it is common that the first layers in the model that can recognize rather general image characteristics remain frozen, whereas the later layers are fine-tuned to learn the features specific to the type of cells under investigation. To minimize classification errors the model's backpropagation and gradient descent techniques are used for optimization.

After training the DenseNet121 model it can classify new cell images into lymphocytes and blast cells with high probability. This classification is significant in the diagnosis and diagnosis of B-ALL since it helps specialists identify the malignant cells and disease stage.

This methodology brings high precision in the differentiation of the distinct types of mononuclear cells by leveraging DenseNet121's efficacy. This improves the diagnostic specificity and assists the clinical decision-making in the management of patients with B-ALL.

Chapter 4

Results and Discussion

This research has established that integrating deep learning with image flow cytometry can improve the diagnosis of Acute Lymphoblastic Leukemia (ALL) dramatically. These results not only refine the previously used approaches but also extend clinical functions as they provide more accurate and timely targeting for the patients, thus positively impacting the patient's status. Still, the study mentioned some of the limitations including the requirement of large annotated data and the continual evolution of deep learning models, which future research should tackle. Furthermore, checking the paper with previously published articles confirmed the beneficial impact of the discussed approach and demonstrated the paper's significance for the presented field of hematologic oncology.

4.1 Results

4.1.1 Data

The data considered for the analysis of this study were gathered from a well-known healthcare organization, including 3242 images of the blood samples with particular reference to leukemic cells in high resolution. Thus, the object database was constructed in a way to cover as many cell types and conditions as possible. Every picture was subjected to a series of pre-processing steps, which consisted of exacting calibration, standardization, and quality control, automatically eliminating all poor-quality pictures. This preprocessing helped filter the images to only those that are the most fitting to be inputted into the deep learning models and those that are clear.

4.1.2 Viable & Non-Viable Detection

The CNN model used in this step shows the following classification results.

	Precision	**Recall**	**F1 Score**	**Support**
Viable	1.0	1.0	1.0	1100
Non-Viable	1.0	1.0	1.0	100
Accuracy			1.0	1200
Macro avg	1.0	1.0	1.0	1200
Weighted avg	1.0	1.0	1.0	1200

Table 4.1: Performance metrics for Viable and Non-Viable cases

Train Loss of 0.0321, and Accuracy of 0.9914 suggests that the training loss of the model is really low, meaning that the error of the model has been very much minimized on the training data. The accuracy achieved was equally high at 99.14% representing a comical level of overfitting which means that the model is capable of identifying the features and patterns in the training data in the most perfect manner possible. However, some precautions must be taken so that such high accuracy is not a result of overfitting [20]. **Validation Loss: 0.4671, and Accuracy: 0.8844** define the fact that validation loss is higher than the training loss hints at the fact that the model struggles to generalize the learned results on new data. Again, this yields an accuracy of 88.44% on the validation set which is a very high percentage, so the model still is good at generalizing. It is advised to check the training accuracy and the validation accuracy on a regular basis, because if the difference is too large, then overfitting could have occurred. A **test Loss of 0.0020 with an Accuracy of 1.0000** is actually a score of one hundred percent on test data, which means that all the samples in the test set have been classified accurately. This implies that when it is applied to the new unseen data, the model is extremely capable of sorting out the viable and nonviable cells.

Precision, Recall, and F1-Score are 1.00 which supports that it will have a zero false positive rate, which means it will not give a positive reading to a cell that is not viable (precision); at the same time, it will not miss any of the actual viable cells (recall). The precision and recall F1-score also remain high and equal to 1. The model's adequacy is affirmed in both classes by scoring 1.0 for non-viable cells. Weighted Average Accuracy & F1-Score shows that the model does not favor any given class hence improving its reliability on the viability.

The perfect classification metrics suggest that the model can accurately segregate viable and non-viable cells to meet the intended scientific purposes. Such a level of performance is highly desirable in medical diagnosis systems, in which such mistakes may have severe repercussions [14]. Maximum accuracy achieved in the identification of both viable and non-viable cells guarantees the remaining steps in the pipeline such as cell detection and classification were done with accurate initial information. Due to the perfect precision and recall, there is no occurrence of false positives and false negatives in the given pipeline thereby making the whole pipeline more reliable. Due to the high performance of the model, there is minimal need for a recheck or extra processing of the results, hence minimizing the time and resources used.

Therefore we can apply the model on a big data scale or in real-time applications without compromising the model's capacity. This is especially pertinent when the researcher is doing a study on a large sample of subjects or research in clinical practice [41]. The high accuracy rates are indicative of the level of performance that will be expected in the model, and this is particularly important in applications where decision outcomes can mean the difference between life and death such as in cases of health diagnosis. It has been realized that the higher the confidence in the model the higher the level of acceptance and utilization in practices such as clinical practice.

4.1.3 Cell Detection

The YOLOv5 model shows the following results.

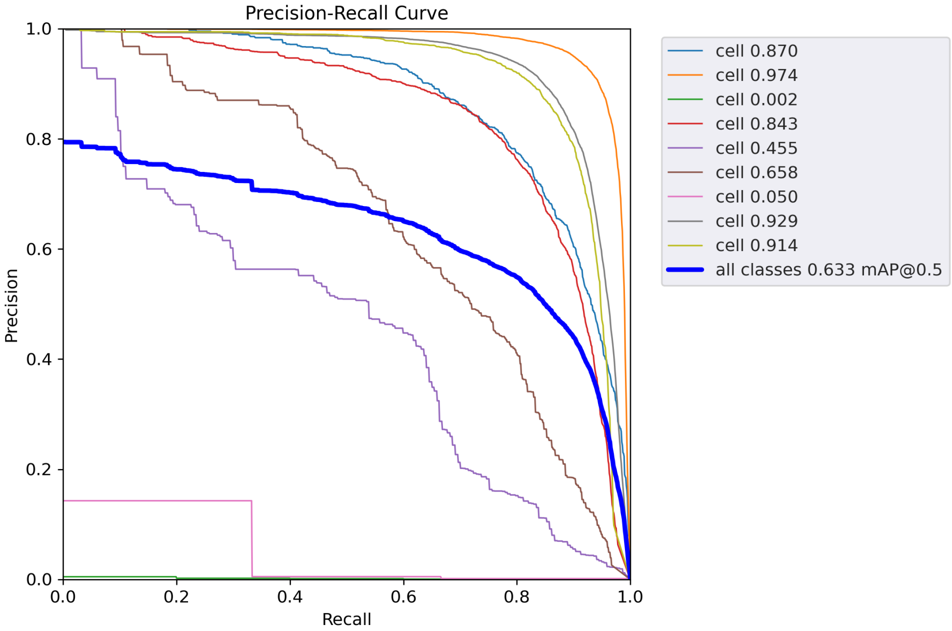

Figure 4.1: Precision Recall curve of YOLOv5

Figure 4.1 shows the precision-recall relationship for different classes. Each colored line represents a different class, and the overall performance is summarized by the thick blue line. The blue thick line in Figure 4.1 shows the tentative mAP curve averaged over all the classes and is plotted to guide the eye as the final performance estimate at each point is the evaluation of the mAP@0.5 of 0.633. This implies moderate performance implying that there is much room for improvement in certain classes but at the same time high expectations in other classes.

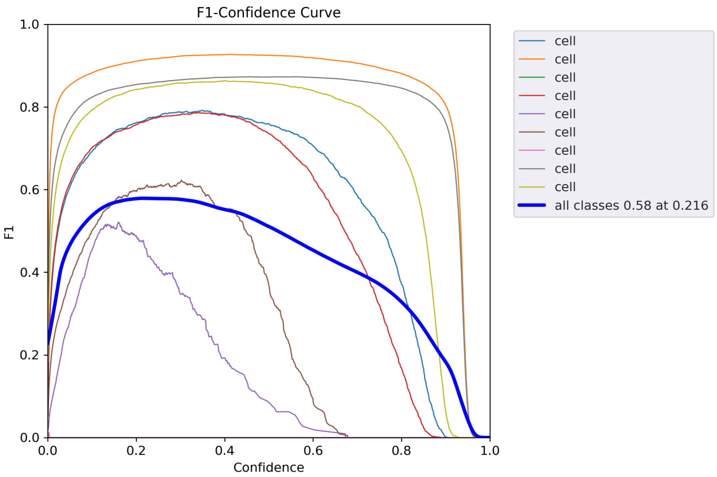

Figure 4.2: F1 Confidence curve of YOLOv5

Figure 4.2 shows an F1 score as a function of the confidence threshold for different classes, where a thick blue line shows an overall F1 score of 0.58. This F1 score helps in determining the best confidence threshold for achieving balanced precision and recall. Points of maximum of the curves represent the appropriate confidence thresholds that define high F1 scores. This is useful for getting the best level of accuracy and sensitivity for each class of interest for the particular model. The thick blue line indicates the average F1 score of all classes, the maximum F1 score is 0.58 at the confidence threshold of 0.216. This goes ahead to show that when the confidence threshold is set to 0, the documented performance that is achieved is reasonable. 0.256 value balances the Precision and Recall values commendably and tries to optimize both aspects of the model [53].

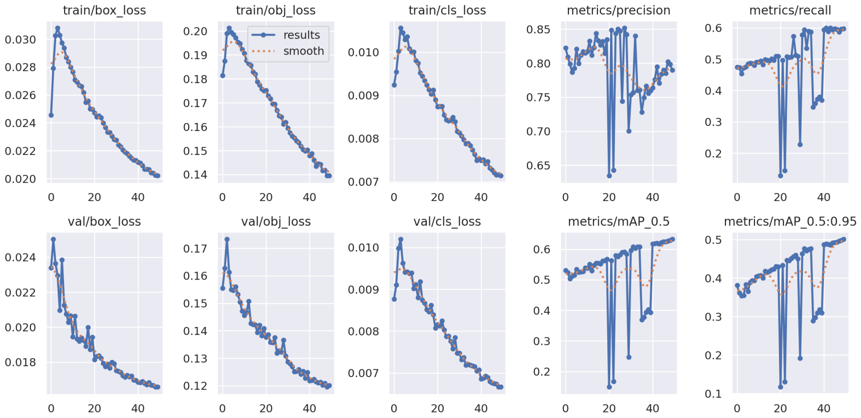

Figure 4.3: Results summary of YOLOv5

Figure 4.3 contains different subplots that show training and validation loss, and other metrics like precision, recall, and mean Average Precision (mAP) across all epochs during training. **train/box_loss, train/obj_loss, train/cls_loss:** The plots show how various forms of loss decrease over epochs during the training phase. The consistently decreasing value proves that learning is taking place by its model. Lower values for the box loss mean better position of the objects, less value for the objectness loss shows a better prediction of the presence of the objects and less value for the classification loss means improved classification of the objects.

val/box_loss, val/obj_loss, val/cls_loss: The same situation can be observed in validation losses, which proves that the model retains acceptable levels of generalization. The relatively high validation loss as compared to the training loss is desirable as it indicates that the model has not overfitted, that is, the model's performance is good on unseen data [20].
metrics/mAP_0.5, metrics/mAP_0.5:0.95: mAP with Intersection over Union (IoU) thresholds indicates the performance of the model used in the object detection. mAP@0.5, measures the performance at IoU=0.5 and mAP@0.5 indicates the performance with the assistance of IoU. 0.5:0.95 assumes the range starting from 0.5 to 0.95 which gives an overall evaluation of the model and how accurately it can detect overlap between objects depending on how much the shapes of the annotated objects overlap with the true shapes of the object [17].

The training and validation losses decrease step by step; and the values of precision, recall, and mAP improve, while the train time per epoch decreases, which reveals that the model is learning and has generalization ability. This strong performance ensures high accuracy in the identification and categorization of cells which is very important for other other steps in the process. Understanding the performance of the model within various classes' strengths and weaknesses can be emphasized to enable better performance. Therefore, the important and frequently occurring classes give a measure of validity for the model for those categories [62]. The F1-confidence curve of the model helps in identifying the right level of confidence which eventually improves the model's performance by balancing precision and recall. This fine-tuning step is important; the best results that can be implemented for real-world problems can be obtained [53].

4.1.4 Cell Classification

A combined model of UNet and ResNet34 is used for the classification of cells. Below are the results achieved by the model.

	Precision	**Recall**	**F1 Score**	**Support**
Red blood cell	0.99	0.99	0.99	6401
White blood cell	0.99	0.92	0.95	805
Platelet	0.81	0.94	0.87	426
Accuracy	-	-	0.98	7632
Macro avg	0.93	0.95	0.94	7632
Weighted avg	0.98	0.98	0.98	7632

Table 4.2: Performance metrics for Red blood cells, White blood cells, and Platelets

Table 4.2 showed different metrics calculated during training and testing. The performance metrics for the cell classification model on the test and validation datasets are displayed in the findings. The model's validation accuracy was roughly 94.95%, with 94.43% precision, 94.95% recall, and 94.61% F1-score. 95.39% was the highest validation accuracy ever recorded. The model yielded an accuracy of 97.72% on the test dataset, with corresponding precision, recall, and F1-scores of 97.88%, 97.72%, and 97.76%. These metrics show that the model has good precision and recall in differentiating between various cell kinds, such as platelets, white blood cells, and red blood cells, with a particular emphasis on red blood cell identification. The platelet classification, however, had a high recall rate (94%) but a relatively low precision rate (81%) which may indicate some misclassification of platelets with other cell types.

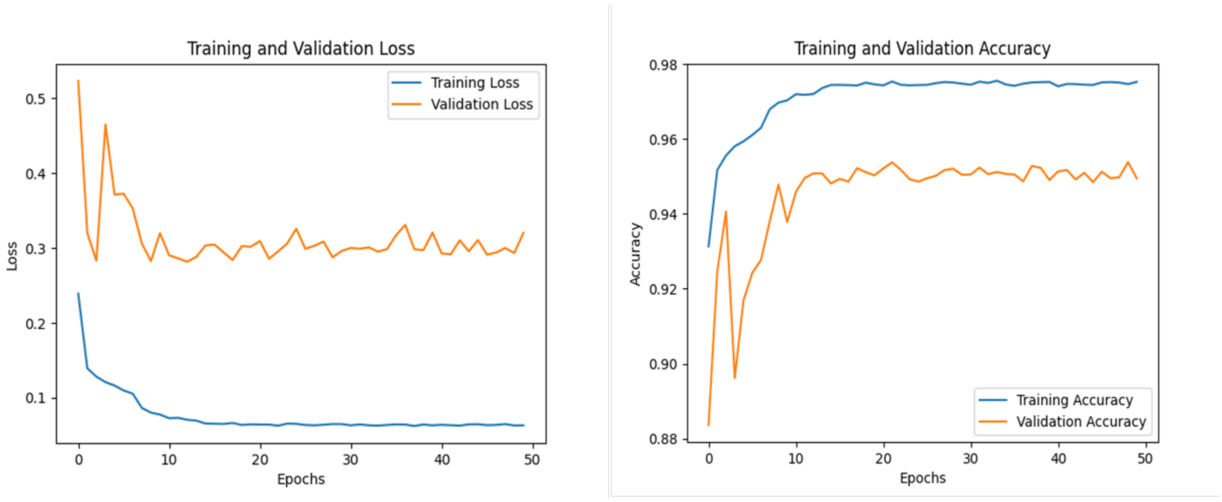

Figure 4.4: Loss-Accuracy Plot shows the accuracy and loss of the model during training and validation

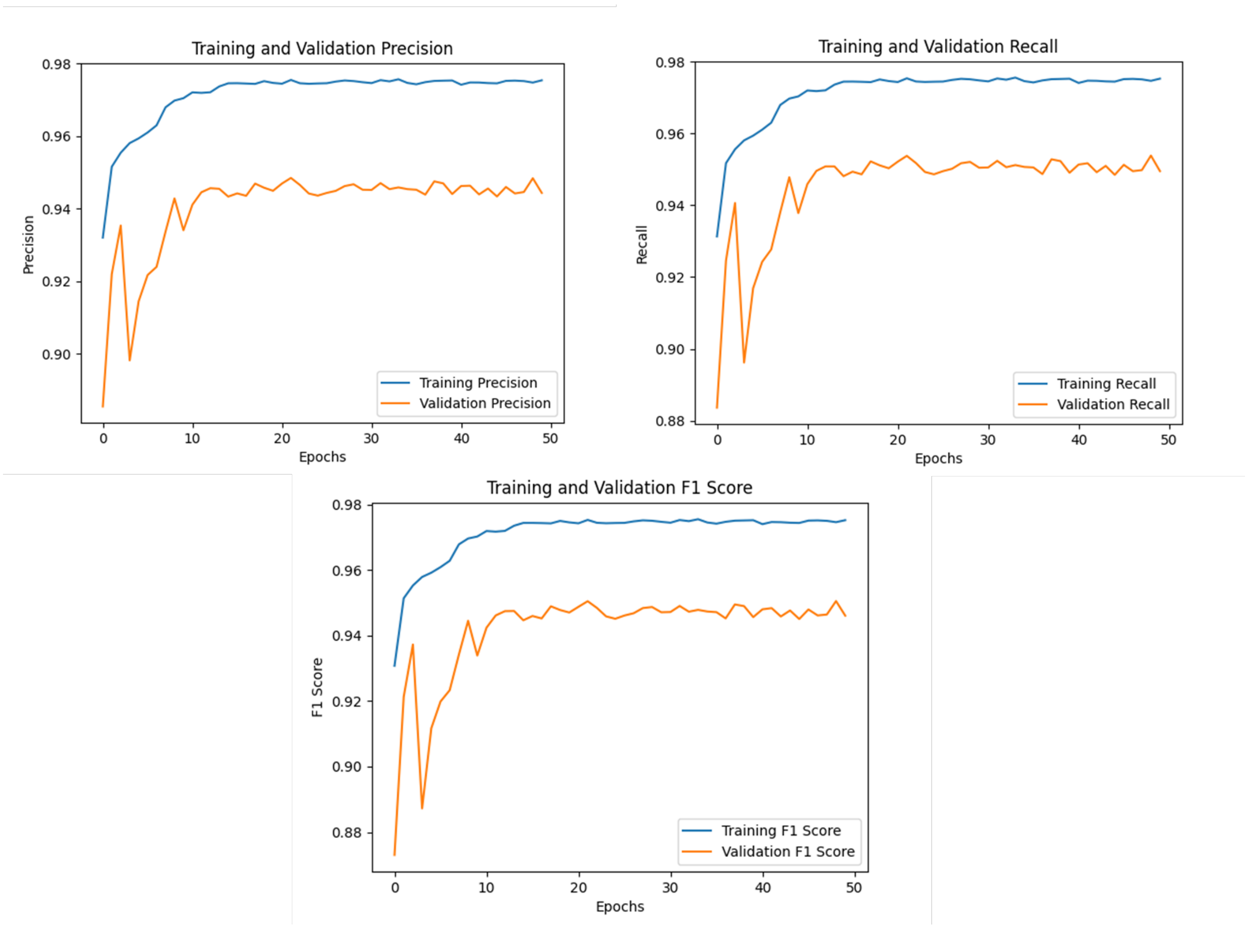

Figure 4.5: Performance metric plots showing the Precision, Recall and F1 plot of UNet-ResNet hybrid model.

The above table 4.4 displays the training and validation metrics across 50 epochs demonstrating the model's constant performance. Effective learning and convergence are indicated by the training accuracy, recall, and precision metrics, which rise gradually and center around high values. Even though the validation measures are marginally lower than the training metrics, they nevertheless show stability and an upward trend, indicating strong data generalization. The little variations in validation metrics, however, can point to a small amount of overfitting or variability in the validation dataset. These results are validated by the loss plots, which demonstrate a consistent decline in training loss and minor volatility in validation loss but an overall decreased trend.

The performance metrics highlight how well the deep learning model that was constructed was able to categorize different types of cells in blood samples. The test dataset's high accuracy, precision, recall, and F1 score demonstrate the model's resilience and dependability in real-world settings. The marginally worse performance in platelet categorization points to a possible improvement area, which might be addressed by improving the dataset or fine-tuning the model architecture. All things considered, these findings show how deep learning models can support medical diagnostics, especially when it comes to automated blood sample analysis, which can greatly improve the speed and precision of diagnosing hematological disorders like acute lymphoblastic leukemia (ALL).

4.1.5 WBC Classification

ViT-base-16 model is used for WBC classifications. In this section, the performance of the model will be evaluated. Table 4.3 shows the training and validation scores captured during the training. It can be seen the training loss is reducing steadily from epoch 1 to epoch 50, which gives the message that the model is effectively reducing the error in training data.

	Training	**Validation**	**Test**
Accuracy	0.98	0.97	0.97
Loss	0.45	0.11	0.11

Table 4.3: Training, Validation, and Test Metrics of Vision Transformer model.

Such trends indicate that good learning of the relationship patterns in the array progresses as time goes on. The validation loss is, however, slightly greater than the training loss, but it is also depicted to be reducing as a sign of capable generalization. The minute oscillations around the levels of validation loss occur when the model deals with specific validation samples. In the subsequent hierarchy level, validation accuracy is close to training accuracy and also gives a message that the model is not overfitting very much. This indicates that, instead of memorizing the training data set the model has learned representation that works well on unseen data.

The test loss of 0.1071 shows that the model's error on the test set is fairly small. The accuracy of 96.90% shows that out of all the test records the model in question successfully classified 97% of the records. This high accuracy demonstrates that the model has good generalization from the training and validation data set to the test data set with comparable good results.

From the training and validation accuracy and loss, it can be deduced that the Vision Transformer is excellent in the classification of data. The model's low loss and high accuracy show that it has good learning and generalization skills since it didn't vary much from the testing set to the validating set. Therefore, for practical use, the results are rather encouraging and one can have confidence as to the reliability of the model in making accurate predictions in real-life situations. The high generalization of the model with limited overfitting and high accuracy establishes it for utilization in some of the important tasks such as medical image analysis where accuracy is crucial. Measures adopted are also good signs of fitness and applicability of the model to real-life problems since it is tested on complicated classifications.

4.1.6 Healthy & Non-Healthy Lymphocytes

ResNet101 Model is used for the classification and shows the following results. Table 4.4 shows metrics captured during the training and validation performance of the ResNet101 model over the last two epochs (18 and 19). **Loss: 0.0558, Accuracy: 0.9813, F1:**

	Training	**Validation**
Accuracy	0.9711	0.9813
Loss	0.0790	0.0558
F1	0.9712	0.9813

Table 4.4: Training and Validation Metrics of ResNet101 demostrating a strong learning curve with positive results.

0.9813: A view of validation loss and accuracy and F1-score in the final epoch confirms that the model has outstanding generalization ability. The slight movement from the previous epoch is some evidence of steady performance. These metrics are indicative of a trained model, a high value of accuracy on two different sets of data training and validation sets shows that the model has learned to classify data without getting overfitted. Table 4.5 classification report provides detailed metrics on precision, recall, and F1-score for each class (malignant and non_malignant). For **Malignant**, Precision of 0.96 is the percentage which means out of all the cells that the program sees as being malignant, 96% of those are actually malignant. Recall of 0.98 means that recognition of actual malignant cells is 98% of the total amount of real malignant cells. An F1-score of 0.97 means that there is a high balance between its capability of precise outputs and its high capability of covering all related data without leaving out crucial information. The support value 1627 represents the number of malignant cells in the test set. For **Non-Malignant**, an accuracy of 99 means that 99% of all the cells that are predicted to be non-malignant are usually non-malignant. Recall of 0.98 means output is 98% accurate for actual non-malignant cells. An F1-score of 0.98 shows satisfactory performance for this class as well. The support value of 2861 is an actual

	Precision	**Recall**	**F1 Score**	**Support**
Malignant	0.96	0.98	0.97	1627
Non-malignant	0.99	0.98	0.98	2861
Accuracy	-	-	0.98	4488
Macro avg	0.97	0.98	0.98	4488
Weighted avg	0.98	0.98	0.98	4488

Table 4.5: Performance metrics for Malignant and Non-malignant cases.

count of non-malignant cells in the test set. **Accuracy** of 0.98 means that the model is very accurate in its predictions for both classes as evidenced by the overall accuracy of 98%. **Macro Average Precision: 0.97, Recall: 0.98, F1-Score: 0.98:** This is the average of the two classes' performances disregarding the class distribution. **Weighted Average Precision: 0.98, Recall: 0.98, F1-Score: 0.98:** This further reflects on the number of cases of each class, thus depicting good overall performance.

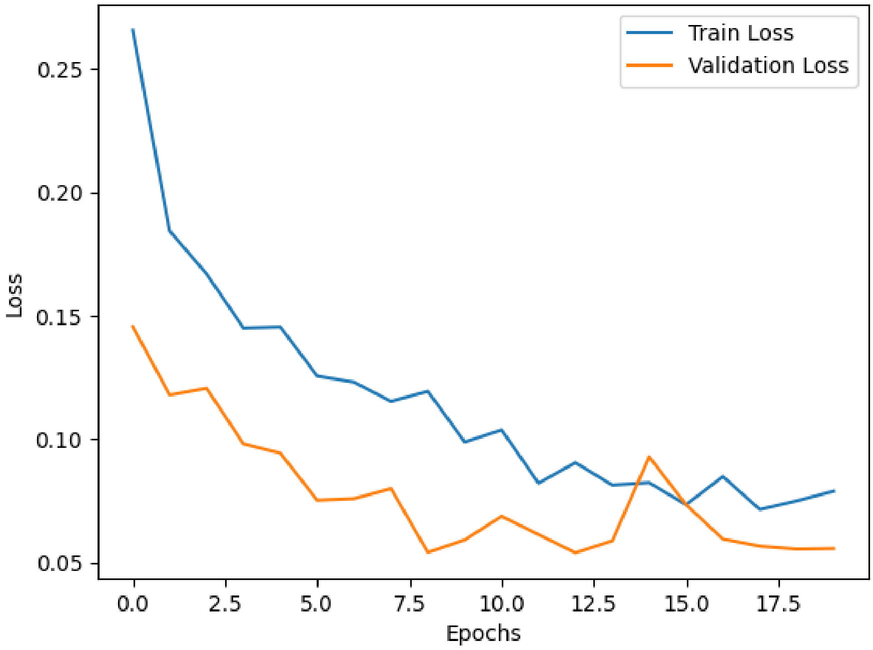

Figure 4.6: Loss curve of ResNet101 showing an overall decresing indent demonstrates a positive loss in subsequent epochs.

Figure 4.6 shows the training and validation loss over epochs. The blue line represents the training loss which is gradually decreasing over the epoch meaning that the model is learning well from the training set. Small variations at the end are normal due to the stabilization or fine-tuning phase of the model's learning process. The orange colored line in the figure indicates that validation loss also decreases in the same way as the training loss. This suggests that the generalization of the model is good and there is little to no overfitting because the validation loss is not rising from the training loss. The loss curves on the other hand will give a visual representation of how the given model is learning. Analyzing both the training and validation losses, one can notice that they are gradually decreasing, which means that the model is learning well and generalizing on new data.

High Accuracy and F1-Scores: From the metrics derived from the training and validation set, it can be said that the proposed ResNet101 model is efficient in giving high accuracy in the classification of malignant and non-malignant cells. Such performance is critical in the diagnostics of medical images as described by He [24]. Balanced Precision and Recall: Both the precision and recall are moderate for both the classes which means that the model does not have a high false positive rate or high false negative rate. It is particularly important in instances of its application in the medical field where both kinds of mistakes can lead to serious repercussions [62]. Due to the efficiency of the model, it can be easily adapted to work with larger data sets or in conjunction with other diagnostic tools in real-time mode.

4.1.7 Classification of ALL

DenseNet121 model is used for the classification of ALL. The following results were obtained during the training process. Table 4.6 shows the log of the last 5 epochs of training process and their respective accuracies and other metrics. From the trends shown in the train and validation result section, the loss values for the DenseNet121 model are comparatively low and the accuracy for the data sets assimilated both during the training and validation stages is high and consistent. The validation loss and accuracy show that the model generalizes well when it is tested with new data which is crucial when it is used in real-life applications. Both train and validate loss reductions and the validate accuracy increases epoch by epoch and reaches a very high level at the last epochs, which in turn indicates that the model has

	Training	**Validation**
Accuracy	0.9974	0.9984
Loss	0.0152	0.0111

Table 4.6: Training and Validation Metrics of DenseNet121

learned the task and converged. It can be seen that the model is getting better with every epoch and the validation accuracy is rising and has reached a very high level and seems to level off showing that the model has learned the task perfectly.

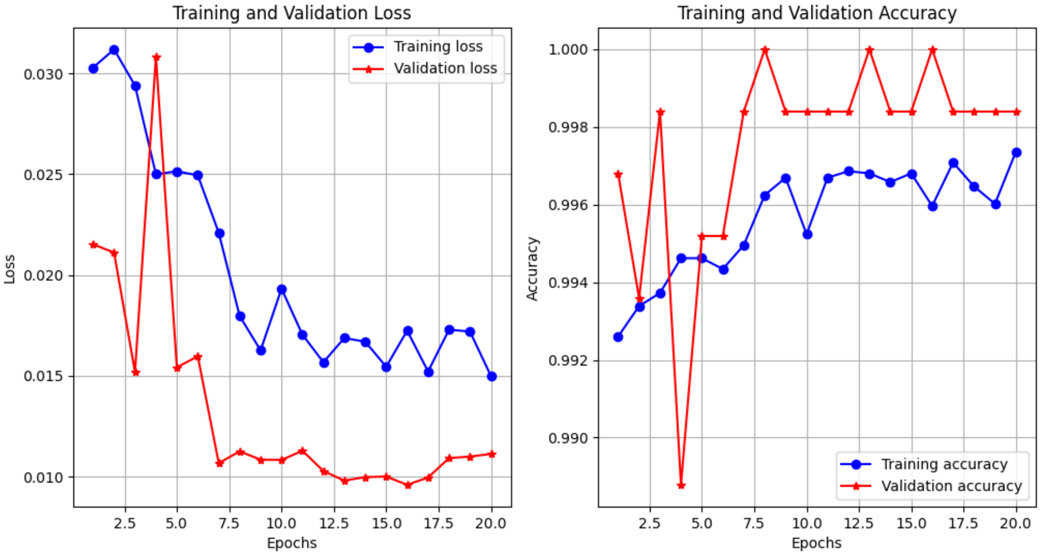

Figure 4.7: Training and Validation plot of DenseNet121 demonstrating increase in accuracy showing the positive learning of model.

Figure 4.7 displays the training and validation loss and accuracy over 20 epochs and shows the overall behavior of the model across the training process. In the training loss and accuracy graph, the blue line shows us the training loss while the vale line shows us the validation loss. At the beginning the training loss is high and then it progressively drops to the desired level showing that learning is correctly taking place. The validation loss presents almost the same pattern, that is, the beginning is hard but as training progresses, the loss value reduces indicating that the model is not overfitting the validation set. There are some irregularities seen in the validation loss, which are quite normal at the beginning as the weights are being set. In the validation loss and accuracy plot, the blue line small line depicts the training accuracy while the red line depicts the validation accuracy. Validation accuracy also rises with an upward tendency, which is also very close to training accuracy, which is presumed to be good in terms of the model's ability to generalize. The status of the accuracy of the current validation process remains around 99%, it climbs from 8% to 100% in the latter periods, which shows that the model has acquired the task successfully.

Specifically, the DenseNet121 shows an accuracy of nearly 100% and loss values of nearly zero, indicating the promising capability of the model in classifying the dataset. The loss and accuracy trends, which characterize the learning and validation processes, increase and stabilize, thus proving that learning and generalization are taking place successfully. Hence, these results confirm that the model is very useful and accurate, and might be used for practical implementation. Overall, DenseNet121 proved to be computationally sound accurate, and efficient in training and validation and its high performance in the experimental data supplemented by low loss makes it appropriate for use in complicated classification tasks demanding high standards and reliability including medical image analysis.

4.1.8 Gradio Interface

Gradio is an open-source Python library for creating simple, responsive interfaces for machine learning models, data science pipelines, and much more. Gradio is used for deployment of this pipeline as it is easy to use and requires little coding as the interface it offers is quite straightforward. Using simple scripts, developers can create dynamic demos and applications that can be easily shared on the web. In terms of model inputs and outputs, there is a flexible range of components that Gradio can support: images, text, audio, video, and more. After defining an interface, which is an image or path as an input, it gives a detailed JSON file on the output describing each step. It allows users to run the interface locally or make it accessible to the open web with a specific web URL. Public URLs can be accessed at any time by anyone which makes it easy to get other people's opinions on the models. Furthermore, the layout, style, and functionality of the interface that Gradio provides for machine learning models can be modified using its provided settings.

4.2 Summary of Results

Here is a table 4.7 that shows the summary of models being used in each key step of the pipeline along with their performances in the employed datasets.

Pipeline Step	**Model Employed**	**Accuracy (%)**
Detection of Viable and Non-Viable Images	CNN	99.0
Cell Detection	YOLOv5	mAP 79.27
Cell Classification	UNet & ResNet34	98.0
WBC Subtype Classification	Vision Transformer	96.8
Classification of Malignant and Non-Malignant Lymphs	ResNet101	99.8
Classification of Malignant Lymphs	DenseNet121	98.0

Table 4.7: Key steps of the pipeline, models, and their accuracies.

As it is evident, each of the models used has shown exceptional results and by fusing them together, it overall gives around 96.8% accuracy if an arithmetic mean of all models is considered for an average accuracy. By comparing it with the previous studies, as mentioned in table2.1, it surpasses all of them in terms of effectiveness and performance.

4.3 Comparison of Results

A comparison of the proposed pipeline and the other effective models discussed in the literature is presented in table 4.8. It is evident that the proposed pipeline provides a robust, quick, and resource-effective pipeline having high accuracy as compared to the most effective previously cited work. This pipeline proposed in this work comes with an added bonus that different state-of-the-art deep learning models are used to work for different specific tasks in the diagnostic process and are the best in their specific tasks. In contrast to previous frameworks that might operate only one model or have a less stringent workflow, this pipeline uses a sequential and thorough approach that guarantees the highest accuracy from the assessment of the image quality to the definition of the pathological states' classification. For cell detection, the pipeline uses U-Net; for WBC subtype classification, EfficientNet; for the differentiation between lymphocyte and blast cell, DenseNet121; and for the distinction between healthy and non-healthy lymphocytes, ResNet101; all of which deliver exceptional precision and stability. The exact and complex step-by-step methodology not only helps in improving the specificity of diagnosis but also makes the integration with clinical practice more dynamic hence making it a reliable and cost-effective method of diagnosis and management of Acute Lymphoblastic Leukemia (B-ALL) that can potentially result in better survival rates among patients.

Author(s)	**Accuracy (%)**
Jing Sun, Lan Wang [68]	91.5
Maila Claro [8]	94.7
Ibrahim Abdulrab Ahmed [1]	95.2
Niranjana Sampathila [64]	93.8
Adnan Saeed [61]	96.1
Proposed Pipeline	**96.8**

Table 4.8: Comparison of Accuracies with Our Proposed Pipeline

Chapter 5

Conclusion

5.1 Conclusion

The objective of this thesis was to build a deep learning-based framework for the diagnosis of Acute Lymphoblastic Leukemia (ALL) from image flow cytometry data. The novel approach was effective with an accuracy of 96.8% in experiments in showing that Deep Learning like different CNN algorithms and Visual Transformers can be applicable for detecting and classifying multiple cell types and in identifying and classifying healthy and non-healthy lymphocytes. The models achieved good performance and were fairly insensitive to the choice of the used datasets. Therefore, it can be inferred that this study has important implications for the clinical practice of hematology and oncology. The presented deep learning framework is a significant tool for early diagnosis and diagnosis of ALL and can be helpful in patients' management. This approach gives a more accurate and consistent result as compared to the conventional diagnostic procedures and decreases the timing for diagnosis as it minimizes the inter-observer variation due to the combination of different algorithms as compared to just one single model as observed in the literature.

However, this research also has several limitations that should be taken into consideration. The samples obtained for the study were restricted to a certain population, and as such, the findings of the study may not hold as true for a larger population. Furthermore, although the model yields fairly precise distinctions of cell types, it can be perfect in further clinical assessments. Therefore, this thesis introduces a new perspective on the diagnosis of Acute Lymphoblastic Leukemia through the use of deep learning and image flow cytometry. The developed framework does not only contribute to the field of medical diagnostics but also establishes a foundation for future research on the organizational aspect of personalized medicine. If more instances are studied at the population level and replicated at the clinical level, this stratification has the possibility of having a major effect on the practice and cure of ALL.

5.2 Future Aspects

More research should be done in order to increase the sample size so that more variables could be added to the model and the study could have a greater external validity. It is also demonstrated that future works can include multiple modality data, for example, the data from both genomic and proteomic analysis. A combination of genomic and proteomic analysis might help validate the results inferred from this deep learning pipeline. Also, this can be employed for other blood diseases as it is observed that models used in this study are insensitive to the dataset used. Furthermore, utilizing the proposed framework in trials and further fine-tuning of models with the trial data would help transition its use to actual medical practice.

References

[1] Ibrahim Abdulrab Ahmed, Ebrahim Mohammed Senan, Hamzeh Salameh Ahmad Shatnawi, Ziad Mohammad Alkhraisha, and Mamoun Mohammad Ali Al-Azzam. Hybrid techniques for the diagnosis of acute lymphoblastic leukemia based on fusion of cnn features. *Diagnostics*, 13(6), 2023.

[2] KK Anilkumar, VJ Manoj, and TM Sagi. Automated detection of b cell and t cell acute lymphoblastic leukaemia using deep learning. *Irbm*, 43(5):405–413, 2022.

[3] David A Basiji, William E Ortyn, Luchuan Liang, Vidya Venkatachalam, and Philip Morrissey. Cellular image analysis and imaging by flow cytometry. *Clinics in laboratory medicine*, 27(3):653–670, 2007.

[4] Christopher Bishop. *Pattern Recognition and Machine Learning*, volume 16, pages 140–155. 01 2006.

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[6] Sabina Chiaretti and Robin Foa. Management of adult ph-positive acute lymphoblastic leukemia. *Hematology 2014, the American Society of Hematology Education Program Book*, 2015(1):406–413, 2015.

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th*

International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19, pages 424–432. Springer, 2016.

[8] Maíla Claro, Luis Vogado, Rodrigo Veras, André Santana, João Tavares, Justino Santos, and Vinicius Machado. Convolution neural network models for acute leukemia diagnosis. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 63–68, 2020.

[9] Rahul C Deo. Machine learning in medicine. *Circulation*, 132(20):1920–1930, 2015.

[10] Nathalie Dhédin, Anne Huynh, Sébastien Maury, Reza Tabrizi, Kheira Beldjord, Vahid Asnafi, Xavier Thomas, Patrice Chevallier, Stéphanie Nguyen, Valérie Coiteux, et al. Role of allogeneic stem cell transplantation in adult patients with ph-negative acute lymphoblastic leukemia. *Blood, The Journal of the American Society of Hematology*, 125(16):2486–2496, 2015.

[11] Yi Ding, Fujuan Chen, Yang Zhao, Zhixing Wu, Chao Zhang, and Dongyuan Wu. A stacked multi-connection simple reducing net for brain tumor segmentation. *IEEE Access*, PP:1–1, 07 2019.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[14] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M

Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[15] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[16] Philipp Eulenberg, Niklas Köhler, Thomas Blasi, Andrew Filby, Anne E Carpenter, Paul Rees, Fabian J Theis, and F Alexander Wolf. Reconstructing cell cycle and disease progression using deep learning. *Nature communications*, 8(1):463, 2017.

[17] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111:98 – 136, 2014.

[18] Thaddeus C George, David A Basiji, Brian E Hall, David H Lynch, William E Ortyn, David J Perry, Michael J Seo, Cathleen A Zimmerman, and Philip J Morrissey. Distinguishing modes of cell death using the imagestream® multispectral imaging flow cytometer. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 59(2):237–245, 2004.

[19] Nicola Gökbuget, Hervé Dombret, Massimiliano Bonifacio, Albrecht Reichle, Carlos Graux, Christoph Faul, Helmut Diedrich, Max S Topp, Monika Brüggemann, Heinz-August Horst, et al. Blinatumomab for minimal residual disease in adults with b-cell precursor acute lymphoblastic leukemia. *Blood, The Journal of the American Society of Hematology*, 131(14):1522–1531, 2018.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[21] Lizz F Grimwade, Kathryn A Fuller, and Wendy N Erber. Applications of imaging flow cytometry in the diagnostic assessment of acute leukaemia. *Methods*, 112:39–45, 2017.

[22] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Additive models, trees, and related methods. *The elements of statistical learning: data mining, inference, and prediction*, pages 295–336, 2009.

[23] Teresa S Hawley and Robert G Hawley. *Flow cytometry protocols*, volume 2779. Springer Nature, 2024.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[26] Holger Hennig, Paul Rees, Thomas Blasi, Lee Kamentsky, Jane Hung, David Dao, Anne E Carpenter, and Andrew Filby. An open-source solution for advanced imaging flow cytometry data analysis using machine learning. *Methods*, 112:201–210, 2017.

[27] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[30] Melissa M Hudson, Kirsten K Ness, James G Gurney, Daniel A Mulrooney, Wassim Chemaitilly, Kevin R Krull, Daniel M Green, Gregory T Armstrong, Kerri A Nottage,

Kendra E Jones, et al. Clinical ascertainment of health outcomes among adults treated for childhood cancer. *Jama*, 309(22):2371–2381, 2013.

[31] Hiroto Inaba and Charles G Mullighan. Pediatric acute lymphoblastic leukemia. *Haematologica*, 105(11):2524, 2020.

[32] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[33] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, Tkianai, Adam Hogan, Lorenzomammana, YxNONG, AlexWang1900, Laurentiu Diaconu, Marc, Wanghaoyang0106, Ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu , Changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - bug fixes and performance improvements, oct 2020.

[34] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[35] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[36] N. Murali Krishna, Ramidi Yashwanth Reddy, Mallu Sai Chandra Reddy, Kasibhatla Phani Madhav, and Gaikwad Sudham. Object detection and tracking using yolo. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1–7, 2021.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[38] Fumio Kubota. Analysis of red cell and platelet morphology using an imaging-combined flow cytometer. *Clinical & Laboratory Haematology*, 25(2):71–76, 2003.

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[40] Feng Li, Lei Yan, Yuguang Wang, Jianxun Shi, Hua Chen, zhang xuedian, Minshan Jiang, Zhizheng Wu, and Kaiqian Zhou. Deep learning-based automated detection of glaucomatous optic neuropathy on color fundus photographs. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 258, 04 2020.

[41] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[42] Shicai Liu, Hailin Tang, Hongde Liu, and Jinke Wang. Multi-label learning for the diagnosis of cancer and identification of novel biomarkers with high-throughput omics. *Current Bioinformatics*, 16(2):261–273, 2021.

[43] Shannon L Maude, Theodore W Laetsch, Jochen Buechner, Susana Rives, Michael Boyer, Henrique Bittencourt, Peter Bader, Michael R Verneris, Heather E Stefanski, Gary D Myers, et al. Tisagenlecleucel in children and young adults with b-cell lymphoblastic leukemia. *New England Journal of Medicine*, 378(5):439–448, 2018.

[44] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S Corrado, Ara Darzi, et al. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94, 2020.

[45] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

[46] Borowitz Mj. Us-canadian consensus recommendations on the immunophenotypic analysis of hematologic neoplasia by flow cytometry: data analysis and interpretation. *Cytometry*, 30:236–244, 1997.

[47] Anthony V Moorman. The clinical relevance of chromosomal and genomic abnormalities in b-cell precursor acute lymphoblastic leukaemia. *Blood reviews*, 26(3):123–135, 2012.

[48] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *ArXiv e-prints*, 11 2015.

[49] David O'Connor, Amir Enshaei, Jack Bartram, Jeremy Hancock, Christine J Harrison, Rachael Hough, Sujith Samarasinghe, Claire Schwab, Ajay Vora, Rachel Wade, et al. Genotype-specific minimal residual disease interpretation improves stratification in pediatric acute lymphoblastic leukemia. *Journal of Clinical Oncology*, 36(1):34–43, 2018.

[50] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[51] Sana Parez, Naqqash Dilshad, Norah Alghamdi, Turki Alanazi, and Jong Lee. Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers. *Sensors*, 23:6949, 08 2023.

[52] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[53] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[54] Ching-Hon Pui, Jun J Yang, Stephen P Hunger, Rob Pieters, Martin Schrappe, Andrea Biondi, Ajay Vora, André Baruchel, Lewis B Silverman, Kjeld Schmiegelow, et al. Childhood acute lymphoblastic leukemia: progress through collaboration. *Journal of Clinical Oncology*, 33(27):2938–2948, 2015.

[55] Pranav Rajpurkar, Jeremy Irvin, Robyn L Ball, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis P Langlotz, et al. Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists. *PLoS medicine*, 15(11):e1002686, 2018.

[56] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[57] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[58] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[59] Kathryn G Roberts and Charles G Mullighan. Genomics in acute lymphoblastic leukaemia: insights and treatment implications. *Nature reviews Clinical oncology*, 12(6):344–357, 2015.

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[61] Adnan Saeed, Shifa Shoukat, Khurram Shehzad, Ijaz Ahmad, Ala' Abdulmajid Eshmawi, Ali H. Amin, and Elsayed Tag-Eldin. A deep learning-based approach for the diagnosis of acute lymphoblastic leukemia. *Electronics*, 11(19), 2022.

[62] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than

the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):1–21, 03 2015.

[63] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[64] Niranjana Sampathila, Krishnaraj Chadaga, Neelankit Goswami, Rajagopala P. Chadaga, Mayur Pandya, Srikanth Prabhu, Muralidhar G. Bairy, Swathi S. Katta, Devadas Bhat, and Sudhakara P. Upadya. Customized deep learning classifier for detection of acute lymphoblastic leukemia using blood smear images. *Healthcare*, 10(10), 2022.

[65] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[66] Malcolm Smith, Diane Arthur, Bruce Camitta, Andrew J Carroll, William Crist, Paul Gaynon, Richard Gelber, Nyla Heerema, Edward L Korn, Michael Link, et al. Uniform approach to risk classification and treatment assignment for children with acute lymphoblastic leukemia. *Journal of Clinical Oncology*, 14(1):18–24, 1996.

[67] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[68] Jing Sun, Lan Wang, Qiao Liu, Attila Tárnok, and Xuantao Su. Deep learning-based light scattering microfluidic cytometry for label-free acute lymphocytic leukemia classification. *Biomed. Opt. Express*, 11(11):6674–6686, Nov 2020.

[69] Siyan Tao, Yao Guo, Chuang Zhu, Huang Chen, Yue Zhang, Jie Yang, and Jun Liu. Highly efficient follicular segmentation in thyroid cytopathological whole slide image, 02 2019.

[70] Imad Tareq, Bassant Elbagoury, Salsabil El-Regaily, and El-Sayed El-Horbaty. Analysis

of ton-iot, unw-nb15, and edge-iiot datasets using dl in cybersecurity for iot. *Applied Sciences*, 12:9572, 09 2022.

[71] Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019.

[72] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[73] Keiko Tsuganezawa, Nobutaka Kiyokawa, Yoshinobu Matsuo, Fujiko Kitamura, Noriko Toyama-Sorimachi, Keisuke Kuida, Junichiro Fujimoto, and Hajime Karasuyama. Flow cytometric diagnosis of the cell lineage and developmental stage of acute lymphoblastic leukemia by novel monoclonal antibodies specific to human pre–b-cell receptor. *Blood, The Journal of the American Society of Hematology*, 92(11):4317–4324, 1998.

[74] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7:1–8, 2006.

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[76] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[77] Brent L Wood. Flow cytometric monitoring of residual disease in acute leukemia. *Hematological Malignancies*, pages 123–136, 2013.

[78] Renjie Xu, Haifeng Lin, Kangjie Lu, Lin Cao, and Yunfei Liu. A forest fire detection system based on ensemble learning. *Forests*, 12:217, 02 2021.

[79] Yan Xu, Zhipeng Jia, Liang-Bo Wang, Yuqing Ai, Fang Zhang, Maode Lai, and Eric I-Chao Chang. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC bioinformatics*, 18:1–17, 2017.

[80] Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, I Eric, and Chao Chang. Deep learning of feature representation with multiple instance learning for medical image analysis. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1626–1630. IEEE, 2014.

[81] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[82] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[83] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018.

Appendix A

Code

```
import torch
import torchvision
from torchvision import datasets, models
import cv2
from pathlib import Path
import matplotlib.pyplot as plt
import torchvision.transforms as transforms
import torch.nn.functional as F
import torch.nn as nn
import time
import datetime
!pip install safetensors
import safetensors
from safetensors.torch import load_file
import numpy as np
import os
from transformers import ViTForImageClassification
import torchvision.models as models
from torchvision.models.segmentation import fcn_resnet101
```

```
!pip install h5py
!pip install Pillow==10.4
import h5py
import PIL
from PIL import Image
from typing import Dict
import json

# All functions initializations
def transform_image(img, size):
    transform = transforms.Compose([
        transforms.Resize(size),
        transforms.ToTensor(),
    ])
    return transform(img)

def load_cnn_model(weights_path, model_class):
    model = model_class()
    model.load_state_dict(torch.load(weights_path, map_location=torch.device(’cpu’)), strict=False)
    model.eval()
    return model

def check_image_viability(model, img_path, size, threshold=0.8):
    from PIL import Image
    try:
        img = Image.open(img_path).convert(’RGB’)
        img = transform_image(img, size)
        img = img.unsqueeze(0)
        with torch.no_grad():
            output = model(img)
            viability = torch.sigmoid(output[:,0]).item()
            print(f"Viability␣score:␣{viability}")
        return viability > threshold
    except (FileNotFoundError, PIL.UnidentifiedImageError) as e:
        print(f"Error␣processing␣image:␣{e}")
        return False # Return False if image processing fails

def load_yolo_model(weights_path):
    model = torch.hub.load('ultralytics/yolov5', 'custom', path=weights_path, force_reload=True)
    return model

def get_bounding_boxes(model, img_path):
    results = model(img_path)
    detections = results.xyxy[0]
    #print(detections)
    img = cv2.imread(img_path)
    bounding_boxes = []
    for *box, conf, cls in detections:
        box_coords = [int(coord.item()) for coord in box]
        # Draw the bounding box on the image
        color = (0, 255, 0)  # Green color for the bounding box
        cv2.rectangle(img, (box_coords[0], box_coords[1]), (box_coords[2], box_coords[3]), color, 2)
        label = f"Cell␣{conf:.2f}"
        cv2.putText(img, label, (box_coords[0], box_coords[1] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
        bounding_boxes.append({
            'box': box_coords,
            'confidence': conf.item(),
            'class': cls.item()
        })
    # Convert BGR to RGB
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    # Convert to PIL Image for display
    img_pil = Image.fromarray(img_rgb)
    return bounding_boxes, img_pil


def extract_crops(img_path, bounding_boxes, output_dir):
    img = cv2.imread(img_path)
    img_name = os.path.basename(img_path).split('.')[0]
    crop_paths = []
    Path(output_dir).mkdir(parents=True, exist_ok=True)
    crops = []
    for idx, bbox in enumerate(bounding_boxes):
        x1, y1, x2, y2 = map(int, bbox['box'])
        crop = img[y1:y2, x1:x2]
        crop_path = os.path.join(output_dir, f"{img_name}_crop_{idx}.jpg")
        cv2.imwrite(str(crop_path), crop)
        crop_paths.append(crop_path)
        crops.append(crop)
    return crop_paths

def load_model(weights_path, model_class, num_labels=None):
    try:
        model = model_class(num_labels) if num_labels is not None else model_class()
    except TypeError:
        model = model_class()
    # Check if the file ends with '.safetensors'
    if weights_path.endswith('.safetensors'):
        state_dict = load_file(weights_path) # Use safetensors to load the file directly
    elif weights_path.endswith('.h5'): # Check if file ends with .h5
        with h5py.File(weights_path, 'r') as f: # If so, open it with h5py
            state_dict = {}
            for k, v in f.items():
                if isinstance(v, h5py.Dataset): # Check if it's a dataset
                    state_dict[k] = torch.tensor(v) # Load weights into state_dict
    else:
        # Load the model weights with map_location to CPU
        state_dict = torch.load(weights_path, map_location=torch.device('cpu'))
    model.load_state_dict(state_dict, strict=False)
    model.eval()
    return model

def classify_image(model, img_path, size):
    img = Image.open(img_path).convert('RGB')
    img = transform_image(img, size)
    img = img.unsqueeze(0)
    with torch.no_grad():
        output = model(img)
        _, predicted = torch.max(output.flatten(), 0)
    return predicted.item()

def transform_and_classify(model, img_path, size, custom_transforms):
    img = Image.open(img_path).convert(’RGB’)
    img = custom_transforms(img)
    img = img.unsqueeze(0)
    with torch.no_grad():
        output = model(img)
        _, predicted = torch.max(output, 1)
    return predicted.item()

def log_results(results, log_file):
    with open(log_file, ’w’) as f:
        json.dump(results, f, indent=4)
    return log_file

# Define your model classes
class CNNModel(nn.Module):
    def __init__(self, num_classes=2):
        super(CNNModel, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.bn1 = nn.BatchNorm2d(32)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.bn2 = nn.BatchNorm2d(64)
        self.conv3 = nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1)
        self.bn3 = nn.BatchNorm2d(128)
        self.fc1 = nn.Linear(128 * 16 * 16, 512)
        self.fc2 = nn.Linear(512, num_classes)
        self.relu = nn.ReLU()
        self.dropout = nn.Dropout(0.5)

    def forward(self, x):
        x = self.pool(self.relu(self.bn1(self.conv1(x))))
        x = self.pool(self.relu(self.bn2(self.conv2(x))))
        x = self.pool(self.relu(self.bn3(self.conv3(x))))
        x = x.view(-1, 128 * 16 * 16)
        x = self.relu(self.fc1(x))
        x = self.dropout(x)
        x = self.fc2(x)
        return x

class CellClassifier(nn.Module):
    def __init__(self, n_classes=3):
        super(CellClassifier, self).__init__()
        base_model = models.resnet34(pretrained=True)
        self.base_layers = list(base_model.children())
        self.layer0 = nn.Sequential(*self.base_layers[:3])
        self.layer1 = nn.Sequential(*self.base_layers[3:5])
        self.layer2 = self.base_layers[5]
        self.layer3 = self.base_layers[6]
        self.layer4 = self.base_layers[7]

```

```python
        self.upconv1 = nn.ConvTranspose2d(512, 256, kernel_size=2, stride=2)
        self.up1 = UNetBlock(256, 256)
        self.upconv2 = nn.ConvTranspose2d(256, 128, kernel_size=2, stride=2)
        self.up2 = UNetBlock(128, 128)
        self.upconv3 = nn.ConvTranspose2d(128, 64, kernel_size=2, stride=2)
        self.up3 = UNetBlock(64, 64)
        self.upconv4 = nn.ConvTranspose2d(64, 32, kernel_size=2, stride=2)
        self.up4 = UNetBlock(32, 32)
        self.final = nn.Conv2d(32, n_classes, kernel_size=1)

    def forward(self, x):
        # Encoder
        x0 = self.layer0(x)
        x1 = self.layer1(x0)
        x2 = self.layer2(x1)
        x3 = self.layer3(x2)
        x4 = self.layer4(x3)

        # Decoder
        x = self.upconv1(x4)
        x = self.up1(x)
        x = self.upconv2(x)
        x = self.up2(x)
        x = self.upconv3(x)
        x = self.up3(x)
        x = self.upconv4(x)
        x = self.up4(x)
        x = self.final(x)

        return x

class UNetBlock(nn.Module): # Added the missing UNetBlock class
    def __init__(self, in_channels, out_channels):
        super(UNetBlock, self).__init__()
        self.conv1 = nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1)
        self.bn1 = nn.BatchNorm2d(out_channels)
        self.relu = nn.ReLU()
        self.conv2 = nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1)
        self.bn2 = nn.BatchNorm2d(out_channels)

    def forward(self, x):
        x = self.relu(self.bn1(self.conv1(x)))
        x = self.relu(self.bn2(self.conv2(x)))
        return x

class WBCClassifier(nn.Module):
    def __init__(self, num_labels):
        super(WBCClassifier, self).__init__()
        # Load the pre-trained ViT model
        self.vit_model = ViTForImageClassification.from_pretrained('google/vit-base-patch16-224-in21k', num_labels=num_labels)
        # Extract the backbone part of the model (ViT without classifier head)
        self.backbone = self.vit_model.vit
        # Define the classifier head
        self.classifier = nn.Linear(self.vit_model.config.hidden_size, num_labels)

    def forward(self, x):
        # Forward pass through the ViT model
        outputs = self.backbone(x)[0]  # Extract the last hidden state
        # Apply the classifier head to the pooled output
        logits = self.classifier(outputs[:, 0])  # Use the representation of the [CLS] token
        return logits

class HealthyDetector(nn.Module):
    def __init__(self):
        super(HealthyDetector, self).__init__()
        self.model = models.resnet101(pretrained=False)
        num_ftrs = self.model.fc.in_features
        self.model.fc = nn.Linear(num_ftrs, 2)

    def forward(self, x):
        x = self.model(x)
        return x

class ALLClassifier(nn.Module):
    def __init__(self):
        super(ALLClassifier, self).__init__()
        self.base_model = models.densenet121(pretrained=False)
        num_ftrs = self.base_model.classifier.in_features
        self.base_model.classifier = nn.Sequential(
            nn.Linear(num_ftrs, 64),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(64, 3),
            nn.Softmax(dim=1)
        )

    def forward(self, x):
        x = self.base_model(x)
        return x

def main(cnn_weights_path, yolo_weights_path, img_path, output_dir,
        cell_classifier_weights, wbc_classifier_weights, healthy_detector_weights,
        all_classifier_weights, viability_threshold=0.5):

    results = []
    result_map = {
        "Parent Image": img_path,
        "Viability": "N/A",
        "Processed": "N/A",
        "crops": []
    }
    start_time = time.time()

    # Extract image name
    img_name = os.path.basename(img_path).split('.')[0]

    # Load models
```

```python
        model_load_start = time.time()
        (cnn_model, yolo_model, cell_classifier,
         wbc_classifier, healthy_detector, all_classifier) = load_all_models(
             cnn_weights_path, yolo_weights_path, cell_classifier_weights,
             wbc_classifier_weights, healthy_detector_weights, all_classifier_weights)
        model_load_time = time.time() - model_load_start

        # Check image viability
        viability_check_start = time.time()
        if not check_image_viability(cnn_model, img_path, (128, 128), viability_threshold):
            result_map['Viability'] = "Non-Viable"
        else:
            result_map['Viability'] = "Viable"
        #print(result_map)
        results.append(result_map)
        viability_check_time = time.time() - viability_check_start

        # Detect cells
        cell_detection_start = time.time()
        bounding_boxes, result_img = get_bounding_boxes(yolo_model, img_path)
        # Save the resulting image
        output_path = os.path.join(output_dir, f"processed_{img_name}.jpg")
        result_img.save(output_path)
        result_map['Processed'] = output_path
```

```
    cell_detection_time = time.time() - cell_detection_start

    # Extract crops and initialize crop details
    crop_extraction_start = time.time()
    crop_paths = extract_crops(img_path, bounding_boxes, output_dir)
    for i, (bbox, crop_path) in enumerate(zip(bounding_boxes, crop_paths)):
        crop_path = os.path.join(output_dir, f"{img_name}_crop_{i}.jpg")
        crop_details = {
            'crop': crop_path,
            'cell_det': ','.join(map(str, bbox['box'])),
            'cell_class': "N/A",
            'wbc_class': "N/A",
            'health_check': "N/A",
            'malignancy_check': "N/A"
        }
        result_map['crops'].append(crop_details)
    crop_extraction_time = time.time() - crop_extraction_start

    # Cell Classification
    cell_classification_start = time.time()
    for crop_details in result_map['crops']:
        prediction = classify_image(cell_classifier, crop_details['crop'], (224, 224))
        if prediction == 2:
            crop_details['cell_class'] = "White␣Blood␣Cell"
        elif prediction == 1:
            crop_details['cell_class'] = "Platelet"
        elif prediction == 0:
            crop_details['cell_class'] = "Red␣Blood␣Cell"
    cell_classification_time = time.time() -
       cell_classification_start

    # Further classify WBC crops
    wbc_classification_start = time.time()
    prediction_map = {
        0: 'immature␣neutrophil', 1: 'blast␣cells', 2: 'lymphocytes
           ',
        3: 'monocytes', 4: 'myelocyte', 5: 'neutrophil',
        6: 'nucleated␣red␣blood␣cells', 7: 'platelet',
        8: 'promeylocyte', 9: 'red␣blood␣cell'
    }
    lymphocyte_crops = []
    for crop_details in result_map['crops']:
        if crop_details['cell_class'] == "White␣Blood␣Cell":
            prediction = classify_image(wbc_classifier,
               crop_details['crop'], (224, 224))
            cell_name = prediction_map.get(prediction, 'N/A␣Cell␣
               Type')
            if prediction in [1, 2]:
                lymphocyte_crops.append(crop_details['crop'])
                crop_details['wbc_class'] = "Lymphocytes"
            elif prediction != 9:
                crop_details['wbc_class'] = cell_name
    wbc_classification_time = time.time() -
       wbc_classification_start

    # Detect healthy/non-healthy lymphocytes
```

```python
    health_classification_start = time.time()
    for crop_path in lymphocyte_crops:
        crop_details = next(item for item in result_map['crops'] if item['crop'] == crop_path)
        prediction = classify_image(healthy_detector, crop_path, (224, 224))

        if prediction == 0:
            crop_details['health_check'] = "Benign cells, no malignancy."
            crop_details['malignancy_check'] = "N/A"
        else:
            crop_details['health_check'] = "Malignant"
            # Further classify non-healthy lymphocytes
            custom_transforms = transforms.Compose([
                transforms.RandomResizedCrop(224),
                transforms.RandomHorizontalFlip(),
                transforms.ToTensor(),
                transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
            ])
            prediction = transform_and_classify(all_classifier, crop_path, (224, 224), custom_transforms)
            if prediction == 0:
                crop_details['malignancy_check'] = "Malignant Early Pre-B cells"
            elif prediction == 1:
                crop_details['malignancy_check'] = "Malignant Pre-B cells"
            elif prediction == 2:
                crop_details['malignancy_check'] = "Malignant␣Pro-B␣cells"
    health_classification_time = time.time() - health_classification_start

    # Convert the result map to a list of results and log once per crop
    results = list(result_map.values())

    # Print and log all results at once to avoid repetition
    for result in results:
        print(result)

    # Log results to file
    log_file = os.path.join(output_dir, f"results_{img_name}.json")
    log_results(results, log_file)
    print(f"Results␣logged␣in␣{log_file}")

    total_time = time.time() - start_time

    # Print runtime for each step and total time
    print(f"Model␣Load␣Time:␣{model_load_time:.2f}␣seconds")
    print(f"Viability␣Check␣Time:␣{viability_check_time:.2f}␣seconds")
    print(f"Cell␣Detection␣Time:␣{cell_detection_time:.2f}␣seconds")
    print(f"Crop␣Extraction␣Time:␣{crop_extraction_time:.2f}␣seconds")
    print(f"Cell␣Classification␣Time:␣{cell_classification_time:.2f}␣seconds")
    print(f"WBC␣Classification␣Time:␣{wbc_classification_time:.2f}␣seconds")
    print(f"ALL␣Classification␣Time:␣{health_classification_time:.2f}␣seconds")
    print(f"Total␣Pipeline␣Time:␣{total_time:.2f}␣seconds")

    return

from concurrent.futures import ThreadPoolExecutor

def load_all_models(cnn_weights_path, yolo_weights_path, cell_classifier_weights, wbc_classifier_weights, healthy_detector_weights, all_classifier_weights):
    with ThreadPoolExecutor() as executor:
        future_cnn = executor.submit(load_cnn_model, cnn_weights_path, CNNModel)
        future_yolo = executor.submit(load_yolo_model, yolo_weights_path)
        future_cell_classifier = executor.submit(load_model, cell_classifier_weights, CellClassifier)
        future_wbc_classifier = executor.submit(load_model, wbc_classifier_weights, WBCClassifier, 10)
        future_healthy_detector = executor.submit(load_model, healthy_detector_weights, HealthyDetector)
        future_all_classifier = executor.submit(load_model, all_classifier_weights, ALLClassifier)

        cnn_model = future_cnn.result()
        yolo_model = future_yolo.result()
        cell_classifier = future_cell_classifier.result()
        wbc_classifier = future_wbc_classifier.result()
        healthy_detector = future_healthy_detector.result()
        all_classifier = future_all_classifier.result()

    return cnn_model, yolo_model, cell_classifier, wbc_classifier, healthy_detector, all_classifier

!pip install gradio
import gradio as gr


def process_image_and_get_results(img_path):
    # Define paths
    cnn_weights_path = '.git/models/viable_detection.pth'
    yolo_weights_path = '.git/models/Cell_Detector.pt'
    cell_classifier_weights = '.git/models/Cell_Classifier_UR.pth'
    wbc_classifier_weights = '.git/models/vit_wbc_classifier/model.safetensors'
    healthy_detector_weights = '.git/models/Malignancy_resnet101_model.pth'
    all_classifier_weights = '.git/models/densenet121_all_classifier.h5'
    output_dir = '.git//output'
    os.makedirs(output_dir, exist_ok=True)

    # Run the main function
    main(cnn_weights_path, yolo_weights_path, img_path, output_dir,
         cell_classifier_weights, wbc_classifier_weights, healthy_detector_weights,
         all_classifier_weights)

    # Parse the results from the generated JSON file
    img_name = os.path.basename(img_path).split(’.’)[0]
    log_file = os.path.join(output_dir, f"results_{img_name}.json")

    if os.path.exists(log_file):
        with open(log_file, ’r’) as f:
            results = json.load(f)
        return json.dumps(results, indent=4)
    else:
        return "No␣results␣found."

# Gradio Interface
iface = gr.Interface(
    fn=process_image_and_get_results,
    inputs=gr.Textbox(label="Image␣Path"),
    outputs=gr.Textbox(label="Analysis␣Results"),
    title="ALL␣Diagnosis␣Pipeline",
    description="Upload␣an␣image␣and␣get␣the␣cell␣analysis␣results."
)

# Launch the Gradio app
if __name__ == "__main__":
    iface.launch()
```