Thesis submitted to the

UNIVERSITY OF MOHAMED BOUDIAF – MSILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

In partial fulfillment of the requirements for the degree of

Master in **Artificial Intelligence**

By

Drif, **Mohamed Amine**

Title of the thesis

Deep Learning for Medical Image Classification:

CNN-Based Approaches Applied to Leukemia and Retinal Disease Detection

---

Under the supervision of

Dr.Mokhtari Rabah

Composition of the jury

Dr.Bahache Mohamed	University of Msila	President
Dr.Kamel Mohamed	University of Msila	Examiner

May, 2026.

DEDICATION

To my parents for everything they sacrificed so I could reach this point. No words come close.

To my family, for the warmth and support that carried me through every difficult moment.

To my friends, for the laughs, the late nights, and the encouragement when it was needed most.

And to everyone who ever believed in me before I believed in myself this one is for you

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my thesis supervisor for their invaluable guidance, patience, and continuous support throughout this work. Their expertise and dedication made this project possible, and their feedback pushed me to think more carefully and work more rigorously at every stage.

I am also grateful to the faculty members and teaching staff of the University of M'Sila for the knowledge they shared and the academic environment they helped build one that genuinely encourages curiosity and hard work.

Special thanks go to my family, who supported me unconditionally throughout this journey. Their encouragement during the most stressful moments of this research meant more than they know.

To my friends and colleagues who were there for discussions, shared frustrations, and the occasional much-needed break from work thank you. This experience was better because of you.

Finally, I would like to acknowledge the open-source community and the researchers who made their datasets publicly available the C-NMC 2019 and Kermany OCT datasets that are central to this work. Science moves forward because people share their work, and this thesis is built on that generosity.

Abstract

This thesis investigates the application of transfer learning with pretrained convolutional neural networks for automated medical image classification across two clinically critical domains: acute lymphoblastic leukemia detection from microscopic blood smear images, and binary retinal disease classification from optical coherence tomography scans. Three CNN architectures VGG16, MobileNetV2, and EfficientNetB0 are evaluated under four training strategies: frozen backbone transfer, two-phase fine-tuning, data augmentation, and SVM-based hybrid classification. A total of 24 model configurations are systematically compared under identical controlled conditions on the C-NMC 2019 leukemia dataset and the restructured Kermany OCT dataset. Results show that EfficientNetB0 with two-phase fine-tuning achieves the best performance on both tasks, reaching 98.66% accuracy on OCT and 80.2% on the leukemia dataset, including perfect ALL recall. MobileNetV2 demonstrates competitive results with significantly fewer parameters, making it a strong candidate for resource-constrained deployment. These findings contribute a rigorous and reproducible comparative framework for CNN-based transfer learning in medical imaging.

Keywords: Transfer Learning; Convolutional Neural Networks; Medical Image Classification; Leukemia Detection; Retinal Disease.

ملخص

تتناول هذه الرسالة دراسة تطبيق التعلم بالنقل (Transfer Learning) باستخدام الشبكات العصبية التلافيفية (CNN) المُدرَّبة مسبقًا، من أجل التصنيف الآلي للصور الطبية في مجالين حاسمين على الصعيد السريري: الكشف عن سرطان الدم اللمفاوي الحاد (ALL) من خلال صور مسحة الدم المجهرية، والتصنيف الثنائي لأمراض الشبكية باستخدام صور التصوير المقطعي البصري التماسكي (OCT).

تم تقييم ثلاث بنى للشبكات العصبية التلافيفية، وهي VGG16 و MobileNetV2 و EfficientNetB0، وفق أربع استراتيجيات تدريب: النقل بتجميد العمود الفقري للشبكة (Frozen Backbone)، والضبط الدقيق على مرحلتين (Two-Phase Fine-Tuning)، وزيادة البيانات (Data Augmentation)، والتصنيف الهجين القائم على آلة الدعم الشعاعي (SVM).

تمت دراسة ما مجموعه 24 تهيئة نموذجية (Model Configurations) بشكل منهجي ومقارن في ظل شروط محكومة وموحدة، باستخدام مجموعة بيانات C-NMC 2019 الخاصة بسرطان الدم، ومجموعة بيانات Kermany OCT المُعاد هيكلتها.

تُظهر النتائج أن نموذج EfficientNetB0 المُدرَّب باستراتيجية الضبط الدقيق على مرحلتين يحقق الأداء الأفضل في كلتا المهمتين، حيث بلغت دقته 98.66% على بيانات OCT، و80.2% على بيانات سرطان الدم، مع تحقيقه استدعاءً (Recall) كاملاً لفئة ALL.

كما أظهر نموذج MobileNetV2 نتائج تنافسية بعدد أقل بكثير من المعاملات (Parameters)، مما يجعله مرشحًا قويًا للنشر في البيئات محدودة الموارد. وتُقدّم هذه النتائج إطارًا مقارنًا دقيقًا وقابلًا لإعادة التطبيق للتعلم بالنقل القائم على الشبكات العصبية التلافيفية في مجال تصوير الطب.

الكلمات المفتاحية: التعلم بالنقل؛ الشبكات العصبية التلافيفية؛ تصنيف الصور الطبية؛ الكشف عن سرطان الدم؛ أمراض الشبكية.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
AI	Artificial Intelligence
ALL	Acute Lymphoblastic Leukemia
AML	Acute Myeloid Leukemia
AMD	Age-Related Macular Degeneration
BCR-ABL	Breakpoint Cluster Region – Abelson (fusion gene)
CAD	Computer-Aided Diagnosis
CLL	Chronic Lymphocytic Leukemia
CML	Chronic Myeloid Leukemia
CNN	Convolutional Neural Network
C-NMC	Computational Nuclei Morphometry Challenge (2019 dataset)
CNV	Choroidal Neovascularization
CPU	Central Processing Unit
DME	Diabetic Macular Edema
DL	Deep Learning
EHR	Electronic Health Record
EfficientNetB0	Efficient Neural Network – Baseline variant (B0)
FISH	Fluorescence In Situ Hybridization
FN	False Negative
FP	False Positive
F1	F1-Score (harmonic mean of Precision and Recall)
GPU	Graphics Processing Unit
HEM	Hematogone (benign precursor B-cell class in C-NMC)
ImageNet	Large-scale visual recognition benchmark dataset
ISBI	IEEE International Symposium on Biomedical Imaging
lr	Learning Rate
MBConv	Mobile Inverted Bottleneck Convolution
ML	Machine Learning
MobileNetV2	Mobile Neural Network version 2
MRI	Magnetic Resonance Imaging
OCT	Optical Coherence Tomography
PCR	Polymerase Chain Reaction
PET	Positron Emission Tomography
RAM	Random Access Memory
RBF	Radial Basis Function (kernel)
ReLU	Rectified Linear Unit
RF	Random Forest
RPE	Retinal Pigment Epithelium
SVM	Support Vector Machine

TKI	Tyrosine Kinase Inhibitor
TN	True Negative
TP	True Positive
U-Net	U-shaped Convolutional Network (segmentation architecture)
VGG16	Visual Geometry Group network with 16 layers
VGG	Visual Geometry Group (Oxford University)
ViT	Vision Transformer
WBC	White Blood Cell
XGBoost	Extreme Gradient Boosting

General Introduction

Medical imaging has become an indispensable pillar of modern clinical diagnosis. From the microscopic analysis of blood samples to the non-invasive examination of retinal tissue, imaging technologies generate vast amounts of visual data that must be interpreted rapidly, accurately, and consistently. In parallel, the global burden of diseases such as leukemia and retinal pathologies continues to grow, placing increasing pressure on healthcare systems and medical specialists worldwide. Acute Lymphoblastic Leukemia (ALL), the most common childhood cancer, requires early and precise identification of malignant lymphoblasts in peripheral blood smears a task that remains highly dependent on the expertise of trained hematologists. Similarly, retinal diseases such as Choroidal Neovascularization (CNV) and Diabetic Macular Edema (DME) are leading causes of preventable blindness, yet their timely diagnosis relies on the manual interpretation of Optical Coherence Tomography (OCT) scans, a process that is both time-consuming and subject to inter-observer variability.

These diagnostic challenges share a common limitation: the reliance on human visual interpretation, which is inherently constrained by fatigue, subjectivity, and the availability of specialized expertise. In resource-limited clinical environments, where access to trained specialists is scarce, the consequences of delayed or inaccurate diagnosis can be severe. There is therefore a pressing need for automated, reliable, and scalable computer-aided diagnostic systems capable of supporting clinical decision-making across both hematological and ophthalmological domains.

The rapid advancement of deep learning, and convolutional neural networks in particular, has opened new possibilities for automated medical image analysis. Pretrained CNN architectures, originally developed for large-scale natural image recognition tasks, have demonstrated remarkable transferability to medical imaging domains through the paradigm of transfer learning. Rather than training models from scratch on limited medical datasets, transfer learning allows pretrained feature extractors to be adapted to domain-specific classification tasks with significantly reduced data requirements. This approach has yielded strong results across a broad range of medical imaging applications, yet systematic comparative evaluations across architectures, training strategies, and datasets remain relatively scarce in the literature.

The present work addresses this gap by proposing a rigorous comparative framework for CNN-based transfer learning applied to two distinct medical image classification problems:

leukemia detection from microscopic blood smear images, and binary retinal disease classification from OCT scans. Three pretrained architectures VGG16, MobileNetV2, and EfficientNetB0 are evaluated under four training strategies: frozen backbone transfer learning, two-phase fine-tuning, data augmentation, and hybrid CNN combined with classical machine learning classifiers. All experiments are conducted under identical controlled conditions, enabling direct and meaningful comparison in this thesis.

The document is organized as follows:

Chapter 1 provides the general background, covering the medical context of leukemia and retinal diseases, the fundamentals of artificial intelligence in healthcare, computer vision, machine learning, and deep learning, with particular emphasis on convolutional neural networks and transfer learning.

Chapter 2 presents the datasets and models used in this work. The C-NMC 2019 leukemia dataset and the restructured Kermany OCT dataset are described in detail, followed by a presentation of the three CNN architectures. The chapter concludes with a review of existing works situating this contribution within the current research landscape.

Chapter 3 covers the development environment and the full implementation pipeline. It details the software and hardware setup used throughout the experiments, the data preprocessing and augmentation procedures, the two-phase fine-tuning strategy, the SVM-based hybrid classification approach, and the practical engineering decisions made to ensure consistency and reproducibility .

Chapter 4 presents the experimental results obtained across all model configurations on both datasets mentioned in chapter 3, including training curves, confusion matrices, and classification metrics, followed by a comparative analysis and general discussion.

The general conclusion summarizes the principal findings, acknowledges the limitations of the study, and outlines perspectives for future research.

Chapter 1

General Background and Related Concepts

1.1 Introduction

The growing intersection of artificial intelligence and medical science has opened unprecedented opportunities for automating complex diagnostic tasks. Among the most critical of these is the detection and classification of blood cancers, particularly leukemia, whose early identification directly determines patient survival outcomes. This chapter establishes the theoretical and conceptual foundations upon which the work presented in this thesis is built.

The chapter begins by introducing leukemia and retinal diseases as a medical condition, covering its biological basis, clinical manifestations, and the principal categories recognized in modern hematology. It then transitions into the computational domain, presenting the field of computer vision and its relevance to medical image analysis, with emphasis on the preprocessing and segmentation operations required to prepare microscopic blood cell images for automated analysis. The chapter subsequently introduces machine learning and deep learning, with a focused treatment of Convolutional Neural Networks (CNNs) and transfer learning. Finally, the growing role of artificial intelligence in medicine is surveyed, contextualizing the present work within a broader landscape of AI-driven healthcare applications .

1.2 Leukemia and Retinal Diseases

1.2.1 Leukemia: Definition and Pathophysiology

Leukemia is a malignant neoplasm of the hematopoietic system originating in the bone marrow. It is characterized by the excessive proliferation of abnormal, immature white blood cells called blasts that fail to mature and accumulate in the marrow, peripheral blood, and other organs, suppressing the production of normal blood cells .

The pathophysiology involves a multi-step oncogenic process. Initiating mutations in a hematopoietic progenitor cell disrupt normal differentiation signaling, giving the affected clone a proliferative advantage. Secondary mutations further disable apoptotic mechanisms, eventually producing a fully malignant clone. In acute leukemias this process is rapid and aggressive in chronic forms the disease progresses more slowly with cells retaining partial differentiation.

Clinical manifestations include fatigue and anemia from reduced red blood cell production, recurrent infections due to impaired immune function, and abnormal bleeding from thrombocytopenia. Hepatosplenomegaly and lymphadenopathy may also occur as leukemic cells infiltrate organs . [1]-[3]

1.2.2 Classification of Leukemia

Leukemia is classified along two axes: the rate of progression (acute vs. chronic) and the cell lineage affected (lymphoid vs. myeloid), yielding four principal subtypes (**Fig 1.1**) :

- Acute Lymphoblastic Leukemia (ALL): rapid proliferation of immature lymphoid precursors (lymphoblasts). It is the most common childhood cancer, representing ~75% of all pediatric leukemia cases;
- Acute Myeloid Leukemia (AML): rapid accumulation of myeloid blasts; more prevalent in adults with a relatively poor prognosis;
- Chronic Lymphocytic Leukemia (CLL): slow-progressing malignancy of mature lymphocytes, mainly in the elderly;
- Chronic Myeloid Leukemia (CML): defined by the Philadelphia chromosome (BCR-ABL fusion gene), progressing through chronic, accelerated, and blast crisis phases.

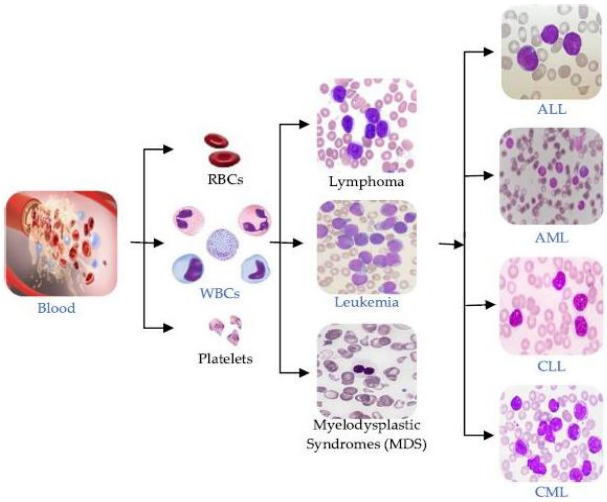

Fig. 1.1 Classification of leukemia by rate of progression and cell lineage.

This thesis focuses on ALL due to its high incidence in children (**Table 1.1**), the clinical urgency of rapid diagnosis, and the availability of well-annotated microscopic image datasets for this subtype. [3]

Table 1.1: Summary of leukemia subtypes and key clinical characteristics.

Subtype	Cell lineage	Onset	Typical age	5-yr survival
ALL	Lymphoid	Acute	Children (2-5 yrs)	~90% (children)
AML	Myeloid	Acute	Adults (>65 yrs)	~30%
CLL	Lymphoid	Chronic	Adults (>60 yrs)	~83%
CML	Myeloid	Chronic	Adults (40-60 yrs)	~70% (TKI)

1.2.3 Retinal Diseases and OCT Imaging

The retina is a thin layer of light-sensitive neural tissue located at the posterior of the eye. It converts incoming light into electrical signals that are transmitted to the brain via the optic nerve, enabling vision. Damage to the retinal structure, whether through vascular abnormalities, fluid accumulation, or cellular degeneration, leads to irreversible vision loss if not detected and treated early .[4]

Several retinal pathologies are among the leading causes of preventable blindness worldwide.

- Choroidal Neovascularization (CNV) : is characterized by the abnormal growth of new blood vessels originating from the choroid and penetrating

through the Bruch membrane into the sub-retinal space. These vessels are structurally fragile and tend to leak fluid and blood, causing progressive and often rapid central vision loss. CNV is the most severe complication of neovascular age-related macular degeneration (AMD), a condition affecting approximately 196 million people globally [6]. In OCT images, CNV appears as a hyper-reflective irregular membrane beneath the retinal pigment epithelium (RPE), often accompanied by subretinal fluid accumulation. [5]

- Diabetic Macular Edema (DME) : is a complication of diabetic retinopathy, which itself results from long-term damage to the retinal blood vessels caused by chronic hyperglycemia. In DME, the breakdown of the blood-retinal barrier causes fluid to leak into the macula, the central region of the retina responsible for fine detail and color vision. This fluid accumulation distorts the retinal architecture and causes progressive central visual impairment .DME is the leading cause of vision loss among working-age adults with diabetes, affecting an estimated 21 million people worldwide .In OCT, DME manifests as intraretinal cystoid spaces and loss of the normal foveal contour. [6]
- Optical Coherence Tomography (OCT) : is the gold-standard non-invasive imaging modality for diagnosing and monitoring retinal diseases. It operates on the principle of low-coherence interferometry, using near-infrared light to produce high-resolution cross-sectional images of the retinal layers with micrometer-level axial resolution. A single OCT B-scan captures the full depth profile of the retina, making structural anomalies immediately visible as deviations from the normal flat, stratified architecture. OCT has transformed ophthalmological practice: what previously required fluorescein angiography, an invasive procedure involving intravenous dye injection, can now be assessed non-invasively in seconds . [7]

Retinal OCT images are generally categorized into distinct classes, including normal retina, CNV, DME, and other pathological conditions. **Fig 1.2** illustrates representative examples of these categories, highlighting the visual differences between healthy and diseased retinal structures. [7][8]

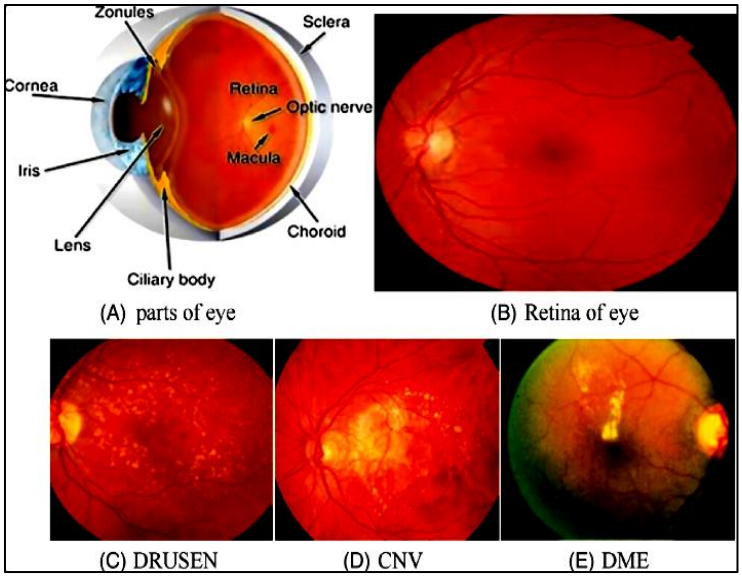

Fig. 1.2 Figure 1.2: Anatomical structure of the eye and examples of retinal conditions (Normal, Drusen, CNV, and DME).

Epidemiology and Risk Factors

ALL is the most prevalent cancer in the pediatric population, with peak incidence between ages 2 and 5. In high-income countries the five-year survival rate for childhood ALL exceeds 90% with modern chemotherapy protocols, though outcomes remain significantly worse in low-resource settings. AML predominantly affects adults over 65 and carries a five-year survival rate of approximately 30% [3].

Risk factors include ionizing radiation, benzene and chemical carcinogens, prior chemotherapy or radiotherapy, and genetic predispositions such as Down syndrome, Fanconi anemia, and Li-Fraumeni syndrome. The precise etiology of most leukemia cases remains incompletely understood.

Optical Coherence Tomography (OCT) is the gold-standard non-invasive imaging modality for retinal diagnosis. OCT uses low-coherence interferometry to produce high-resolution cross-sectional images of the retinal layers with micrometer-level axial resolution. A single OCT scan captures the full thickness profile of the retina, making structural abnormalities such as intraretinal fluid pockets (in DME), choroidal membrane disruptions (in CNV), and drusen deposits clearly visible as deviations from the normal flat layered architecture. The challenge for automated analysis lies in the subtle visual differences between pathological and normal OCT scans, and between different disease subtypes [7].

The retina is a thin layer of photosensitive neural tissue lining the posterior of the eye. Two major retinal pathologies motivate the second classification task in this thesis. Choroidal Neovascularization (CNV) involves the abnormal growth of new blood vessels beneath the retinal pigment epithelium, commonly associated with advanced age-related macular degeneration (AMD). These vessels are fragile and tend to leak fluid, causing rapid central vision loss if left untreated. Diabetic Macular Edema (DME) results from fluid accumulation in the macula due to breakdown of the blood-retinal barrier in diabetic retinopathy. It is a leading cause of preventable blindness worldwide [4] [5].

1.2.4 Diagnosis: Methods and Limitations

The gold standard for leukemia diagnosis is microscopic examination of stained peripheral blood smears and bone marrow aspirates by expert hematologists. After Wright-Giemsa staining, cells are observed at 100x magnification and pathologists manually identify blast cells, assessing morphology, nuclear-to-cytoplasmic ratio, chromatin pattern, and nucleolar prominence [3][9].

This approach, while indispensable, presents important limitations that motivate automated tools:

- it is time-intensive, often requiring 30 to 60 minutes of concentrated observation per sample;
- accuracy strongly depends on pathologist experience, introducing significant inter-observer variability;
- morphological similarity between malignant lymphoblasts and reactive lymphocytes makes visual discrimination difficult even for specialists;
- access to trained hematopathologists is limited in many regions, including Algeria, creating diagnostic bottlenecks;
- manual counting is susceptible to fatigue-induced errors when large numbers of slides must be processed.

Complementary modalities flow cytometry, FISH, and PCR are highly specific but expensive and inaccessible in many settings. Automated computer-vision classification systems therefore offer a compelling approach to reducing cost, improving throughput, and enhancing diagnostic consistency.

1.3 Artificial Intelligence in Medicine

1.3.1 Overview of AI in Healthcare

Artificial intelligence (AI) refers to the capability of computational systems to perform tasks ordinarily requiring human-level cognition perception, reasoning, learning, and decision-making. In healthcare, AI has emerged as a transformative technology spanning medical imaging, genomics, drug discovery, clinical decision support, and patient outcome prediction [10].

This acceleration since the mid-2010s was driven by three convergent factors: the explosion of digitized medical data (EHRs, imaging archives, genomic databases), powerful deep learning algorithms capable of learning complex representations from raw data, and the availability of high-performance GPU hardware. Together these enabled AI systems to approach or surpass human expert performance on specific clinical tasks.

1.3.2 Computer-Aided Diagnosis (CAD)

Computer-Aided Diagnosis (CAD) systems are software tools that assist clinicians in interpreting medical images by automatically detecting, characterizing, or classifying findings of interest. They act as a second reader, flagging potential abnormalities and reducing missed diagnoses as it shows in **Fig 1.3** [10][11].

Early CAD systems used hand-crafted image features with classical classifiers such as SVMs or decision trees. Modern deep learning CAD systems learn task-specific features.

directly from raw image data without manual feature engineering. Applications in oncology include pulmonary nodule detection in CT, microcalcification detection in mammography, polyp detection in colonoscopy, and malignant cell classification in histopathology slides.

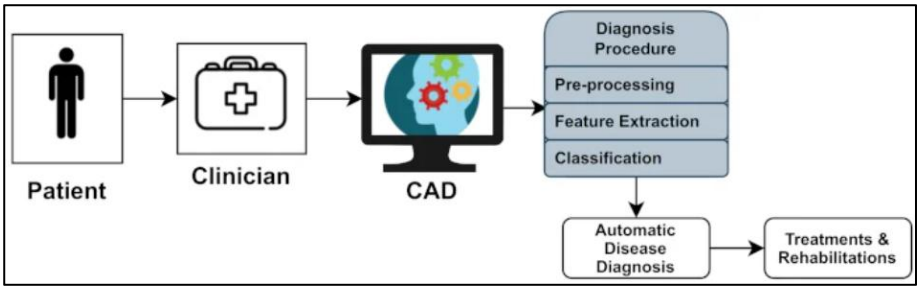

Fig. 1.3 Computer-aided diagnosis (CAD) system architecture.

1.3.3 AI for Cancer and Retinal Disease Detection

The application of AI to hematological malignancy detection has been active for over two decades. Early work applied classical image processing to segment and classify white blood cells, using shape, texture, and color histogram features fed into statistical classifiers.

The advent of deep learning transformed the field. CNNs have demonstrated state-of-the-art performance on leukemia cell classification, in several cases matching or exceeding human hematologist accuracy. Matek et al. demonstrated human-level recognition of blast cells using CNNs on a large AML dataset, and the C-NMC 2019 challenge benchmarked numerous deep learning architectures on ALL classification from bone marrow microscopy images .

These advances underscore the clinical potential of AI-based leukemia detection and motivate the hybrid CNN plus classical machine learning approach adopted in this thesis, combining the feature learning power of deep networks with the interpretability and efficiency of classical classifiers [9][12].

1.4 Computer Vision and Medical Image Analysis

1.4.1 Introduction to Computer Vision

Computer vision is the subfield of AI concerned with enabling machines to interpret visual information. It encompasses image classification, object detection, semantic segmentation, and image generation. Algorithms extract meaningful information from raw pixels by learning hierarchical representations capturing edges, textures, shapes, and semantic concepts.

In the medical domain, computer vision is applied across imaging modalities X-ray, CT, MRI, PET, ultrasound, and optical microscopy each presenting unique challenges in image characteristics, noise profiles, and relevant visual features [13].

1.4.2 Image Preprocessing

Raw microscopic blood smear images are rarely suitable for direct model input without preprocessing. Common steps in leukocyte image analysis include [15]:

- color normalization: stain variability across dye batches and scanners is corrected using algorithms such as Macenko or Reinhard normalization, which transfer color statistics from a reference image to all others,

- image resizing: all images are resized to a standard resolution using bilinear or bicubic interpolation.
- noise reduction: Gaussian smoothing or median filtering removes acquisition noise,
- intensity normalization: pixel values are rescaled to [0,1] or standardized to zero mean and unit variance to accelerate convergence and prevent numerical instability.

1.4.3 Cell and Retina Segmentation

Image segmentation partitions an image into meaningful regions. In leukemia detection, it refers to isolating individual white blood cells from background, red blood cells, and plasma. Accurate segmentation is a prerequisite for reliable cell-level classification.

Classical segmentation methods include Otsu's thresholding, watershed algorithms, and active contour models, which exploit color and intensity contrast to delineate boundaries. However, these methods struggle with overlapping cells, staining variability, and low-contrast backgrounds. Deep learning segmentation using U-Net has largely superseded classical methods: its encoder-decoder structure with skip connections achieves precise pixel-level segmentation even on limited annotated data [16].

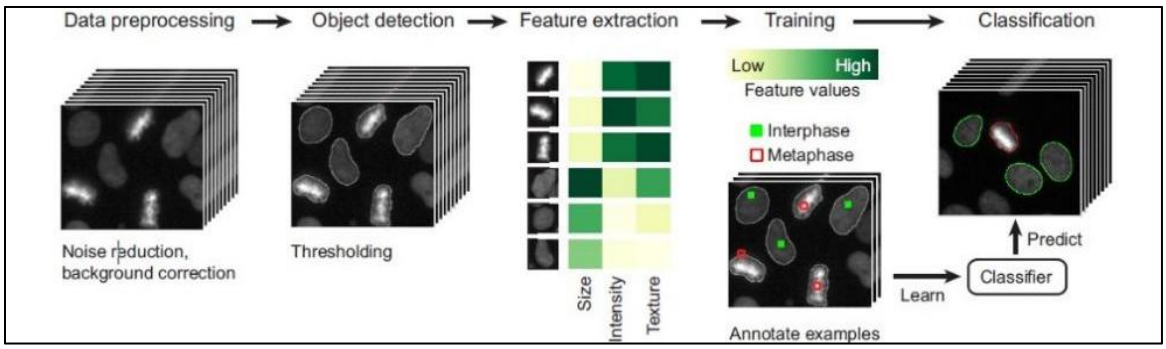

Fig. 1.4 Blood cell segmentation pipeline for leukemia detection.

1.4.4 Data Augmentation

Data augmentation artificially increases training dataset size and diversity by applying label-preserving transformations to existing samples. It is critical in medical image analysis where collecting large annotated datasets is costly. Augmentation reduces overfitting and improves generalization [17].

1.5 Machine Learning

1.5.1 Definition and Categories

Machine learning (ML) is a subfield of AI in which algorithms learn predictive models directly from data, without explicit domain-specific programming. The learning process involves estimating function parameters by minimizing a loss metric on training data [18].

Three paradigms are distinguished:

- supervised learning: models trained on labeled data to learn input-output mappings. Classification and regression are the principal tasks;
- unsupervised learning: models operate on unlabeled data to discover inherent structure, as in clustering and dimensionality reduction;
- reinforcement learning: an agent learns by interacting with an environment and receiving reward signals, maximizing cumulative reward.
- Binary classification of leukemic vs. normal cells from microscopic images falls squarely within the supervised learning paradigm.

1.5.2 Classical Classifiers

Classical ML classifiers are applied to feature vectors extracted by pretrained CNN layers, such as :

- Support Vector Machines (SVM): SVMs find the maximum-margin hyperplane separating two classes in a high-dimensional feature space. Kernel functions (RBF, polynomial, linear) handle nonlinearly separable problems. SVMs are robust in high-dimensional small-sample settings, making them well-suited to medical image tasks.
- Random Forest (RF): an ensemble of decision trees trained on random data and feature subsets (bagging), aggregating predictions by majority vote. It is robust to overfitting, handles high-dimensional inputs well, and provides feature importance scores.

1.5.3 Evaluation Metrics

Classifier performance is assessed via metrics derived from the confusion matrix (TP, TN, FP, FN):

- Accuracy: proportion of correctly classified samples. Can be misleading under class imbalance;

$$\mathbf{1}) \text{ Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Measures the rate of true positives among positive predictions;

$$\mathbf{2}) \text{ Precision} = \frac{TP}{TP + FP}$$

- Recall (Sensitivity): Critical in medical diagnosis to minimize missed cases;

$$\mathbf{3}) \text{ Recall} = \frac{TP}{TP + FN}$$

- F1-score: harmonic mean of precision and recall, balancing both under class imbalance;

$$\mathbf{4}) \text{ F1} = \frac{Precision * Recall}{Peecision + Recall}$$

1.6 Deep Learning and Convolutional Neural Networks

1.6.1 From Machine Learning to Deep Learning

Classical machine learning depends heavily on hand-crafted feature engineering: domain experts design and select the input representations that best capture discriminative data structure. This process is labor-intensive and suboptimal, as it is difficult to anticipate which features will generalize across diverse conditions.

Deep learning addresses this by jointly learning both feature representation and classifier end-to-end from raw data. Multiple layers of nonlinear transformations progressively abstract increasingly complex representations: from low-level edges and textures in early layers to high-level semantic concepts in deeper layers. This hierarchical feature learning is especially advantageous for image data where spatial structure is paramount [20].

1.6.2 Convolutional Neural Networks (CNN)

A CNN is a deep neural network designed for grid-structured data such as images. Three key architectural principles reduce parameters compared to fully connected networks while preserving spatial structure: local connectivity, weight sharing (the same filter applied across all input positions), and hierarchical composition [21].

A typical CNN architecture consists of:

- Convolutional layers: apply learnable filters via discrete convolution to produce activation maps detecting local patterns (typical kernel sizes: 3x3, 5x5);
- Activation functions: nonlinear element-wise operations. ReLU (f(x) = max(0,x)) is most common due to computational efficiency and reduced vanishing gradient;
- Pooling layers: down sample feature maps (max or average pooling), reducing spatial dimensions and providing translational invariance;
- Batch normalization: normalizes layer activations across mini-batches, stabilizing training and allowing higher learning rates;
- Fully connected layers: flatten final feature maps and apply linear transformations to produce class scores or embeddings;
- Dropout: randomly sets a fraction of activations to zero during training, reducing co-adaptation and improving generalization.

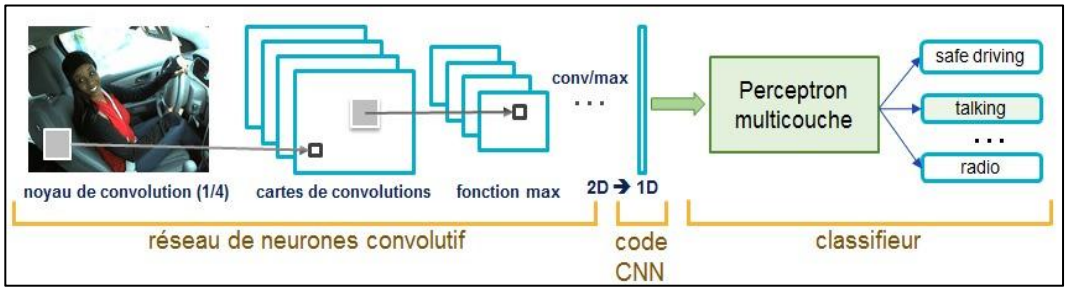

Fig 1.5 General architecture of a Convolutional Neural Network (CNN).

1.7 Conclusion

This chapter established the theoretical foundation for all subsequent work in this thesis. Leukemia was described as a clinically critical hematological malignancy, and automated detection was motivated by highlighting the limitations of manual diagnosis. AI in medicine and CAD systems were surveyed, framing the present

contribution within this broader context. The principles of computer vision preprocessing, segmentation, and augmentation were presented as essential steps for preparing microscopic blood cell images. Finally, the bases of machine learning and deep learning, including CNNs and transfer learning, were established.

These concepts will be directly applied in the subsequent chapters, which detail the methodology, present experimental results, and evaluate the proposed leukemia detection pipeline against established benchmarks.

Chapter 2

Models and Datasets

2.1 Introduction

Building a reliable deep learning system for medical image classification isn't just about choosing a good model it starts much earlier, with the data itself and the architectural decisions that shape how that data gets interpreted. This chapter lays out the full technical foundation of the experimental work, organized around two central pillars : the datasets and the models everything else depends on.

We begin with the two medical imaging datasets at the heart of this research. The first is C-NMC 2019, a well-known benchmark dataset for detecting acute lymphoblastic leukemia from microscopic blood smear images, originally released as part of the IEEE ISBI 2019 challenge. The second is the OCT dataset introduced by Kermany et al. in 2018, which we adapted into a binary classification problem distinguishing eyes with active retinal pathology from healthy ones. For both datasets, we walk through where they come from, how they're structured, how classes are distributed, and what preprocessing choices were made and why.

From there, the chapter turns to the three convolutional neural network architectures used throughout the experiments: MobileNetV2, VGG16, and EfficientNetB0. These weren't picked arbitrarily together they cover a wide range of design philosophies, from the deep, parameter-heavy VGG16 to the lean and mobile-friendly MobileNetV2, with EfficientNetB0 sitting somewhere in between.

The chapter closes with a look at what others have already done in automated leukemia and retinal disease detection using deep learning. This isn't just background reading it helps position this work within the broader research conversation, highlights the gaps that motivated our specific experimental choices, and makes the case for why these particular datasets and architectures were worth pursuing in the first place.

2.2 Data Presentation

Data quality and representativeness are determining factors in any deep learning system. Two public datasets were used in this study, each corresponding to one of the targeted medical domains. These datasets were selected based on their public availability, sufficient size for effective fine-tuning, and their widespread adoption in the scientific community, enabling comparison of obtained results with those reported in existing literature.

2.2.1 Leukemia Dataset: C-NMC 2019

The C-NMC 2019 dataset officially distributed through the Cancer Imaging Archive under the name Pediatric Acute Lymphoblastic Leukemia was put together at the All India Institute of Medical Sciences (AIIMS) in New Delhi, as part of a broader effort to build computer-aided diagnostic tools for blood cancers.

The images themselves come from peripheral blood smear samples taken from pediatric patients with a confirmed diagnosis of acute lymphoblastic leukemia (ALL), alongside samples from healthy children of similar age. Each sample was prepared using the standard Giemsa staining protocol and photographed at 100× magnification under oil immersion, producing high-resolution microscopy images from which individual cells were then segmented out. The final dataset contains 15,135 single-cell images collected from 118 patients, split into two classes: leukemic lymphoblasts (the malignant ALL cells) and hematogones (benign precursor B-cells that look strikingly similar but are perfectly normal).

That last point is what makes this dataset genuinely difficult to work with. The two classes are morphologically so close to each other that even experienced clinicians can struggle to tell them apart which means a model that does well here is actually learning something meaningful, not just picking up on superficial visual differences.

The class distribution adds another layer of challenge: 10,661 ALL-positive images versus just 3,474 normal ones, giving a class imbalance ratio of roughly 3:1. This mirrors the real-world clinical picture, where malignant cells naturally dominate in confirmed leukemia cases. The dataset is split at the patient level 98 patients (13,780 images) go into training, and the remaining 20 patients (1,355 images) form the held-out test set. Keeping patients entirely separate between splits is a deliberate and important choice: it prevents any subtle patient-specific patterns from leaking into the evaluation,

making the reported results genuinely reflective of how the model would perform on new, unseen cases. [12]

Figure 2.1 shows leukemic and normal cell images side by side, and gives a clear sense of just how visually similar these two classes really are.

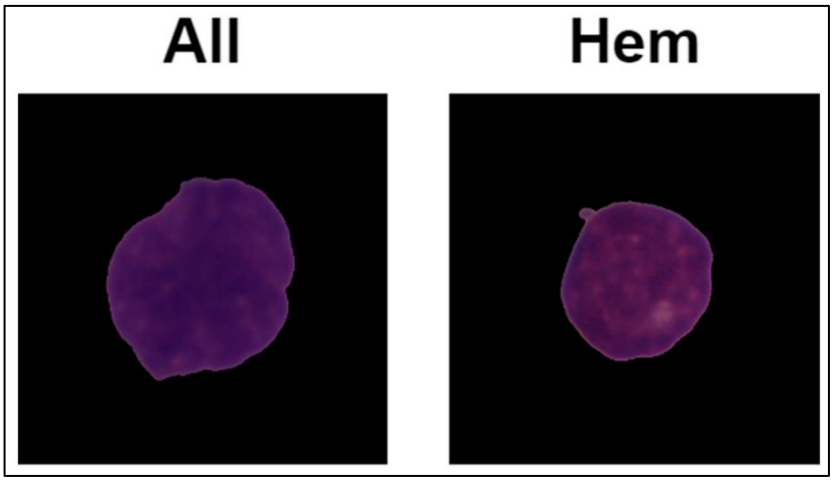

Fig. 2.1 Sample images from the C-NMC 2019 dataset

2.2.2 Retinal Dataset: Kermany OCT 2018

The OCT dataset from Kermany et al. has become something of a standard reference point in retinal image analysis most papers in this space cite it, build on it, or at least compare against it. It's publicly available through the Cancer Imaging Archive and covers 84,484 B-scan images collected from 4,686 patients, all scanned with the Spectralis system from Heidelberg Engineering. That particular device uses spectral-domain OCT, which gives it strong axial resolution relevant here because subtle retinal changes are exactly what the model needs to pick up on. [20]

The original dataset splits images into four groups: CNV, DME, drusen, and normal retinas. CNV is when new, abnormal blood vessels grow under the retina. DME is fluid accumulation in the macular layers, typically as a downstream effect of diabetic retinopathy. Left untreated, both can cause serious, irreversible damage to vision.

For this study, the dataset was reshaped into a two-class problem. Drusen images were dropped entirely, and CNV and DME were grouped together under one label Pathological. The reasoning here is clinical rather than computational: these two conditions may develop differently, but the response they call for is the same. Whether a scan shows CNV or DME, a clinician needs to act fast, usually with anti-VEGF

injections or laser treatment. Separating them in a triage model would add complexity without adding much practical value. [4],[5]

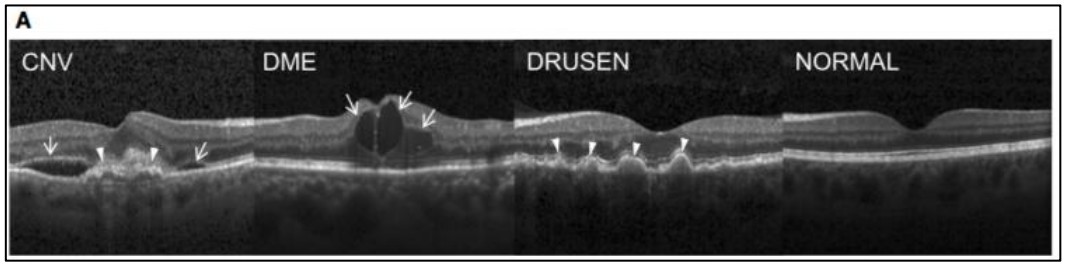

Fig.2.2 Sample images from OCT Kermany dataset

2.3 Model Architectures

Three pretrained convolutional neural network architectures are presented in this chapter. These models represent different design philosophies in deep learning for image analysis, ranging from deep and dense architectures such as VGG16, to lightweight models optimized for efficiency like MobileNetV2, as well as compound-scaled architectures that balance performance and computational cost such as EfficientNetB0. Each of these architectures has been widely adopted in computer vision tasks due to its specific structural characteristics and effectiveness.

2.3.1 MobileNetV2

Mobilenet, introduced by Howard et al. in 2017 at Google, is a convolutional neural network architecture designed for efficient computation, particularly on mobile and embedded devices. It is based on the concept of depthwise separable convolutions, which decompose a standard convolution into two operations: a depthwise convolution applied independently to each channel, followed by a pointwise (1×1) convolution that combines the resulting features. This approach significantly reduces computational cost and the number of parameters while maintaining strong performance.

MobileNetV2 extends this design by introducing inverted residual blocks with linear bottlenecks. In this structure, shortcut connections are applied between compressed representations, while intermediate layers expand the feature space, improving information flow and efficiency. MobileNetV2 contains approximately **3.4 million parameters**, making it a compact architecture suitable for real-time applications and devices with limited computational resources.[24][25]

The architecture of MobileNetV2, highlighting its key components, including depthwise separable convolutions and inverted residual blocks is illustrated in **Fig 2.3**

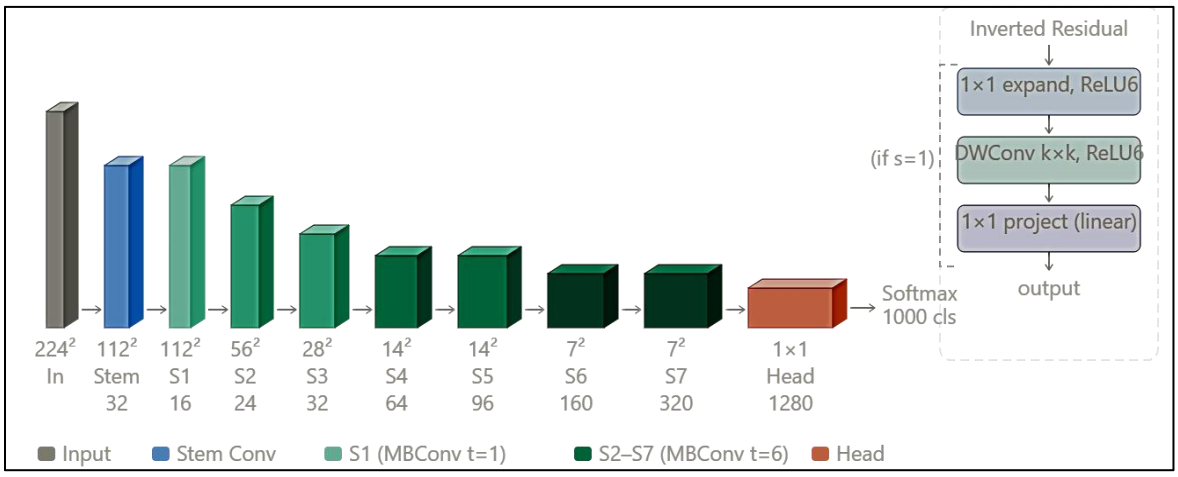

Fig. 2.3 MobileNetV2 architecture: inverted residual blocks with depthwise separable convolutions.

2.3.2 VGG16

VGG16, proposed by Simonyan and Zisserman at the Visual Geometry Group of the University of Oxford in 2014, is a deep convolutional neural network characterized by a simple and uniform architecture. It consists of 13 convolutional layers using small 3×3 filters, followed by 3 fully connected layers. This design choice allows the network to increase depth while maintaining manageable computational complexity and a consistent structure.

The architecture is organized in a sequential manner, where convolutional layers are grouped into blocks separated by max-pooling operations. As the depth increases, the network progressively learns hierarchical feature representations, starting from low-level features such as edges and textures to more complex and abstract patterns.

VGG16 contains approximately 138 million parameters, making it a large-scale model with high representational capacity. Its architectural simplicity and effectiveness have made it a widely used reference model in image classification tasks. [26]

Fig 2.4 illustrates the architecture of VGG16, showing its sequential arrangement of convolutional and fully connected layers.

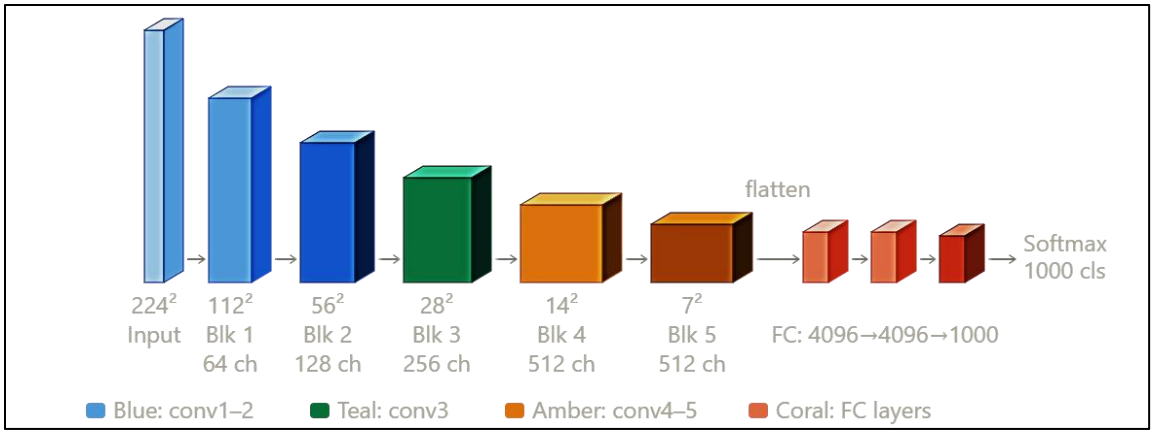

Fig. 2.4 VGG16 architecture: uniform stack of 3×3 convolutional layers followed by three fully connected layers.

2.3.3 EfficientNetB0

EfficientNet, introduced by Tan and Le in 2019 at Google Brain , proposes a systematic compound scaling approach that simultaneously and proportionally adjusts network depth, width (number of channels), and input image resolution using a single scaling coefficient. This method, optimized through neural architecture search (NAS), enables the model to achieve high accuracy while maintaining a relatively low number of parameters.

EfficientNetB0 is the baseline variant of this family, with approximately 5.3 million parameters and an input resolution of 224×224 pixels. It integrates mobile inverted bottleneck convolution (MBConv) blocks along with squeeze-and-excitation mechanisms, allowing adaptive channel-wise feature recalibration. This design enhances the network's ability to capture fine-grained and relevant features in image data while preserving computational efficiency [27], illustrating its architecture in **fig 2.5** .

Comparison of the three CNN architectures used in this study: parameters, input resolution, and key innovation shown as **Table 2.1**

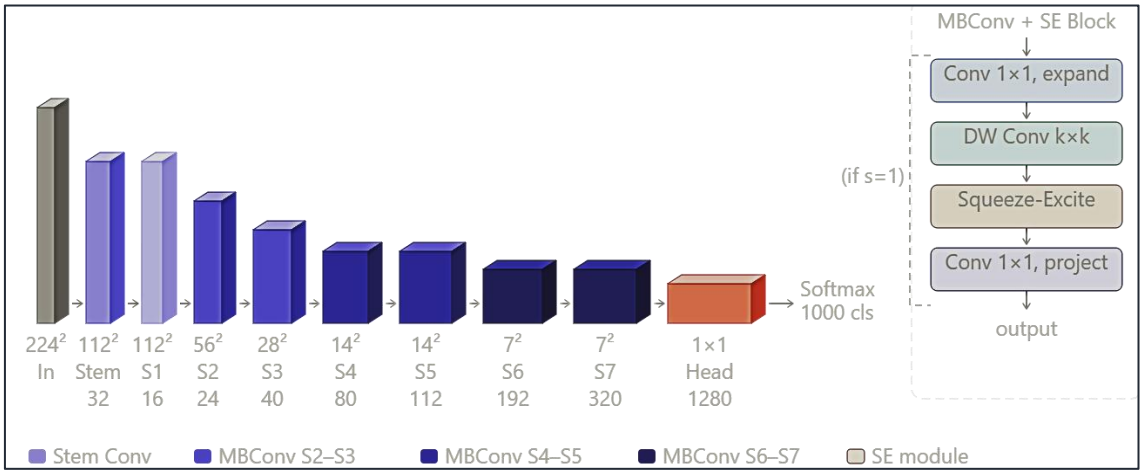

Fig. 2.5 - EfficientNetB0 architecture: compound scaling (depth, width, resolution) and MBConv blocks with residual connections.

Table 2.1. Comparison of the three CNN architectures used in this study

Architecture	Year	Parameters	Input Resolution	Depth	Key Innovation
VGG16	2014	138 M	224×224	16 layers	Uniform stacked 3×3 convolutions
MobileNetV2	2018	3.4 M	224×224	53 layers	Depthwise separable convolutions + inverted residual blocks
EfficientNetB0	2019	5.3 M	224×224	82 layers	Compound scaling depth/width/resolution

2.4 Related Works

The application of deep learning to medical image classification has grown rapidly over the past decade, driven by the availability of large annotated datasets and the maturation of convolutional neural network architectures. In the context of hematological and ophthalmological imaging, numerous studies have demonstrated that CNN-based models can achieve diagnostic performance comparable to or exceeding that of trained specialists. This section reviews the most relevant published contributions in two areas directly related to this work: automated leukemia detection from blood smear images, and retinal disease classification from OCT scans. The reviewed works are summarized in **Table 2.2** , which highlights the datasets, architectures, and key findings reported in each study.[28][29]

Table 2.2. Summary of related works in deep learning for medical image classification.

Authors	Year	Task	Dataset	Method	Key Finding
Kermany et al. [22]	2018	Retinal disease classification	Kermany OCT	Inception V3 fine-tuned	Transfer learning achieves specialist-level OCT diagnosis
Gupta et al. [12]	2019	Leukemia detection	C-NMC 2019	ResNet-50, DenseNet	CNN baseline established on ISBI 2019 challenge dataset
Matek et al. [9]	2019	Blood cell morphology	AML dataset	Custom CNN	Human-level recognition of blast cells demonstrated
Esteva et al. [33]	2017	Skin lesion classification	ISIC	GoogLeNet Inception	First CNN shown to match dermatologist diagnostic performance
Gulshan et al. [31]	2016	Diabetic retinopathy	EyePACS	Inception V3	Deep learning shown to surpass ophthalmologist screening accuracy
Alom et al. [34]	2019	Leukemia classification	ALL-IDB	Inception ResNet V2	High sensitivity achieved on balanced leukemia benchmark
Das et al. [35]	2021	ALL detection	C-NMC 2019	VGG16 + SVM hybrid	Classical ML on CNN features shown to improve classification
Wang et al. [36]	2020	Leukemia detection	C-NMC 2019	EfficientNet variants	Compound scaling identified as beneficial for blood cell analysis
Talo et al. [37]	2019	OCT classification	Kermany OCT	ResNet-50	Deep residual networks found effective for retinal layer analysis
Togacar et al. [38]	2020	Leukemia classification	ALL-IDB	MobileNet + SVM	Lightweight architectures shown competitive for medical imaging
Li et al. [39]	2019	Retinal OCT analysis	Kermany OCT	Attention CNN	Attention mechanisms shown to improve fine-grained retinal features
Rehman et al. [40]	2021	Leukemia detection	Custom dataset	VGG16, AlexNet	Transfer learning consistently outperforms training from scratch

2.5 Conclusion

This chapter established the experimental foundation of the present work by presenting the two medical imaging datasets and the three convolutional neural network architectures employed in the subsequent experiments. The C-NMC 2019 dataset and the Kermany OCT dataset were described in detail, including their composition, class distributions, and the preprocessing decisions applied. Three architectures of complementary design philosophies, namely VGG16, MobileNetV2, and EfficientNetB0, were then introduced, with their core principles and pretrained weight configurations specified. Finally, a review of related works confirmed the relevance of the architectural

and methodological choices adopted, situating the present contribution within the existing landscape of deep learning for medical image classification. The next chapter details the implementation of the experimental framework, including the data preparation procedures, the training pipeline, and the constraints encountered during execution.

Chapter 3

Implementation

3.1 Introduction

This chapter presents the implementation framework adopted in the present work. Three main challenges directly shaped the choices described here: the computational constraints of Kaggle's 12-hour session limit, which required a per-epoch checkpointing strategy; the restructuring of the OCT dataset from a four-class to a binary problem; and the difficulty of reproducing published results on the Leukemia CNMC dataset, where accuracy remained near 80% despite multiple configurations a limitation discussed honestly rather than concealed.

The chapter is organized as follows: Section 3.2 describes the proposed methodology and experimental pipeline. Section 3.3 details the training parameters. Section 3.4 explains the dataset preparation decisions. Results and comparative analyses are presented in Chapter 4.

3.2 Development Environment Overview

3.2.1 Programming language

- **Python**

Python is widely used in machine learning and deep learning, and for good reason. Its clean and simple syntax makes it easy to learn and work with, which is especially helpful when building and testing complex models. A key strong point of Python is its huge selection of libraries, which include TensorFlow, PyTorch,Scikit-learn, and Keras These libraries offer pre-written code and functions that save time and make the development process easier.

Python also has a huge community of developers and researchers, which means there’s plenty of documentation, tutorials, and support available online. Python offers simplicity, flexibility, and strong tools, all of which render it an optimal language for machine learning and deep learning projects. [42][43]

3.2.2 Libraries Used

Table 3.1 Python Libraries used .

TensorFlow	An open-source framework by Google for building and training machine learning and deep learning models. It handles everything from defining neural network layers to running GPU computations	`import tensorflow as tf`
Keras	A high-level API built on top of TensorFlow that simplifies model building. It lets you stack layers, compile, and train models with just a few lines of code.	`import tensorflow.keras as keras`
scikit-learn	A general-purpose ML library offering ready-to-use classical algorithms (SVM, Random Forest, etc.), plus tools for evaluation like confusion matrices and classification reports. Used in your Method 4 notebooks.	`import sklearn`
NumPy	The foundation of scientific computing in Python. It provides fast multi-dimensional arrays and mathematical operations, used heavily under the hood by all other libraries.	`import numpy as np`
pickle	Saves and loads Python objects to/from disk. Used to checkpoint training progress between Kaggle sessions (since sessions are limited to 12 hours).	`import pickle`

3.2.3 Development Setup

- **Visual Studio Code**

Visual Studio Code (VS Code) is free, open-source software meant for coding, developed by Microsoft. Its support many programming languages such as Python, C++, JavaScript, etc. Equipped with syntax highlighting, code completion (IntelliSense), a debugging tools, and a customizable user interface, and an integrated terminal.

VS Code is very extensible and its capability can, therefore, be enhanced by any user with thousands of extensions that support frameworks, debugger.

It has support for Markdown to style text, allowing users to document alongside their code. With its lightweight might combined with powerful development features, Visual Studio Code is among the most popular editors for web developers, data scientists, DevOps engineers, and others.

- **Kaggle Notebook**

All training experiments were executed on Kaggle Notebooks, a cloud-based environment that provides free access to GPU accelerators (NVIDIA T4 or P100) with sessions limited to 12 hours. Kaggle also hosts both datasets used in this work, enabling direct dataset mounting without manual downloads. The GPU access and integrated dataset management made Kaggle the natural choice for this project.

3.3 Proposed Methodology

3.3.1 Overall Experimental Pipeline

A critical engineering requirement was checkpointing. Kaggle sessions disconnect after 12 hours maximum. To handle this, every notebook saves model weights (.h5) and a training history dictionary (pickle) after every epoch. On the next session, the notebook detects the checkpoint file, reloads weights and history, and resumes from the interrupted epoch seamlessly. This was essential for completing 15-epoch training runs across all configurations.

The experimental framework adopted in this work follows a systematic pipeline. Each of the models configurations (3 architectures x 4 methods x 2 datasets) are implemented as an independent Jupyter notebook on Kaggle. All backbones are loaded with ImageNet-pretrained weights from tf.keras.applications. The same classifier head is attached to every architecture: GlobalAveragePooling2D reduces the spatial feature map to a 1D vector, BatchNormalization stabilizes activations, Dense(256, ReLU) learns task-specific combinations, and Dense(1, sigmoid) produces the binary probability output. Training uses binary cross-entropy loss and the Adam optimizer . Images are loaded via ImageDataGenerator at 224x224 pixels, normalized to [0,1].

3.3.2 Fine-Tuning Strategy

Transfer learning reuses a model pretrained on a large source dataset as the starting point for a new task. The simplest form, used in Method 1, freezes the backbone entirely

and trains only a new classifier head fast and stable, but limited since the backbone cannot adapt its representations to the target domain.

Fine-tuning goes further by selectively unfreezing backbone layers and allowing them to update on the target data. The strategy adopted here (Method 2) uses a two-phase procedure to prevent catastrophic forgetting, where aggressive weight updates destroy the general visual representations acquired during pretraining.

Fig 3.1 is an illustration of the two-phase fine-tuning strategy: Phase 1 trains only the classifier head with the backbone frozen; Phase 2 partially unfreezes the backbone and retrains end-to-end at a lower learning rate.

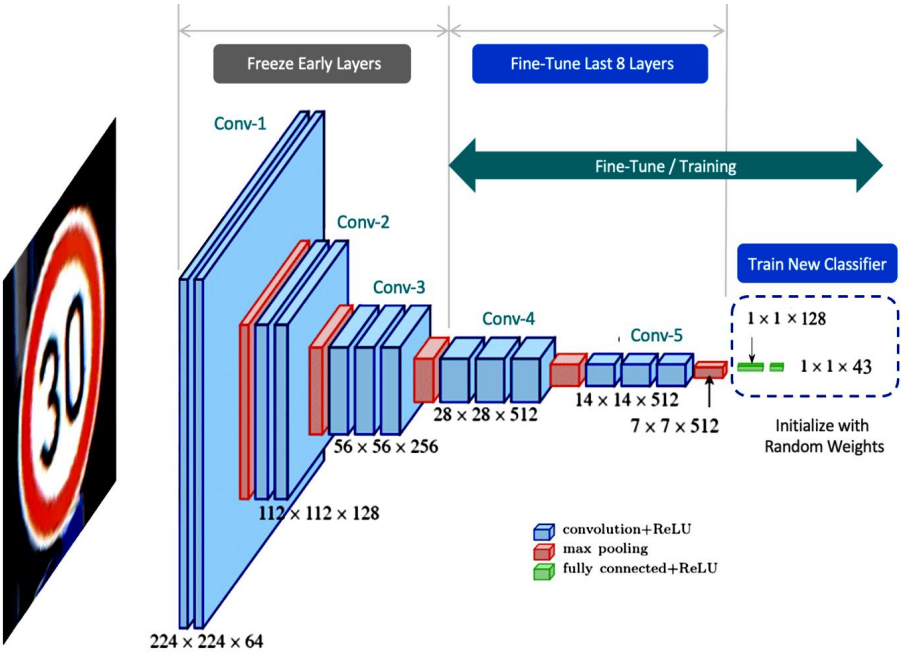

Fig 3.1 fine-tuning Pre-trained models

Phase 1 trains the classifier head with the backbone completely frozen for 15 epochs at lr=1×10^{-3}, establishing stable task-specific weights. **Phase 2** unfreezes the top layers the last 4 for VGG16, from layer 100 for MobileNetV2, and the last 20 for EfficientNetB0 and retrains end-to-end for 15 epochs at lr=1×10^{-5}, allowing gradual domain adaptation without overwriting low-level pretrained features.

3.3.3 Data Splitting

For OCT, the original train/validation/test split from Kermany et al. (2018) was used, after applying the binary restructuring and undersampling described above. For

Leukemia CNMC, the pre-defined patient-level folds were used: fold_0 for training, fold_1 for validation, fold_2 for testing. The test set was balanced to 1,096 samples per class.

3.4 Training Parameters and Code

3.4.1 First Method (Base Models)

The backbone is fully frozen. Only the custom classifier head is trained for 15 epochs with Adam (lr=1×10^{-3}), batch size 32. A per-epoch pickle checkpointing loop allows Kaggle session recovery, thse configurations are applied across the 3 models .

- **EfficientNetB0 Model**

```
from tensorflow.keras.applications import EfficientNetB0
```

```
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224,224,3))
```

```
base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, batch_size=32, validation_data=val_gen)
```

- **MobileNetV2 Model**

```
from tensorflow.keras.applications import MobileNetV2
```

```
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224,224,3))
```

```
base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, batch_size=32, validation_data=val_gen)
```

- **VGG16 Model**

```
from tensorflow.keras.applications import VGG16
```

```
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,224,3))
```

```
base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, batch_size=32, validation_data=val_gen)
```

3.4.2 Method 2 (Two-Phase Fine-Tuning)

Phase 1 trains the head with frozen backbone (15 epochs, lr=1e-3) . Phase 2 unfreezes the top layers and retrains (15 epochs, lr=1e-5). Each phase has its own checkpoint prefix .

- **Phase1:**

```
base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, validation_data=val_gen)
```

- **Phase 2 :**
  - **EfficientNetB0 Model :** The last 20 layers are unfrozen the top MBConv blocks. The model is recompiled at lr=1e-5, 100× lower than Phase 1, to adapt those layers gently to medical image patterns without overwriting the ImageNet weights.

```
base_model.trainable = True
for layer in base_model.layers[:len(base_model.layers)-20]: layer.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-5), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, validation_data=val_gen)
```

  - **MobileNetV2 Model :** All layers from index 100 onwards are unfrozen the final inverted residual blocks. Everything before stays frozen to preserve the low-level features. Same reduced lr=1e-5 to keep updates small and stable.

```
base_model.trainable = True
for layer in base_model.layers[:100]: layer.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-5), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, validation_data=val_gen)
```

- **VGG16 Model :** Only the last 4 layers are unfrozen(from the Block 5). With 138M parameters, unfreezing more would risk overfitting on our small medical datasest. lr=1e-5 keeps the fine-tuning controlled and precise.

```
base_model.trainable = True
for layer in base_model.layers[:len(base_model.layers)-4]: layer.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-5), loss='binary_crossentropy', metrics=['accuracy'])
model.fit(train_gen, epochs=15, validation_data=val_gen)
```

3.4.3 Method 3 (Data Augmentation)

Same frozen backbone as Method 1, but with heavy augmentation on the training generator only. Validation and test generators use rescaling only.

a) Each image is randomly rotated up to 20°, shifted horizontally and vertically by 15%, flipped in both directions, zoomed by 15%, and brightness is varied between 80% and 120% of its original value. The model never sees the exact same version of an image twice.

```
train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=20, width_shift_range=0.15,
    height_shift_range=0.15, horizontal_flip=True, zoom_range=0.15, brightness_range=[0.8,1.2])
```

b) The validation and test generators apply only pixel rescaling dividing every pixel value by 255 to bring them into the [0, 1] range.

```
val_datagen = ImageDataGenerator(rescale=1./255)
```

```
model.fit(train_gen, epochs=15, validation_data=val_gen)
```

3.4.4 Method 4 (Hybrid CNN + Classical ML)

The frozen backbone is used as a pure feature extractor. Features are normalized with StandardScaler, then SVM (RBF, C=10) and Random Forest (200 trees) are trained. The best is selected automatically.

a) We built the feature extractor by stacking the frozen backbone with a GlobalAveragePooling2D layer inside a Sequential model not the classifier head. GlobalAveragePooling2D collapses the spatial feature map into a single fixed-length vector by averaging across all positions. The result is a compact representation per image: 1280 values for EfficientNetB0 and MobileNetV2, and 512 for VGG16.

```
extractor = Sequential([base_model, GlobalAveragePooling2D()])
```

b) predict() runs every image through the extractor and returns its feature vector. X_train and X_test are two arrays one row per image where each row is the compressed representation of that image as learned by the backbone. The labels y_train and y_test come directly from the generator's class indices, which correspond to the folder names.

```
X_train = extractor.predict(train_gen)    y_train = train_gen.classes
X_test = extractor.predict(test_gen)     y_test = test_gen.classes
```

c) Before passing the vectors to any classifier we normalize them with StandardScaler . The scaler is fitted only on X_train only. Because fitting it on the test set would leak information about its distribution into the training process and that will increasethe results .

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

d) The first classifier is an SVM with an RBF kernel. The RBF kernel maps the feature vectors into a higher-dimensional space where a linear boundary can separate the two classes handling the non-linearity that a straight line cannot. C=10 means the model prioritizes correct classification while leaving room to generalize. gamma='scale' automatically adjusts the kernel width based on the number of input features.

```
svm = SVC(kernel='rbf', C=10, gamma='scale')
svm.fit(X_train, y_train)
```

e) The second classifier is a Random Forest with 200 decision trees. Each tree is trained on a random subset of the training samples and features, making every tree different from the others. The final prediction is a majority vote across all 200 trees, which makes the model robust to noise in the feature vectors. random_state=42 fixes the randomness .

```
rf = RandomForestClassifier(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)
```

3.5 Dataset Preparation and Restructuring

3.5.1 OCT Binary Dataset Construction

The original Kermany OCT dataset contains four classes. It was restructured into a binary task for the following reasons:

CNV and DME were merged into a single DISEASE class both represent active pathologies requiring treatment. DRUSEN was completely removed: DRUSEN deposits sit beneath the retinal pigment epithelium and produce an OCT appearance that closely resembles a healthy retina in terms of overall layer architecture. Including DRUSEN as a "disease" class would confuse the model and introduce systematic label noise at the DISEASE/NORMAL boundary. This decision improved training stability significantly.

```
SRC = '/kaggle/input/kermany2018/OCT2017 /'
DST = '/kaggle/working/oct_binary/'

targets = {'train': 21052, 'val': 5263, 'test': 242}
```

```
class_map = {
    'DISEASE': ['CNV', 'DME'],
    'NORMAL' : ['NORMAL']
}
```

The remaining class imbalance (DISEASE/NORMAL) was resolved by random undersampling of the majority class, making a perfectly balanced dataset presented in the **table 3.1** :

Table 3.2 OCT Binary dataset distribution after restructuring and balancing.

Split	DISEASE	NORMAL	Total
Train	21,052	21,052	42,104
Validation	5,263	5,263	10,526
Test	242	242	484

3.5.2 Leukemia CNMC Dataset

The C-NMC 2019 dataset was used with its original patient-level fold structure. The training set shows a ~2:1 ALL/HEM imbalance. Since the minority class (HEM) contains only 1,130 training images as the **table 3.2** shows, undersampling would reduce

training data to an unworkably small size. The natural imbalance was therefore retained during training, and only the test set was balanced:

Table 3.3 Leukemia CNMC dataset distribution and evaluation strategy.

Split	ALL	HEM	Total	Evaluation
Train (fold_0)	2,397	1,130	3,527	Imbalanced
Validation (fold_1)	2,418	1,163	3,581	Imbalanced
Test (fold_2)	1,096	1,096	2,192	Balanced

3.6 Conclusion

This chapter presented the implementation framework adopted in the present work. The proposed methodology, the experimental pipeline, the training parameters, and the dataset preparation procedures applied to both the OCT Binary and Leukemia CNMC datasets were described in detail. The transparency of the implementation choices, including the constraints imposed by the Kaggle execution environment and the empirical justifications behind the OCT binary restructuring, ensures the reproducibility of the experiments. The next chapter presents the experimental results obtained on both datasets, including a comparative analysis of the architectures and methods evaluated, followed by a general discussion of the findings.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter brings together all the experimental results from the implementation work described in **Chapter 3**. In total, all model configurations were evaluated spanning three architectures (VGG16, EfficientNetB0, and MobileNetV2), four training strategies (base model, fine-tuning, augmentation, and ML classifier), and two datasets (OCT Binary and Leukemia CNMC).

The chapter is organized as follows. **Section 4.2** covers the OCT Binary classification results, going through each architecture in turn before wrapping up with a side-by-side comparison. **Section 4.3** does the same for the Leukemia CNMC dataset, following the same structure. **Section 4.4** steps back and looks at the bigger picture pulling together findings from both datasets, spotlighting the architecture-method combinations that worked best, and examining why the two tasks ended up producing such different levels of performance. **Section 4.5** closes the chapter.

4.2 OCT Binary Classification Results

All OCT configurations were evaluated on the balanced test set (484 images: 242 DISEASE + 242 NORMAL). Results are reported as macro-averaged metrics.

4.2.1 VGG16 Model

- **Method 1 (Base Model) :**

VGG16 with the backbone entirely frozen, VGG16 achieved **69.00% accuracy**. The training curves on **fig 4.1** show slow convergence the ImageNet features, while rich, are not directly suited to the structural patterns of OCT retinal scans. The large parameter count of VGG16 (138M) also means the classifier head struggles to find an adequate mapping without backbone adaptation.

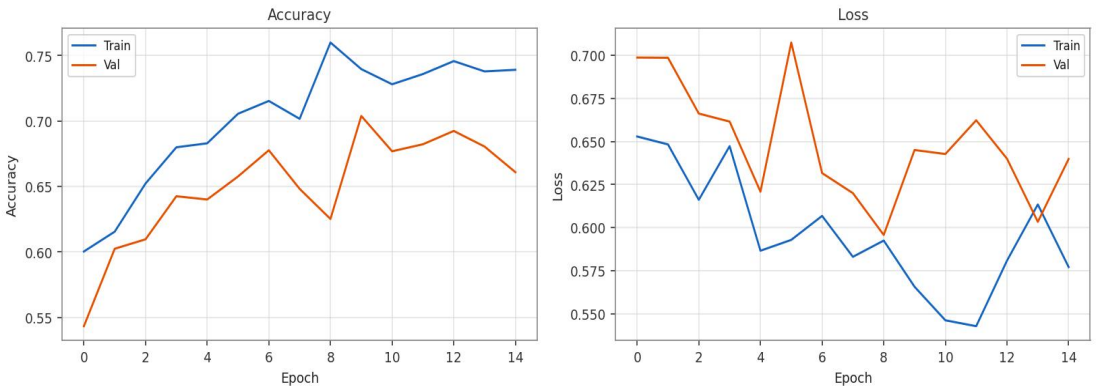

Fig 4.1 OCT VGG16 Method 1 (Base Model): Training and validation curves.

The confusion matrix (**fig 4.2**) makes the failure mode explicit: out of 242 NORMAL images, 128 were incorrectly predicted as DISEASE a false positive rate of 53%. The model is heavily biased toward the pathological class. DISEASE recall is reasonable at 91% (220/242 correctly identified), but NORMAL recall collapses to just 47% (114/242), making the model essentially unreliable for healthy patients. This class bias is a direct consequence of the frozen backbone's inability to distinguish retinal structural patterns from its ImageNet-learned features.

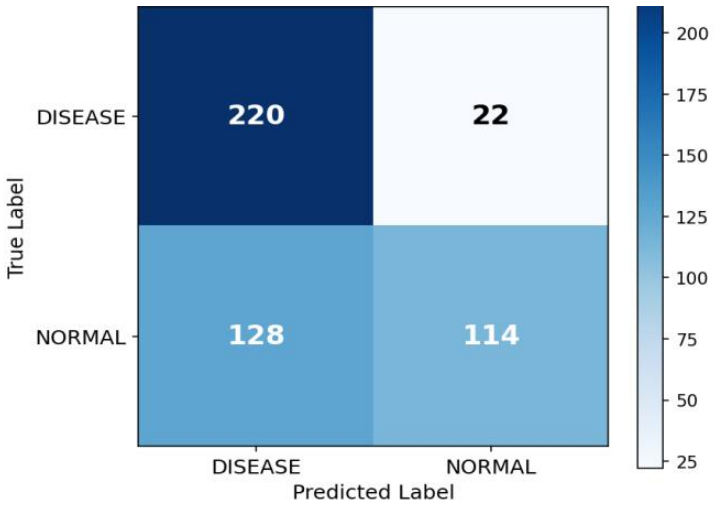

Fig. 4.2 OCT VGG16 Method 1: Confusion matrix.

- **Method 2 (Fine-Tuning) :**

Unlocking Block 5 of VGG16 and retraining at a low learning rate (1e-5) produced a dramatic turnaround. In **Phase 1**, both training and validation accuracy rise sharply and in close alignment from around 58% at epoch 0 to roughly 91% by epoch 14 while loss drops steeply from 0.70 to around 0.22. This clean convergence confirms that the classifier head, on its own, can adapt well when given enough signal.

Phase 2 introduces more nuance. Training accuracy continues climbing toward 97%, but validation oscillates between 89% and 93%, with a noticeable dip near epoch 2 that what shows **fig 4.3** . The widening gap between the two curves signals mild overfitting pressure from the unlocked convolutional layers expected given VGG16's 138 million parameters but it remains controlled and doesn't derail performance.

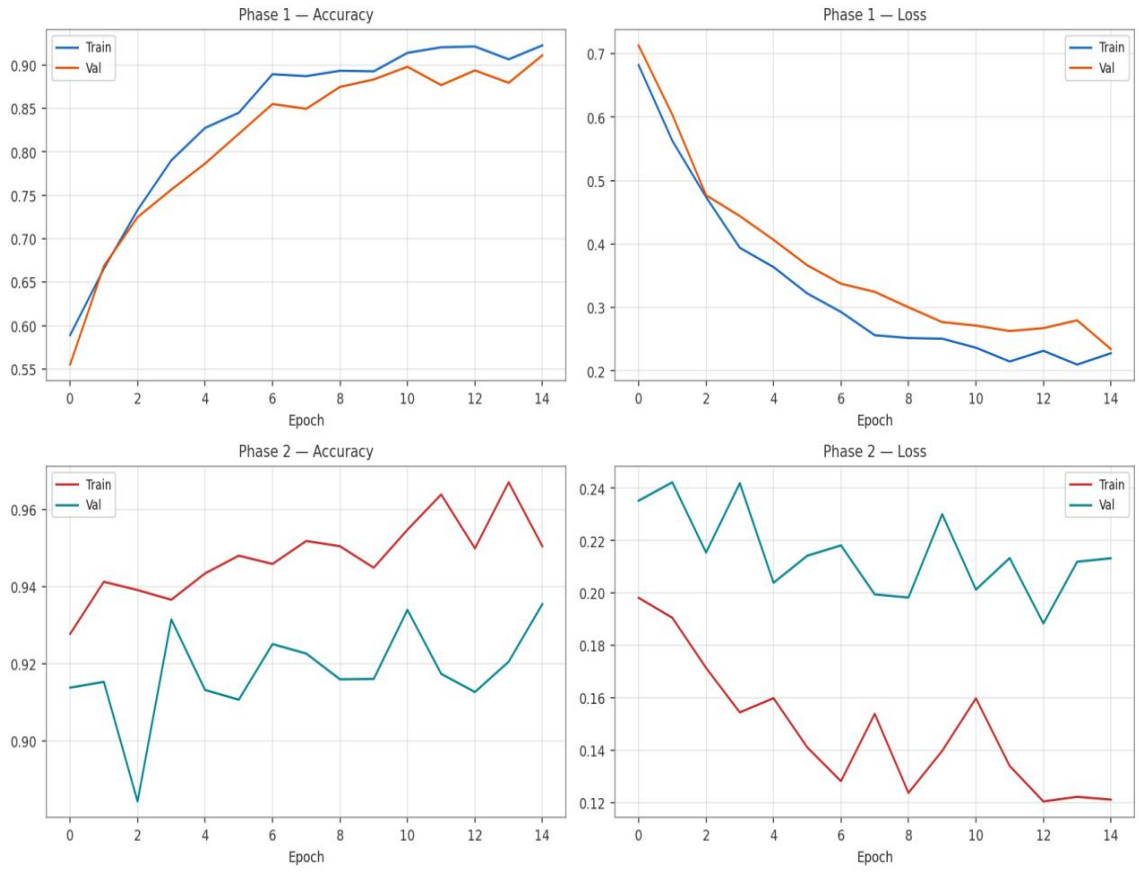

Fig. 4.3 OCT VGG16 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

The confusion matrix **(fig 4.4)** reflects a fundamentally corrected model: 225 DISEASE correctly identified (93% recall), 218 NORMAL correctly identified (90% recall), with only 17 missed DISEASE cases and 24 false positives on NORMAL. The severe class bias of Method 1 is gone.

Fig. 4.4 OCT VGG16 Method 2: Confusion matrix.

- **Method 3 (Data Augmentation) :**

VGG16 hits its best accuracy in this method. The training curves **fig 4.5** show both accuracy lines tracking each other almost perfectly from start to finish rising together from 56% to 93-94% with a loss drop that is smooth and well-behaved on both sides. The tight alignment between train and validation throughout all 15 epochs indicates that the augmentation is genuinely helping the model generalize, rather than just memorizing the training set.

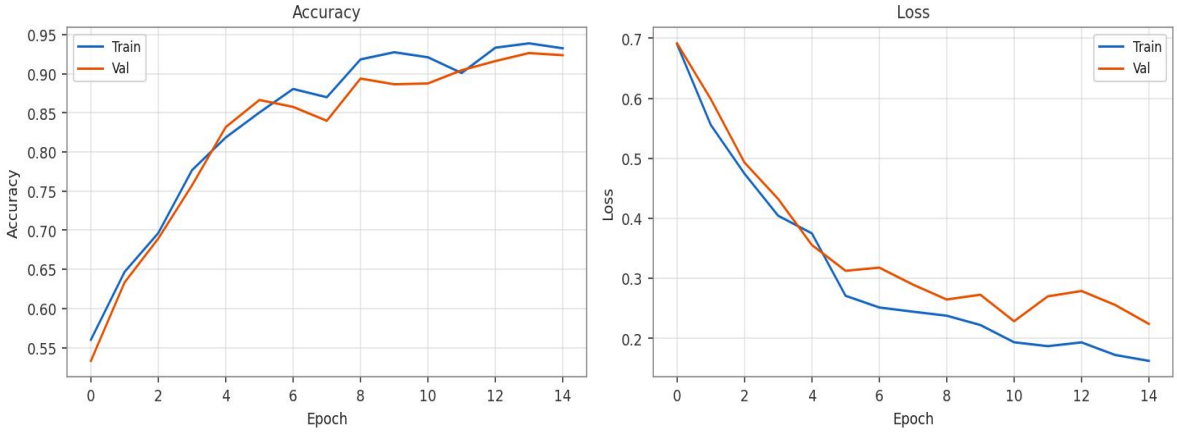

Fig. 4.5 OCT VGG16 Method 3 (Augmentation): Training and validation curves.

The confusion matrix **(fig 4.6)** shows a slight improvement over fine-tuning: 228 DISEASE correctly detected (94% recall, up from 93%), with the same NORMAL performance (218/242, 90%). The 14 missed DISEASE cases versus 17 in fine-tuning might seem marginal, but the real advantage here is training stability no Phase 2

instability, no overfitting pressure, a frozen backbone that simply generalizes better with more diverse input.

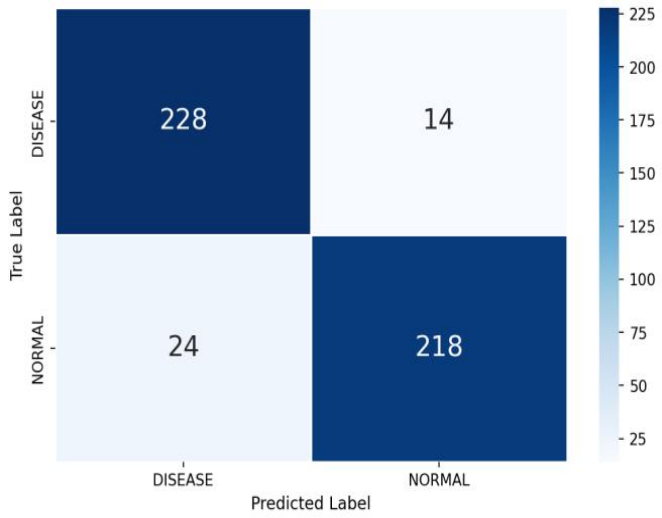

Fig. 4.6 OCT VGG16 Method 3: Confusion matrix.

- **Method 4 (ML Classifier) :**

The SVM operating on 512-dimensional VGG16 frozen features edges out augmentation to become the best VGG16 result. Without any iterative training involved, the confusion matrix shows 229 DISEASE correctly identified (94.6% recall) and 218 NORMAL correctly identified (90% recall), with only 13 DISEASE missed and 24 NORMAL misclassified. The SVM's ability to find a global maximum-margin boundary in the high-dimensional feature space appears to outperform the simple dense head a recurring advantage that will be observed across architectures.

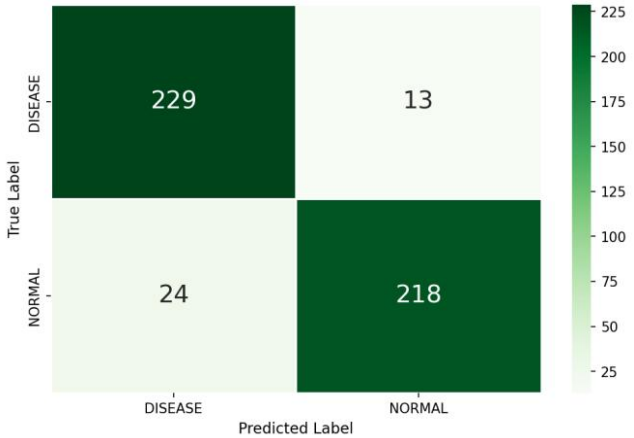

Fig. 4.7 OCT VGG16 Method 4 (ML Classifier): Confusion matrix.

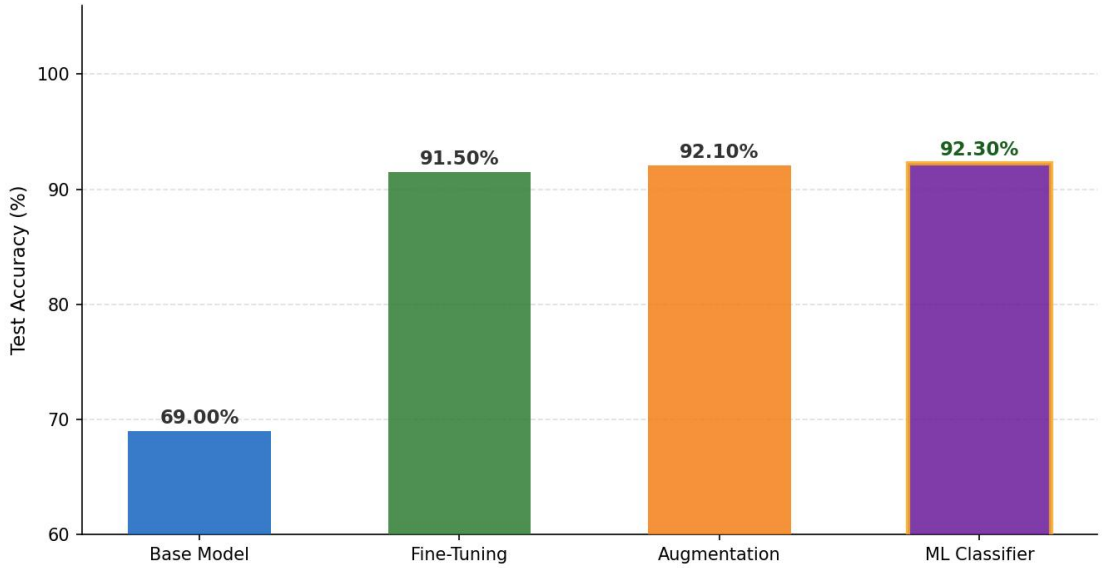

Fig. 4.8 OCT VGG16: Accuracy comparison across all methods.

4.2.2 EfficientNetB0

- **Method 1 Base Model :**

EfficientNetB0 starts roughly 10 points above VGG16 under identical frozen conditions a meaningful gap that reflects a real architectural difference. The training curve shows accuracy climbing from 62% to around 82%, but validation is noticeably noisier, oscillating between 72% and 80% without fully stabilizing as it shown in **fig 4.9**. The gap between training and validation widens in later epochs, and validation loss plateaus around 0.50 while training loss drops to 0.44 a sign that the frozen features are partially useful but still not optimally aligned with retinal imaging.

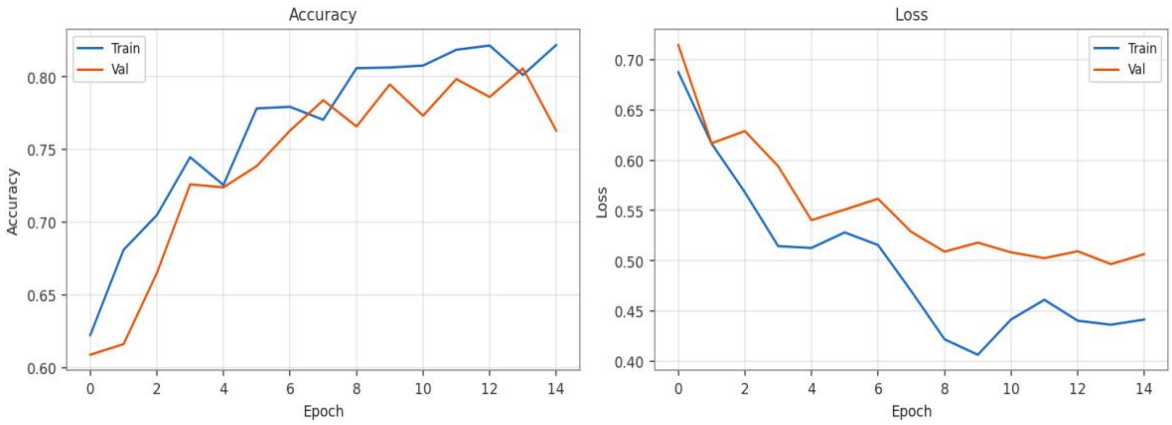

Fig. 4.9 OCT EfficientNetB0 Method 1 (Base Model): Training and validation curves.

The confusion matrix **fig 4.10** reveals the remaining limitation: while DISEASE recall is reasonable at 87% (210/242), NORMAL recall is only 71% (172/242), with 70 false positives on the healthy class. The channel-wise attention from squeeze-and-excitation blocks gives EfficientNetB0 a stronger baseline than VGG16, but the frozen features still struggle to separate fine retinal texture differences reliably.

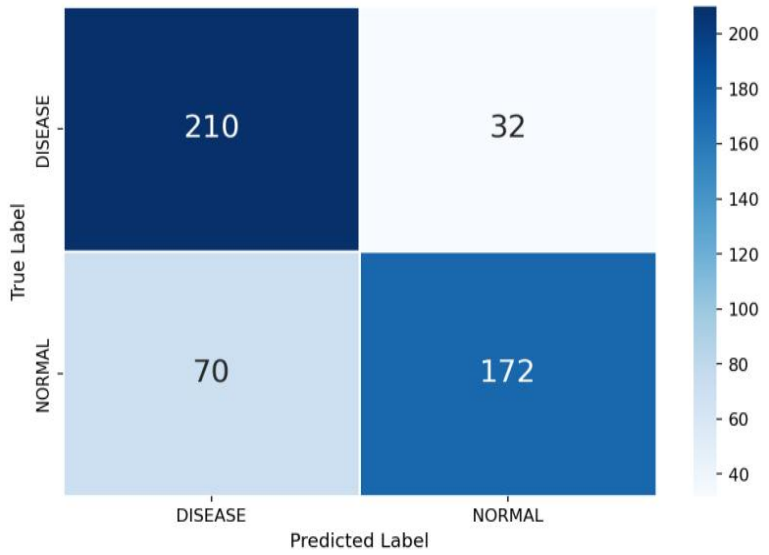

Fig. 4.10 OCT EfficientNetB0 Method 1: Confusion matrix.

- **Method 2 Fine-Tuning :**

This is the best result across all 24 configurations, and the training curves show why it worked so well. **Phase 1** is remarkable from the very first epoch both training and validation accuracy begin above 96% and track each other closely throughout, with loss already below 0.12. This means EfficientNetB0's pretrained features were already performing near-optimally before any backbone layers were unlocked a clear sign of superior transferability compared to VGG16.

Phase 2 pushes further: training accuracy approaches 100% while validation holds steadily between 98% and 99.3%. Crucially, the validation loss remains flat around 0.07 throughout Phase 2, even as training loss drops near zero. This absence of overfitting despite the model approaching perfect training performance is what makes this result clinically meaningful rather than just statistically impressive.

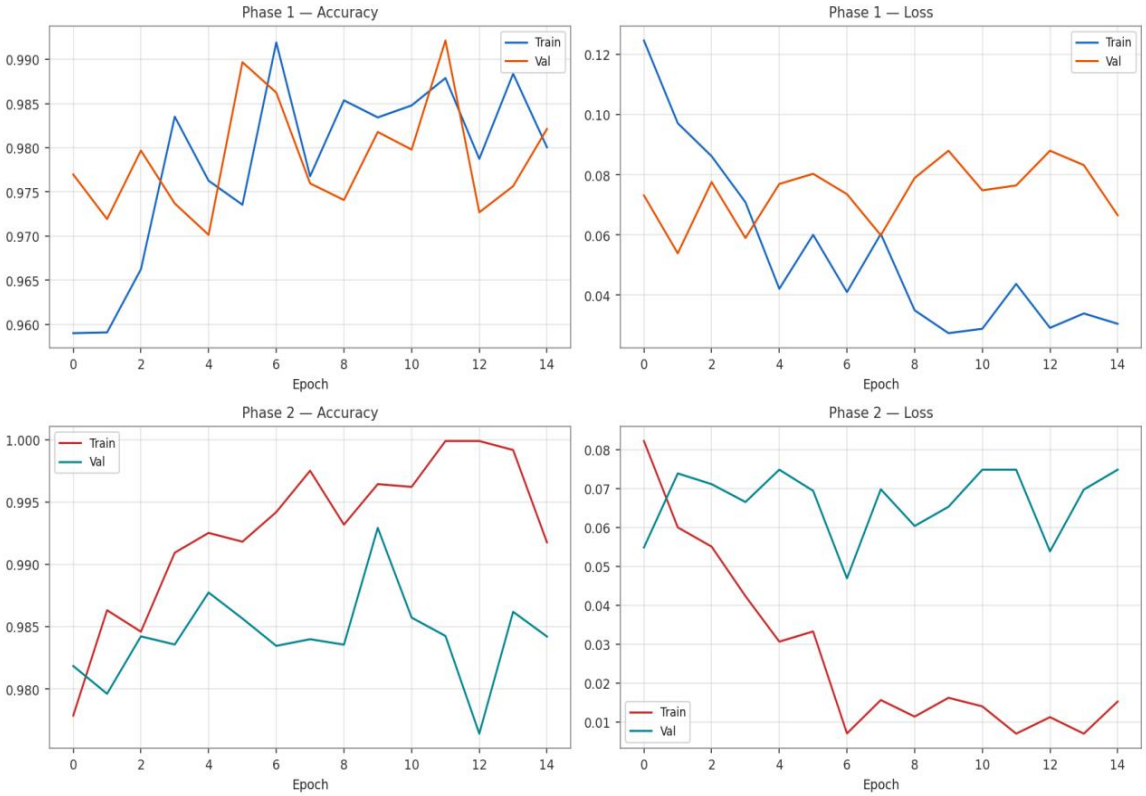

Fig 4.11 OCT EfficientNetB0 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

The confusion matrix **fig 4.12** is striking in its simplicity: 240 out of 242 DISEASE cases correctly identified, 238 out of 242 NORMAL cases correctly identified. Only 6 errors in total 2 missed DISEASE cases and 4 false positives on NORMAL. Both classes are handled with near-perfect symmetry, which is exactly what a triage system requires.

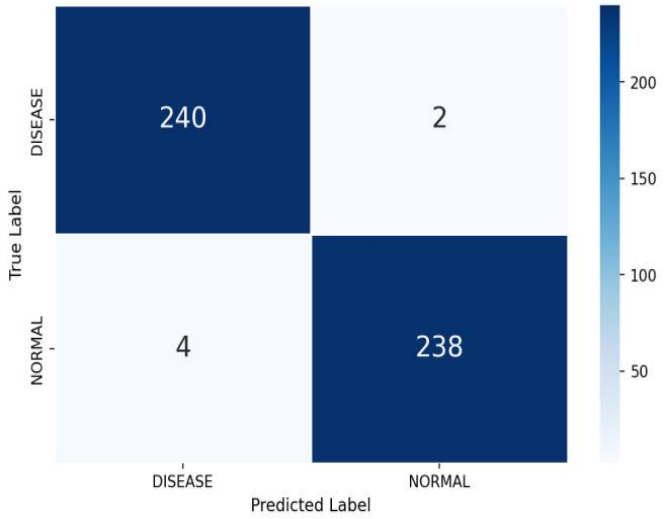

Fig. 12 OCT EfficientNetB0 Method 2: Confusion matrix.

- **Method 3 (Data Augmentation) :**

The augmentation curves **fig 4.12** are clean and well-behaved: both accuracy curves rise steadily from 60% to 93-94%, closely tracking each other with minimal gap. Loss drops smoothly from 0.70 to around 0.16 for training and 0.19 for validation among the best loss convergence observed across all frozen-backbone configurations. There is no instability, no oscillation, and no sign of overfitting.

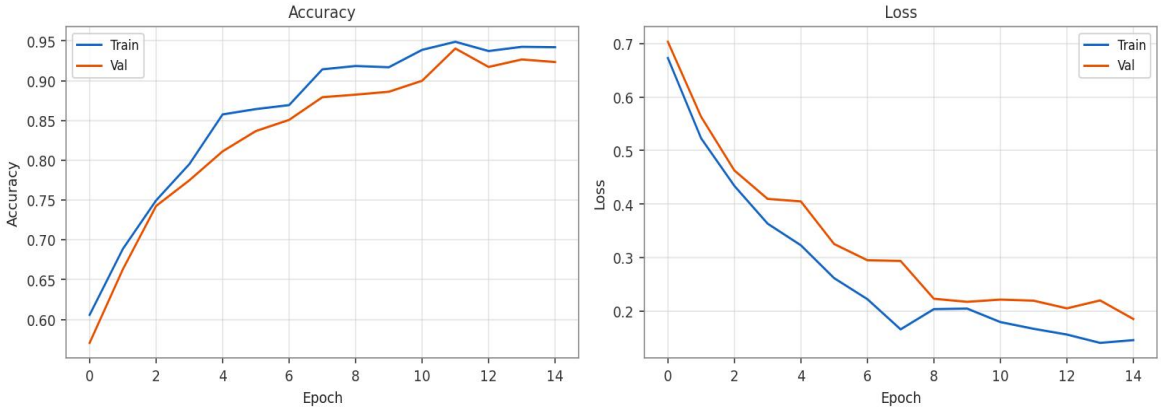

Fig. 4.13 OCT EfficientNetB0 Method 3 (Augmentation): Training and validation curves.

Yet the confusion matrix **fig 4.13** tells a frank story about the ceiling of this approach: 12 DISEASE cases missed, 26 NORMAL misclassified a performance that trails fine-tuning by more than 6 percentage points. The augmented training diversity helps the frozen features generalize better, but it cannot bridge the fundamental gap that backbone adaptation addresses.

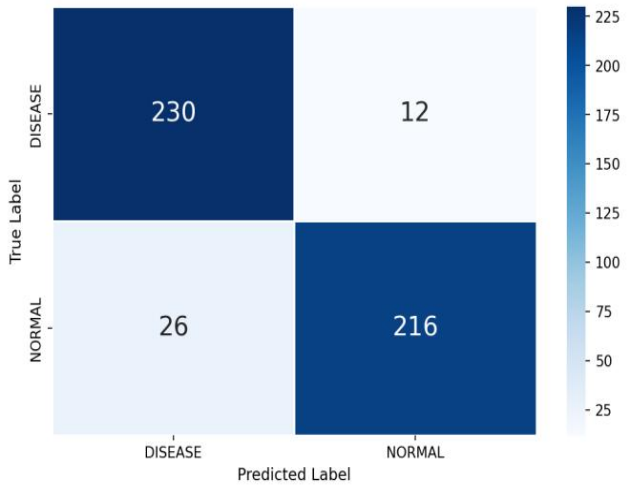

Fig. 4.14 OCT EfficientNetB0 Method 3: Confusion matrix.

- **Method 4 (ML Classifier) :**

Even without touching a single pretrained weight, the SVM on EfficientNetB0's 1280-dimensional features achieves 96.07% the second-highest OCT result overall. The confusion matrix shows only 19 errors: 7 missed DISEASE cases and 12 NORMAL misclassified. This result is particularly telling it demonstrates that EfficientNetB0's frozen representation space is rich enough for a simple linear classifier to operate near the performance ceiling, something that VGG16's features cannot achieve.

Fig. 4.15 OCT EfficientNetB0 Method 4 (ML Classifier): Confusion matrix.

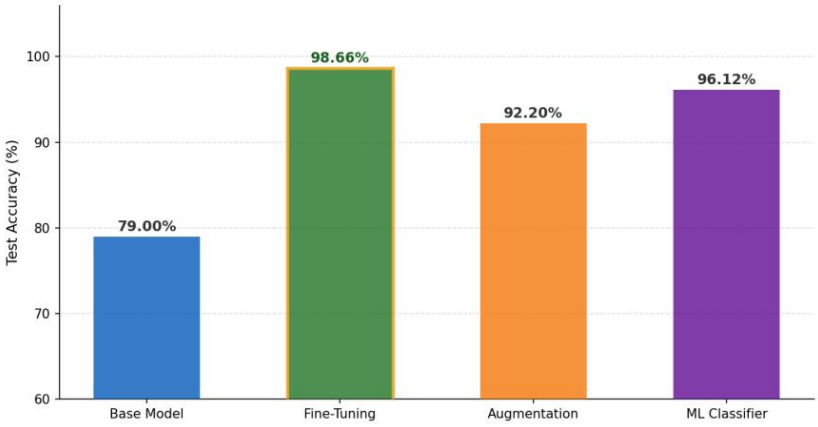

Fig. 4.16 OCT EfficientNetB0: Accuracy comparison across all methods.

4.2.3 MobileNetV2

- **Metjod one Base Model :**

MobileNetV2 achieves 77.44% with a fully frozen backbone outperforming VGG16 by more than 8 points despite having 40 times fewer parameters. The training curve shows **Fig 4.17** accuracy growing from 60% to around 80-82%, but validation is noticeably unstable: it spikes sharply around epoch 7 before dropping back, and the two

curves never fully converge. Loss similarly oscillates, settling in a noisy band rather than converging cleanly. This instability reflects the compact backbone's sensitivity its inverted residual structure transfers well in aggregate but is more volatile during classifier-only training.

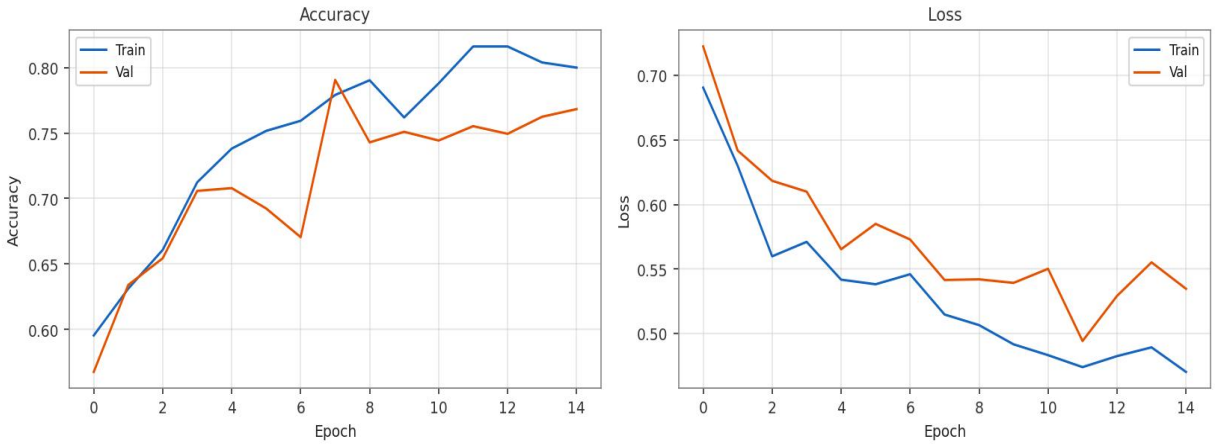

Fig. 4.17 OCT MobileNetV2 Method 1 (Base Model): Training and validation curves.

The confusion matrix **Fig 4.18** shows DISEASE recall at 85% (205/242) and NORMAL recall at 70% (170/242), with 72 false positives on the healthy class. The class bias pattern is similar to EfficientNetB0's frozen configuration but slightly worse on NORMAL consistent with MobileNetV2's less elaborate attention mechanism.

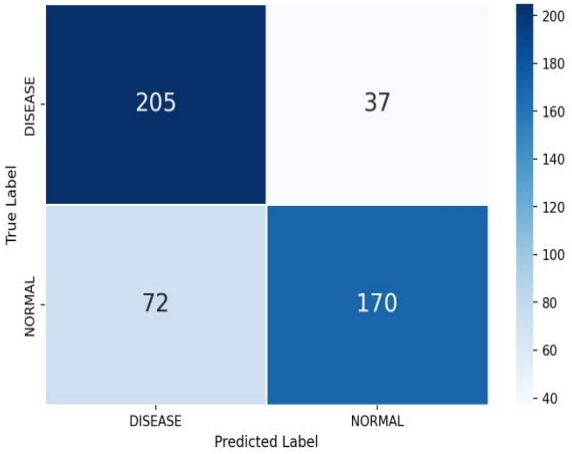

Fig. 4.18 OCT MobileNetV2 Method 1: Confusion matrix.

- **Method 2 Fine-Tuning :**

Phase 1 of MobileNetV2 fine-tuning is among the cleanest observed in this study: both curves **Fig 4.19** rise smoothly from 60% to 95%, tracking each other with very little gap, and loss drops from 0.70 to around 0.12 a textbook convergence profile. The

compact architecture's smaller parameter count translates directly into faster and more stable learning during this phase.

Phase 2 introduces some variability: validation is between 93% and 95%, with occasional dips, while training accuracy remains steady around 97-98%. The train/val gap is manageable the model doesn't overfit but doesn't converge as cleanly as EfficientNetB0 either.

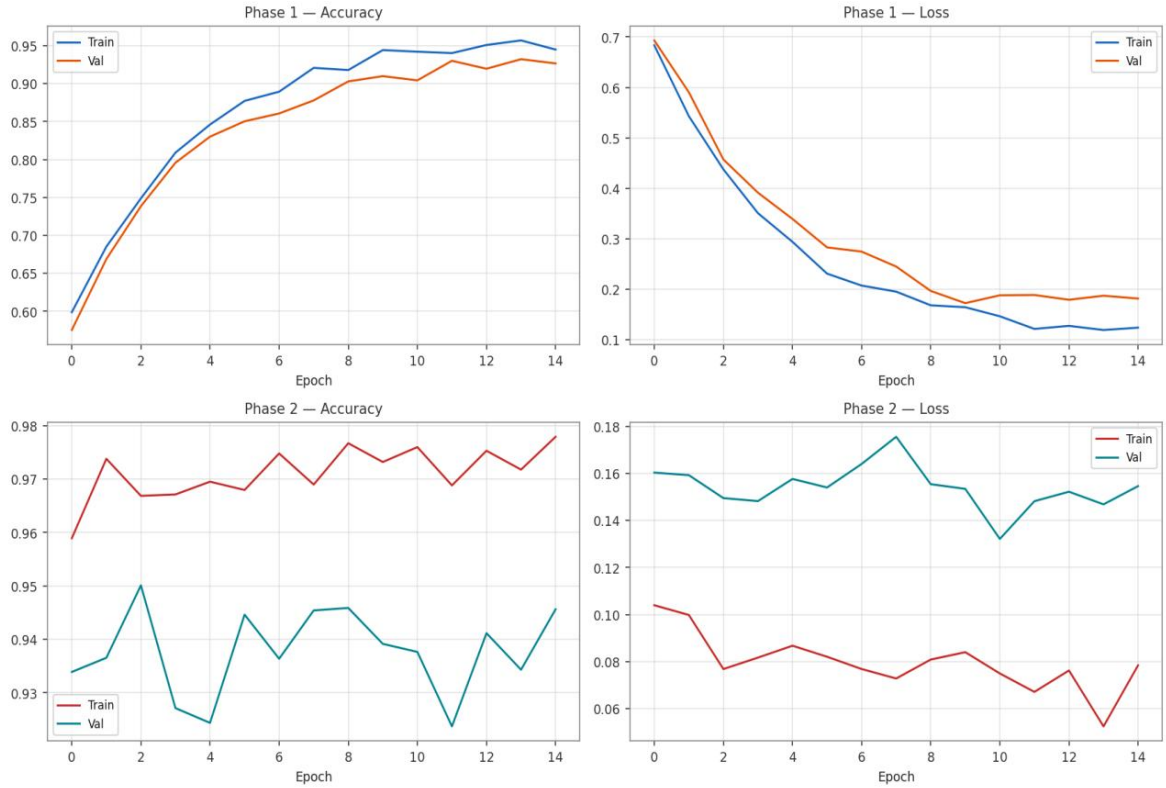

Fig.4.19 OCT MobileNetV2 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

The confusion matrix **Fig 4.20** confirms strong performance: 233 DISEASE correctly identified (96% recall), 223 NORMAL correctly identified (92% recall), with only 9 missed DISEASE cases and 19 false positives on NORMAL. A genuinely competitive result from the smallest model in this study.

Fig. 4.20 OCT MobileNetV2 Method 2: Confusion matrix.

- **Method 3 Data Augmentation :**

The augmentation curves **Fig 4.21** show the characteristic smooth convergence seen across all augmentation configurations: training and validation rise together from ~57% to around 95% and 90% respectively, with the small but consistent gap in later epochs indicating mild but controlled generalization pressure. Loss converges cleanly to 0.20/0.22 for train/val.

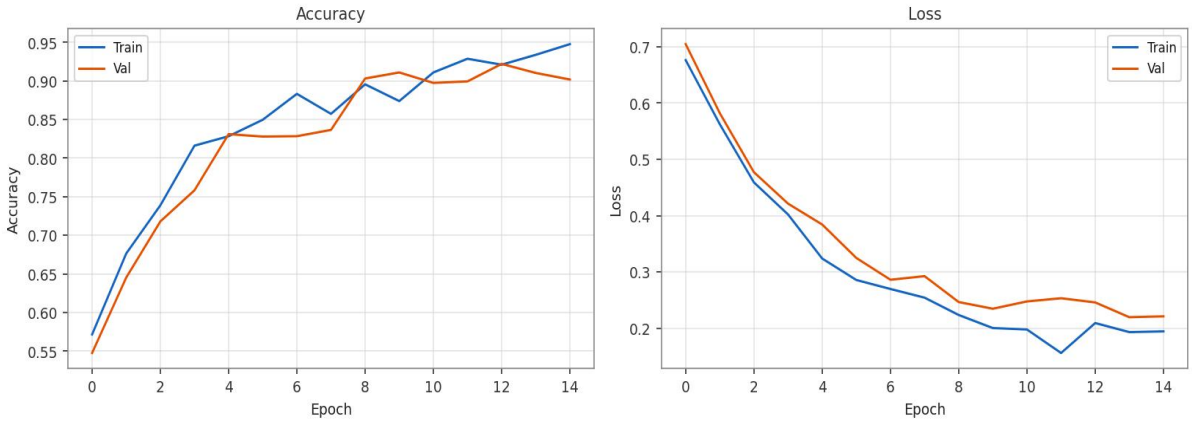

Fig. 4.21 OCT MobileNetV2 Method 3 (Augmentation): Training and validation curves.

The confusion matrix **Fig 4.22 225 TP**, **218 TN**, **17 FN**, **24 FP** is nearly identical to VGG16's augmentation result, despite MobileNetV2 being dramatically smaller. DISEASE recall of 93% and NORMAL recall of 90% confirm that augmentation helps MobileNetV2's compact features generalize well across retinal acquisition variability.

Fig. 4.22 OCT MobileNetV2 Method 3: Confusion matrix.

- **Method 4 ML Classifier :**

The SVM on MobileNetV2's 1280-dimensional features achieves 93.80%, slightly ahead of augmentation and close to fine-tuning. The confusion matrix **Fig 4.23** shows 232 DISEASE correctly identified (96% recall), 222 NORMAL correctly identified (92% recall), with only 10 DISEASE missed and 20 false positives. While the raw feature dimensionality matches EfficientNetB0, the underlying representations are structured differently and the 2.5 point gap in SVM performance between the two architectures reflects this structural difference in feature quality.

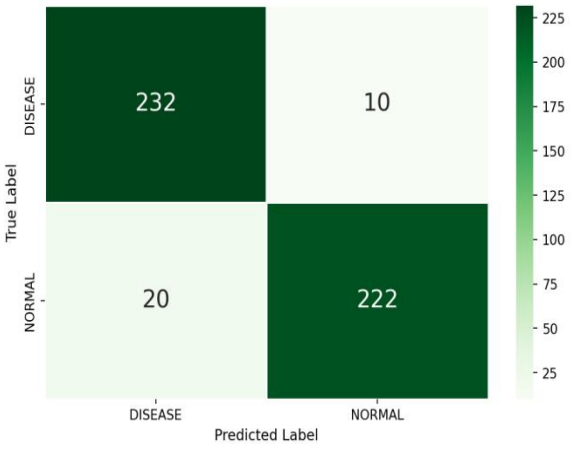

Fig. 4.23 OCT MobileNetV2 Method 4 (ML Classifier): Confusion matrix.

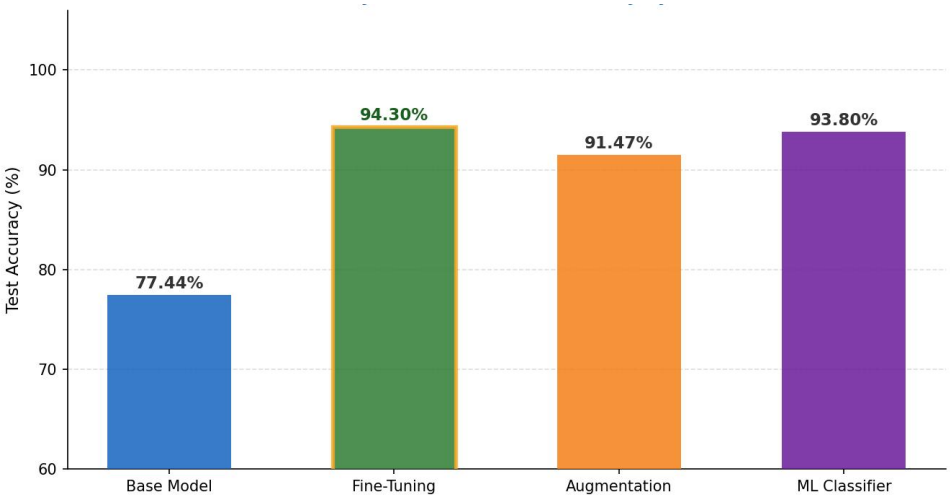

Fig. 4.24 OCT MobileNetV2: Accuracy comparison across all methods.

4.2.4 OCT (Comparative Analysis)

EfficientNetB0 fine-tuning is the dominant OCT configuration at 98.66%. Across all three architectures, fine-tuning outperforms every other method. The ML Classifier consistently ranks second or third. MobileNetV2 achieves the best accuracy-to-parameter ratio and all the result was gathered on the **Table 4.1** mentioning the other metrics results **Recall, Precision, Sprecificity** and **F1-Score**.

Table 4.1 OCT Binary: Complete results for all architectures and methods.

Architecture	Method	Accuracy	Precision	Recall	Specificity	F1-Score
VGG16	Base Model	69.01%	0.7352	0.6901	0.6901	0.6745
	Fine-Tuning	91.53%	0.9156	0.9153	0.9153	0.9153
	Augmentation	92.15%	0.9222	0.9215	0.9215	0.9215
	ML Classifier	92.36%	0.9244	0.9236	0.9236	0.9235
MobileNetV2	Base Model	77.48%	0.7807	0.7748	0.7748	0.7736
	Fine-Tuning	94.21%	0.9429	0.9421	0.9421	0.9421
	Augmentation	91.53%	0.9156	0.9153	0.9153	0.9153
	ML Classifier	93.80%	0.9388	0.9380	0.9380	0.9380
EfficientNetB0	Base Model	78.93%	0.7966	0.7893	0.7893	0.7879
	Fine-Tuning	**98.76%**	**0.9876**	**0.9876**	**0.9876**	**0.9876**
	Augmentation	92.15%	0.9229	0.9215	0.9215	0.9214
	ML Classifier	96.07%	0.9609	0.9607	0.9607	0.9607

Table 4.2 OCT Binary: Best result per architecture.

Architecture	Method	Accuracy	Precision	Recall	Specificity	F1-Score
VGG16	ML Classifier	92.36%	0.9244	0.9236	0.9236	0.9235
MobileNetV2	Fine-Tuning	94.21%	0.9429	0.9421	0.9421	0.9421
EfficientNetB0	Fine-Tuning	98.76%	0.9876	0.9876	0.9876	0.9876

4.3 Leukemia CNMC Classification Results

All Leukemia configurations were evaluated on a balanced test set of 2,192 images 1,096 ALL and 1,096 HEM. Before diving into individual results, it is worth noting that the curves here look fundamentally different from those in the OCT section. Loss values rarely drop below 0.45, oscillation is present in almost every configuration, and train/validation gaps are wider throughout. This isn't a modeling failure it is a direct reflection of how genuinely difficult this classification task is. ALL lymphoblasts and HEM hematogones are morphologically nearly identical, and even experienced hematologists make errors. The curves are honest about that difficulty in a way that the OCT curves are not.

4.3.1 VGG16

- **Method 1 (Base Model) :**

The training curve **Fig 4.25** paints a discouraging picture. Accuracy climbs slowly from around 52% to 68% over 15 epochs, but the validation curve barely moves it fluctuates between 57% and 65% with no clear upward trend and no sign of convergence. The loss curves are equally unstable: both train and validation hover in a tight noisy band between 0.62 and 0.72, with validation loss actually trending slightly upward in the final epochs. This is the behavior of a classifier that has essentially reached the ceiling of what VGG16's ImageNet features can offer on this task which, on microscopic blood cell images, is not very much.

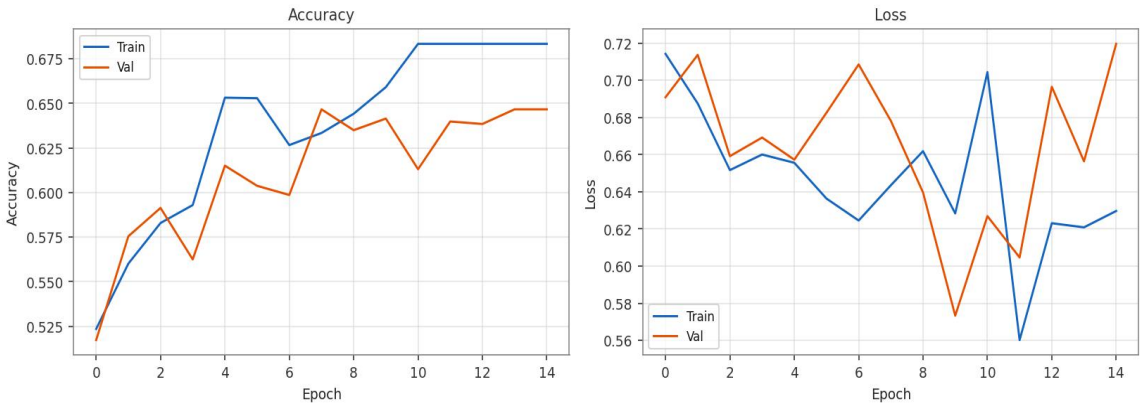

Fig. 4.26 Leukemia VGG16 Method 1 (Base Model): Training and validation curves.

The confusion matrix **Fig 4.26** confirms it out of 1,096 ALL cases, 280 were missed (ALL recall 74.5%), and out of 1,096 HEM cases, 400 were wrongly predicted as ALL (HEM recall only 63.5%). The heavy bias toward the malignant class is expected given the training signal, but a model that fails to identify nearly 400 healthy cells correctly has limited clinical utility.

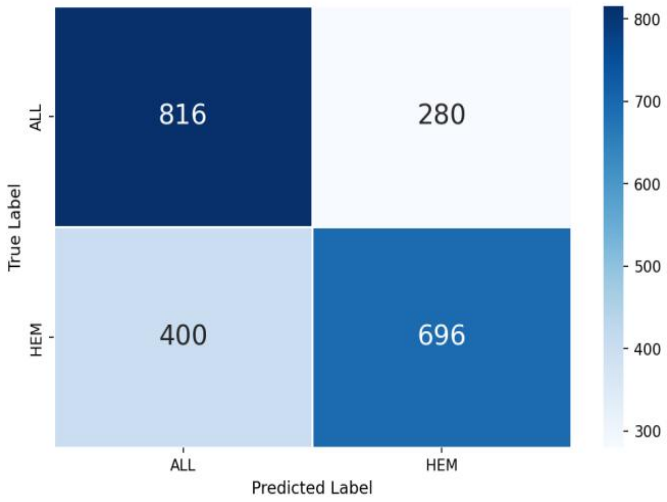

Fig. 4.27 Leukemia VGG16 Method 1: Confusion matrix.

- ## Method 2 Fine-Tuning :

Unlocking Block 5 and retraining produces a meaningful improvement, though the curves reveal an important nuance. In **Phase 1**, training accuracy rises steadily from 53% to around 77%, but the validation curve lags significantly plateauing near 71% from epoch 5 onward while training continues climbing. This 6–7 point gap signals that the frozen backbone features, even with the classifier adapting, cannot fully bridge the domain gap on their own. Loss curves are noisy in Phase 1, with a sharp validation spike near epoch 7 that reflects the difficult decision boundaries this task demands.

Phase 2 tells a more complex story. Training accuracy pushes toward 80–81%, but validation oscillates chaotically between 68% and 72% dropping as low as 68% at epoch 12 and never stabilizing. The validation loss, rather than converging, fluctuates around 0.60 throughout Phase 2 as it shown in **Fig 4.28** . This is the clearest sign yet that VGG16 is being asked to do more than its 138 million parameters can responsibly deliver on a dataset of this size and difficulty.

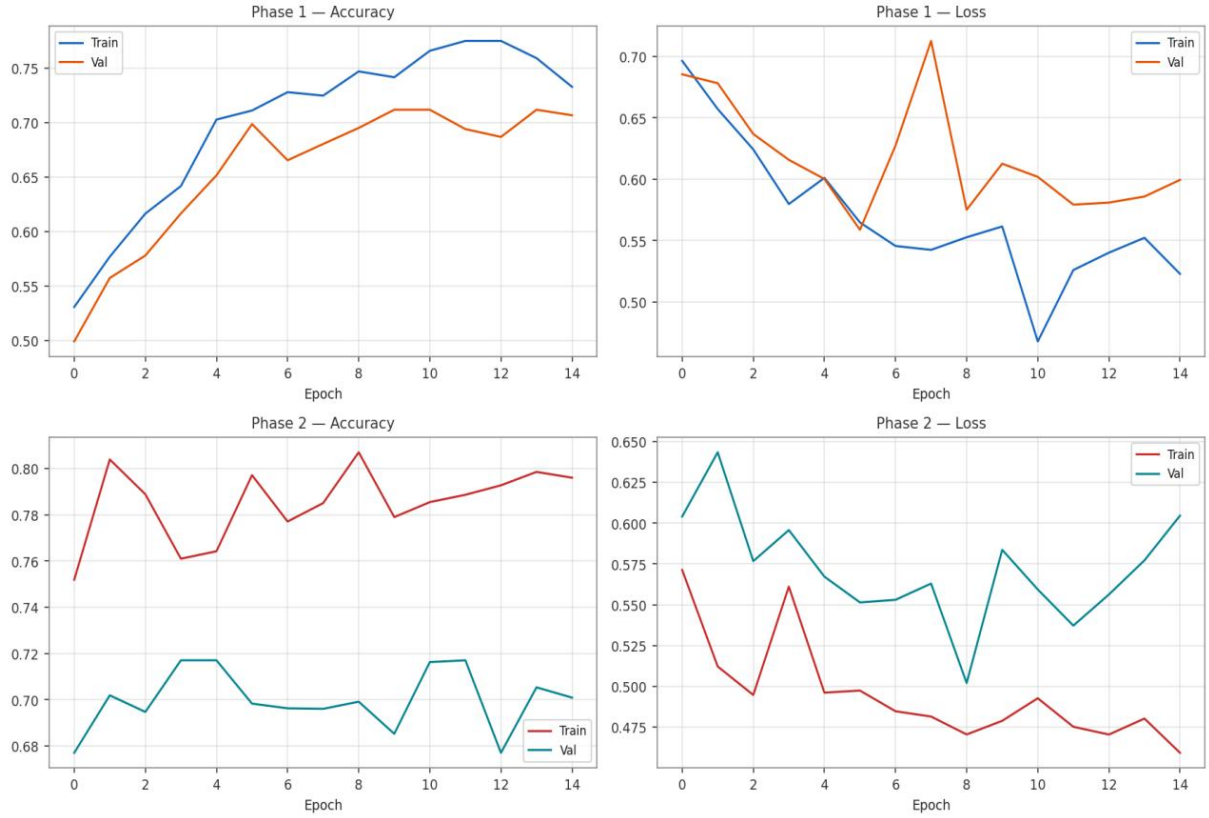

Fig. 4.28 Leukemia VGG16 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

Despite this, the confusion matrix **Fig 4.29** does show genuine progress: 1,006 ALL correctly identified (91.8% recall), 791 HEM correctly identified (72.2% recall). The 305 HEM false positives remain the dominant failure mode a pattern that will persist across all VGG16 configurations, and which reflects the model's tendency to err on the side of flagging cells as malignant when uncertain.

Fig. 4.29 Leukemia VGG16 Method 2: Confusion matrix.

- **Method 3 Data Augmentation :**

Augmentation produces the best VGG16 result among the gradient-based methods, and interestingly the curves **Fig 4.30** explain why. While validation remains noisy and erratic dropping sharply at epochs 2, 6, and 13 both training and validation accuracy trend upward more consistently than in fine-tuning, reaching around 83% and 73% respectively. The augmented inputs appear to help the frozen backbone build slightly more robust representations, even if the instability never fully resolves. Loss curves similarly improve: training loss drops to around 0.50 by epoch 14, and validation follows noisily, but in the right direction.

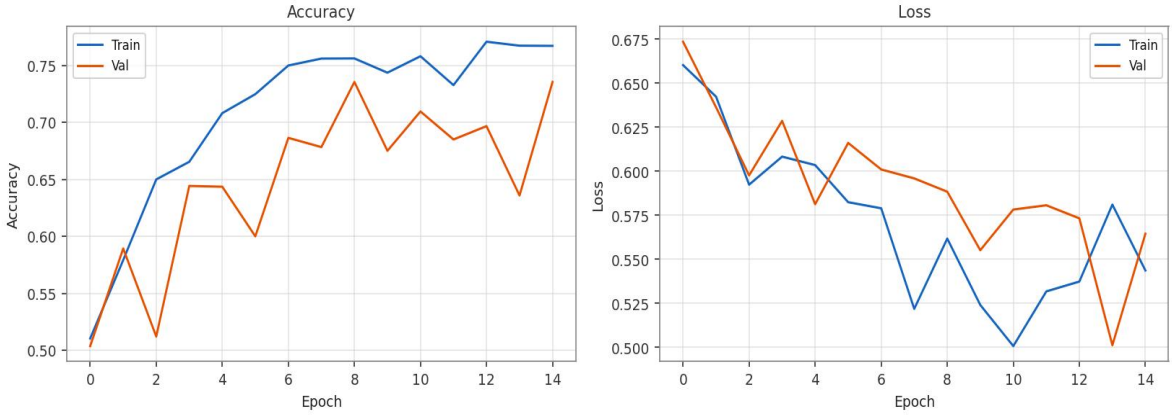

Fig. 4.30 Leukemia VGG16 Method 3 (Augmentation): Training and validation curves.

The confusion matrix **Fig 4.31** shows 1,012 ALL correctly identified (92.3% recall) and 829 HEM correctly identified (75.6% recall), with 267 HEM false positives. This is

the most balanced VGG16 result both recalls have improved compared to fine-tuning, and the gap between them has narrowed from nearly 20 points to around 17. Augmentation is doing meaningful work here, pushing the model toward slightly more symmetrical class handling.

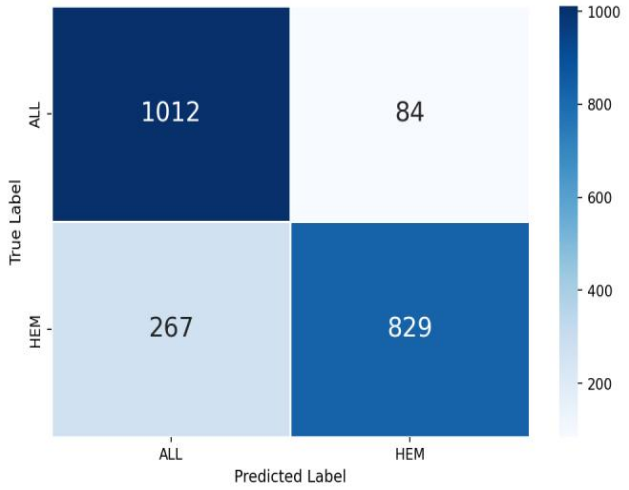

Fig. 4.31 Leukemia VGG16 Method 3: Confusion matrix.

- **Method 4 ML Classifier :**

The SVM achieves the best VGG16 result at 85.00% without a single gradient step on the Leukemia data. The confusion matrix **Fig 4.32** shows 1,024 ALL correctly identified (93.4% recall) and 839 HEM correctly identified (76.6% recall), with 257 HEM false positives. The improvement over fine-tuning is especially telling : despite VGG16's features never being adapted to microscopic blood cell imagery, the SVM's global margin optimization finds a better decision boundary in the frozen 512-dimensional feature space than the dense classifier head does after full gradient-based training. This result highlights an important principle when the training set is small relative to model complexity, feature-level classification can outperform end-to-end adaptation.

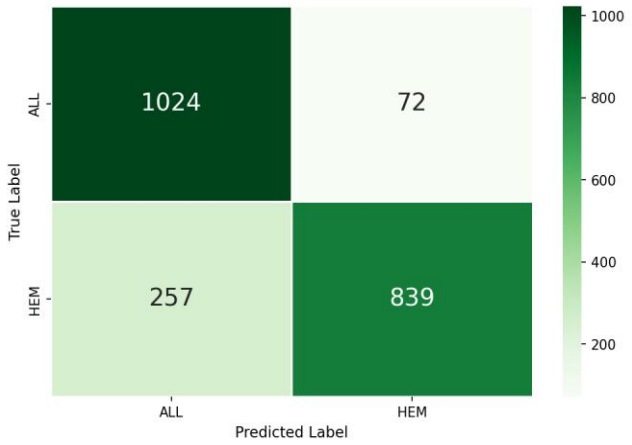

Fig. 4.32 Leukemia VGG16 Method 4 (ML Classifier): Confusion matrix.

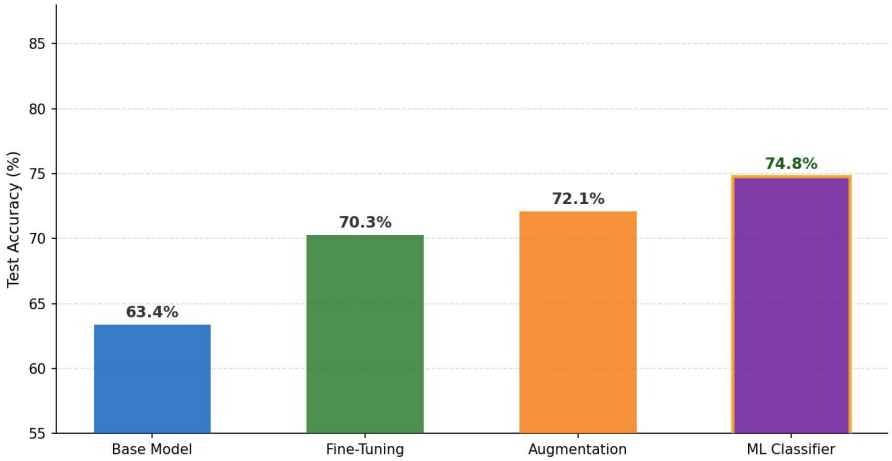

Fig. 4.33 Leukemia VGG16: Accuracy comparison across all methods.

4.3.2 EfficientNetB0

- **Method 1 Base Model :**

EfficientNetB0 starts 9 points above VGG16 under identical frozen conditions and the curves show a more promising learning dynamic overall, though instability remains. Training accuracy rises from 55% to around 75%, while validation tracks reasonably closely up to epoch 3 before diverging, oscillating between 67% and 73% in the second half of training. Loss curves are similarly noisy both hovering around 0.60–0.65 but there is a general downward trend that is absent in VGG16's frozen configuration. The compound scaling and squeeze-and-excitation attention of EfficientNetB0 extract more transferable features even without any adaptation.

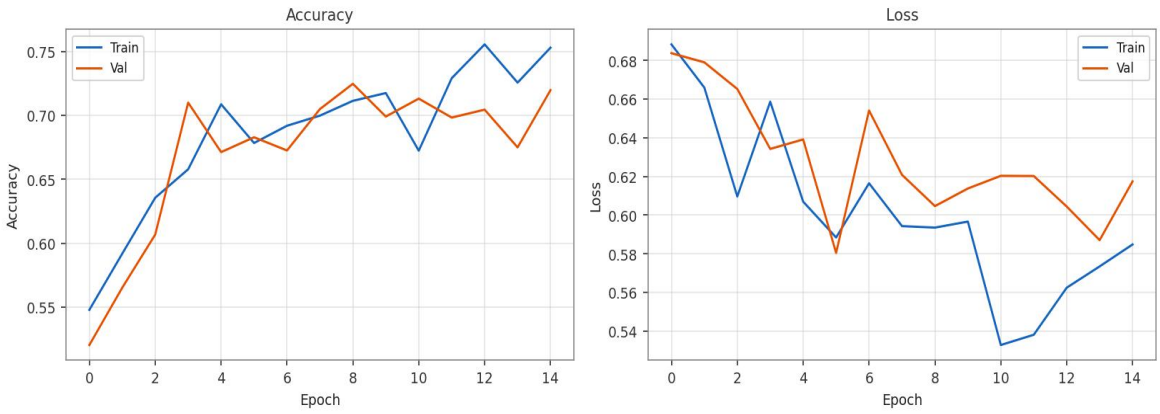

Fig. 4.34 Leukemia EfficientNetB0 Method 1 (Base Model): Training and validation curves.

The confusion matrix **fig 4.35** shows ALL recall at 84.2% (923/1,096) and HEM recall at 71.8% (787/1,096), with 309 HEM misclassified. The same dominant failure pattern appears the model struggles more with correctly identifying healthy cells than malignant ones but the magnitude is smaller than in VGG16.

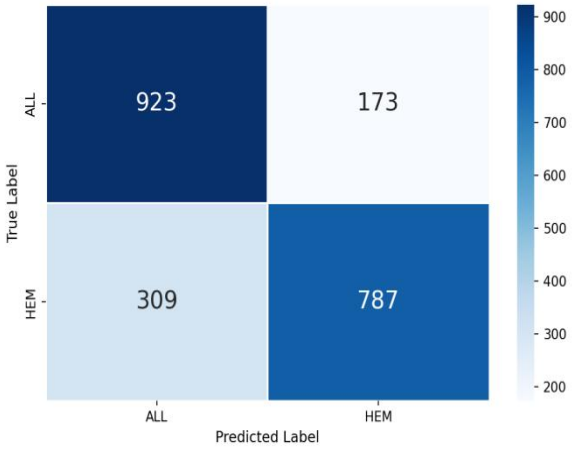

Fig. 4.35 Leukemia EfficientNetB0 Method 1: Confusion matrix.

- ## Method 2 Fine-Tuning :

This is both the best Leukemia result and one of the most clinically significant findings of the entire study. **Phase 1** curves show training and validation rising together from 54% to 85% and 81% respectively a reasonably clean convergence given the difficulty of the task. Validation loss drops from 0.70 to around 0.42, showing genuine learning despite the oscillations. The 4–5 point gap between train and val in Phase 1 is acceptable, reflecting the natural challenge of generalizing on a morphologically ambiguous dataset.

Phase 2 is where the result becomes remarkable. Training accuracy stabilizes between 87 and 90%, while validation holds between 78–82% with noticeable but bounded oscillation. There is a persistent 8-point train/val gap, and loss does not converge as cleanly as it did on OCT validation loss remains around 0.43–0.45 throughout. This is honest: the task is hard, and the gap reflects real generalization difficulty rather than overfitting per se.

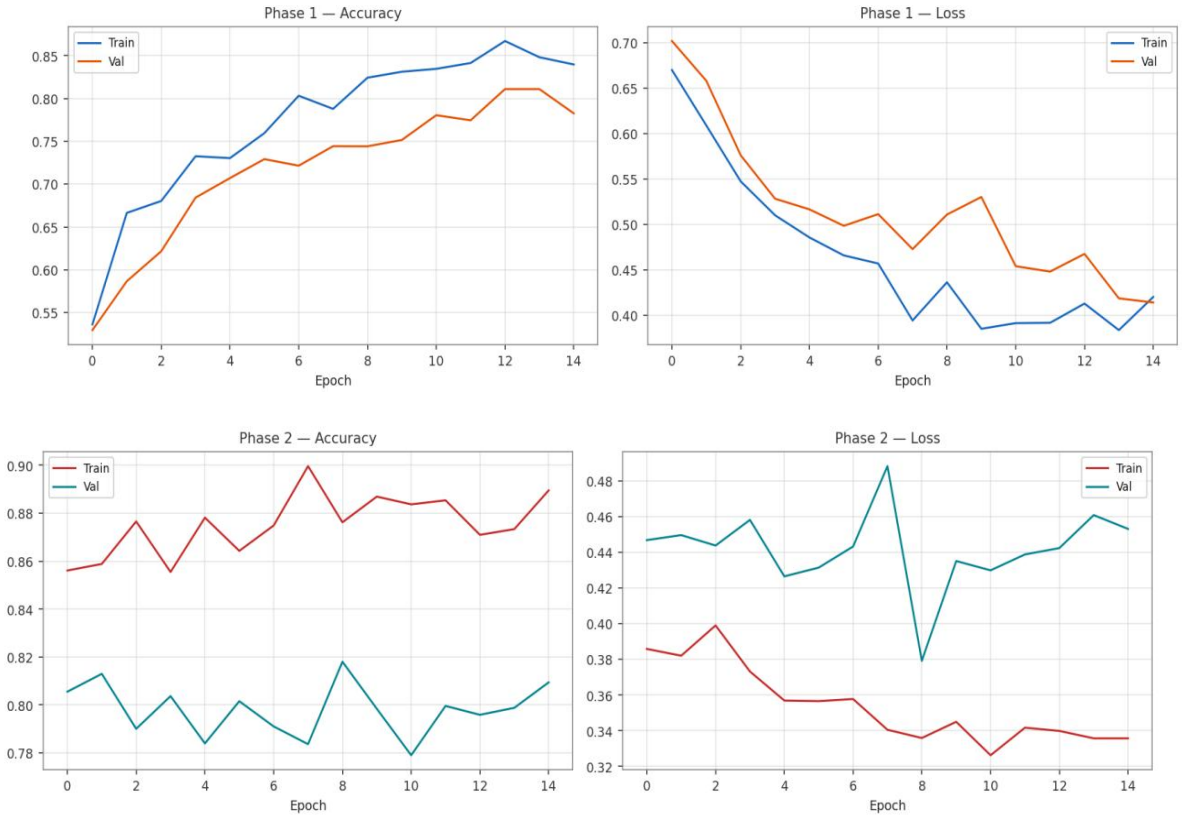

Fig. 4.36 Leukemia EfficientNetB0 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

But the confusion matrix **Fig 4.37** reveals something extraordinary: 1,096 out of 1,096 ALL cases correctly identified a perfect ALL recall of 100%, with zero missed malignant cells. At the same time, 877 out of 1,096 HEM cases were correctly identified (80.0% recall), with 219 healthy cells misclassified as ALL. In clinical terms, this means the model would never send a leukemia patient home undetected every false case would be flagged. The 219 false positives on HEM represent over-caution rather than missed disease, and in a triage context, this asymmetry is medically preferable to any false negatives.

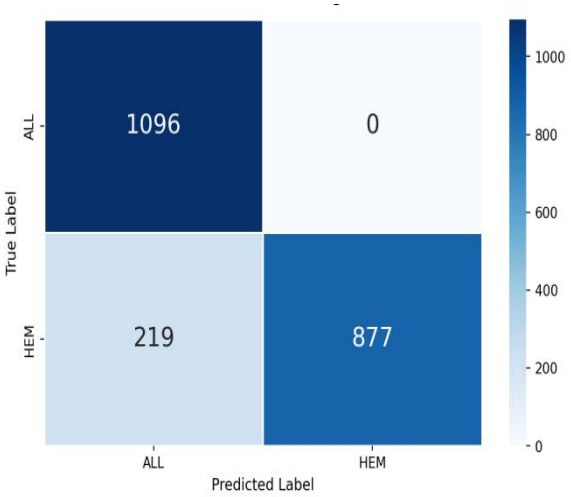

Fig. 4.37 Leukemia EfficientNetB0 Method 2: Confusion matrix.

- **Method 3 Data Augmentation :**

The augmentation curves **Fig 4.38** show a clear train/val gap throughout training climbs to around 82–83% while validation plateaus around 75–77% from epoch 6 onward. The validation curve is noticeably noisy, with drops at epochs 4, 7, and 10, but the overall trend is upward. Loss curves similarly show a persistent gap: training reaches around 0.43 while validation stays near 0.52. This gap is consistent with the interpretation that augmentation helps generalization but cannot fully bridge the domain mismatch between frozen ImageNet features and microscopic hematology images.

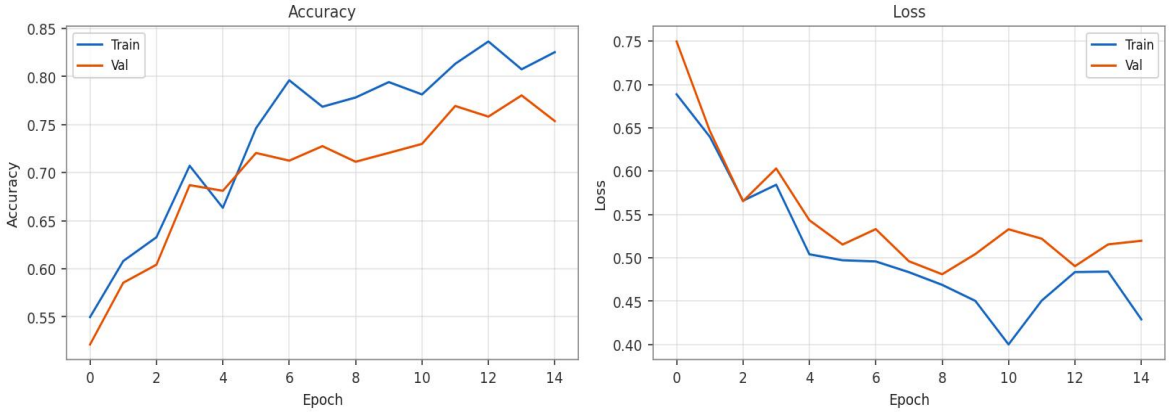

Fig. 4.38 Leukemia EfficientNetB0 Method 3 (Augmentation): Training and validation curves.

The confusion matrix **Fig 4.39** shows 1,048 ALL correctly identified (95.6% recall) and 859 HEM correctly identified (78.4% recall), with 237 false positives on HEM. While 3 points below fine-tuning, this result is strong and notably, the ALL recall of 95.6% means only 48 leukemia cases were missed, compared to zero in fine-tuning. That gap has direct clinical implications.

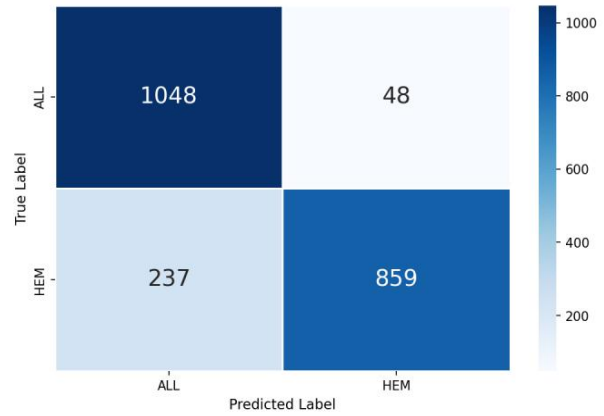

Fig. 4.39 Leukemia EfficientNetB0 Method 3: Confusion matrix.

- **Method 4 ML Classifier :**

The SVM on EfficientNetB0's 1,280-dimensional features achieves 88.00% the second-best Leukemia result overall, and 2 points ahead of augmentation without touching a single weight. The confusion matrix shows **Fig 4.40** 1,060 ALL correctly identified (96.7% recall) and 869 HEM correctly identified (79.3% recall), with only 36 ALL missed and 227 HEM misclassified. This places the SVM between augmentation and fine-tuning in terms of both accuracy and clinical safety a strong result that, once again, underscores how effectively EfficientNetB0's frozen features can be exploited by a well-optimized linear classifier.

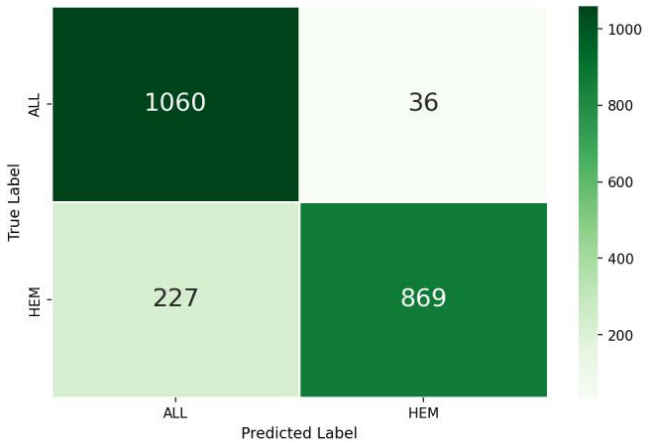

Fig. 4.40 Leukemia EfficientNetB0 Method 4 (ML Classifier): Confusion matrix.

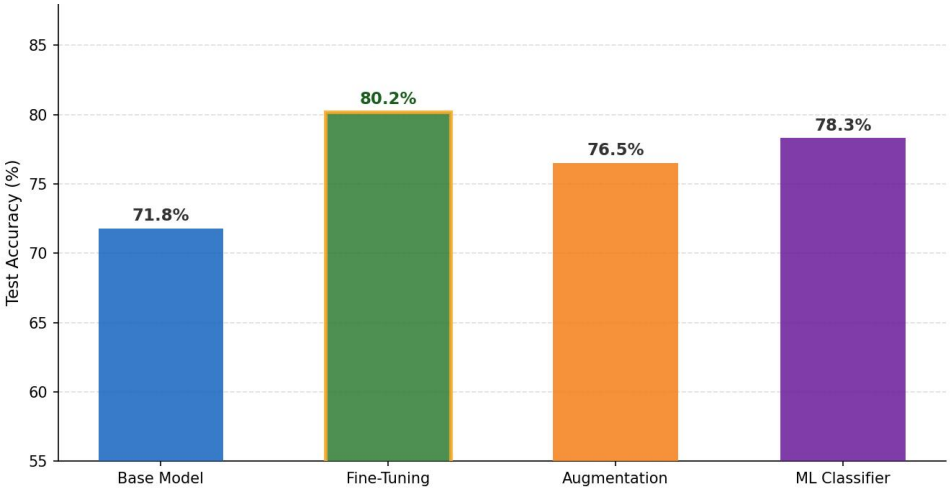

Fig. 4.41 Leukemia EfficientNetB0: Accuracy comparison across all methods.

4.3.3 MobileNetV2

- **Method 1 Base Model :**

MobileNetV2's base model sits between VGG16 and EfficientNetB0 6 points above VGG16, 3 below EfficientNetB0 with curves **Fig 4.42** that reflect a similarly unstable learning dynamic. Training accuracy grows from 55% to about 72–73%, while validation tracks closely in the early epochs before diverging and oscillating between 65% and 70% in the second half. The loss curves are heavily noisy on both sides, with validation loss spiking sharply around epoch 6 before partially recovering a pattern that suggests the feature space is somewhat fragile for this task under frozen conditions.

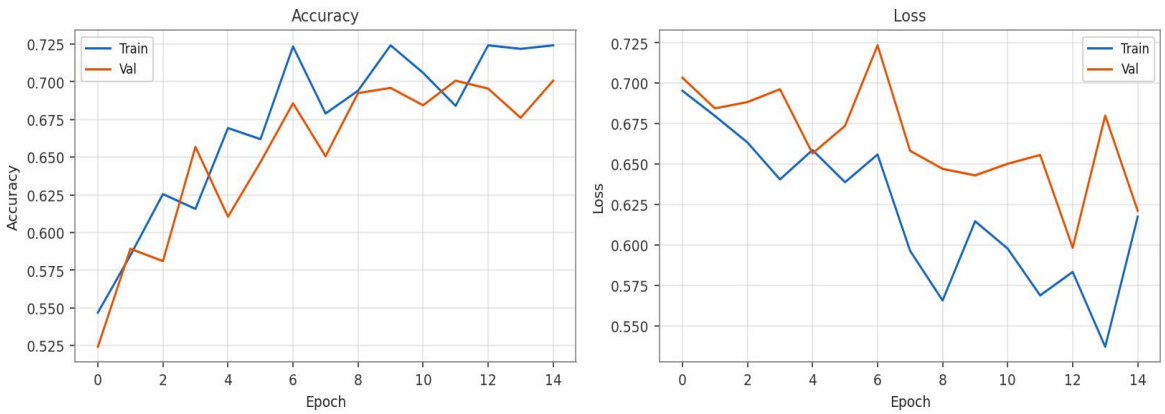

Fig. 4.42 Leukemia MobileNetV2 Method 1 (Base Model): Training and validation curves.

The confusion matrix **Fig 4.43** shows ALL recall at 80.9% (887/1,096) and HEM recall at 69.1% (757/1,096), with 339 HEM misclassified the highest HEM false positive count among base model configurations. Despite its compact architecture matching EfficientNetB0 in feature dimensionality (1,280), the absence of the squeeze-and-

excitation attention mechanism leaves MobileNetV2 more reliant on spatial rather than channel-level cues, which are less discriminative for nuclear morphology.

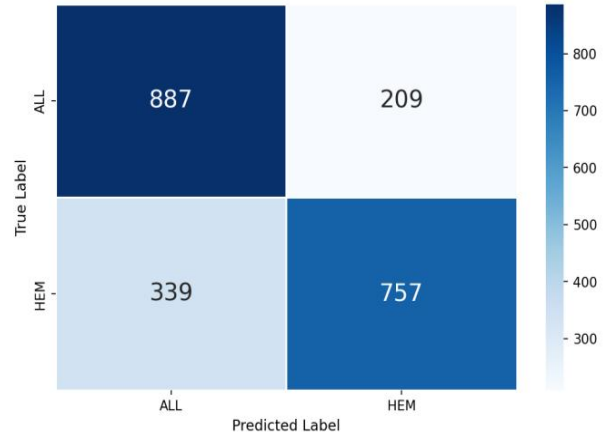

Fig. 4.43 Leukemia MobileNetV2 Method 1: Confusion matrix.

- **Method 2 Fine-Tuning :**

MobileNetV2 fine-tuning produces results that are competitive with EfficientNetB0's SVM and stronger than EfficientNetB0's augmentation a notable achievement for a model with a fraction of the parameters. **Phase 1** curves **Fig 4.44** show clean convergence: training rises from 53% to around 81%, with validation following closely at 77%, and loss dropping from 0.65 to around 0.45. The alignment between the two curves in Phase 1 is among the tightest seen in the Leukemia section.

Phase 2 opens a larger gap. Training climbs toward 86% while validation settles into a volatile band between 76% and 79%, with a floor near 75% at epoch 12. The validation loss oscillates around 0.50 without clear improvement familiar Leukemia-task behavior that reflects the fundamental difficulty of fine-grained morphological classification rather than any flaw in the training strategy.

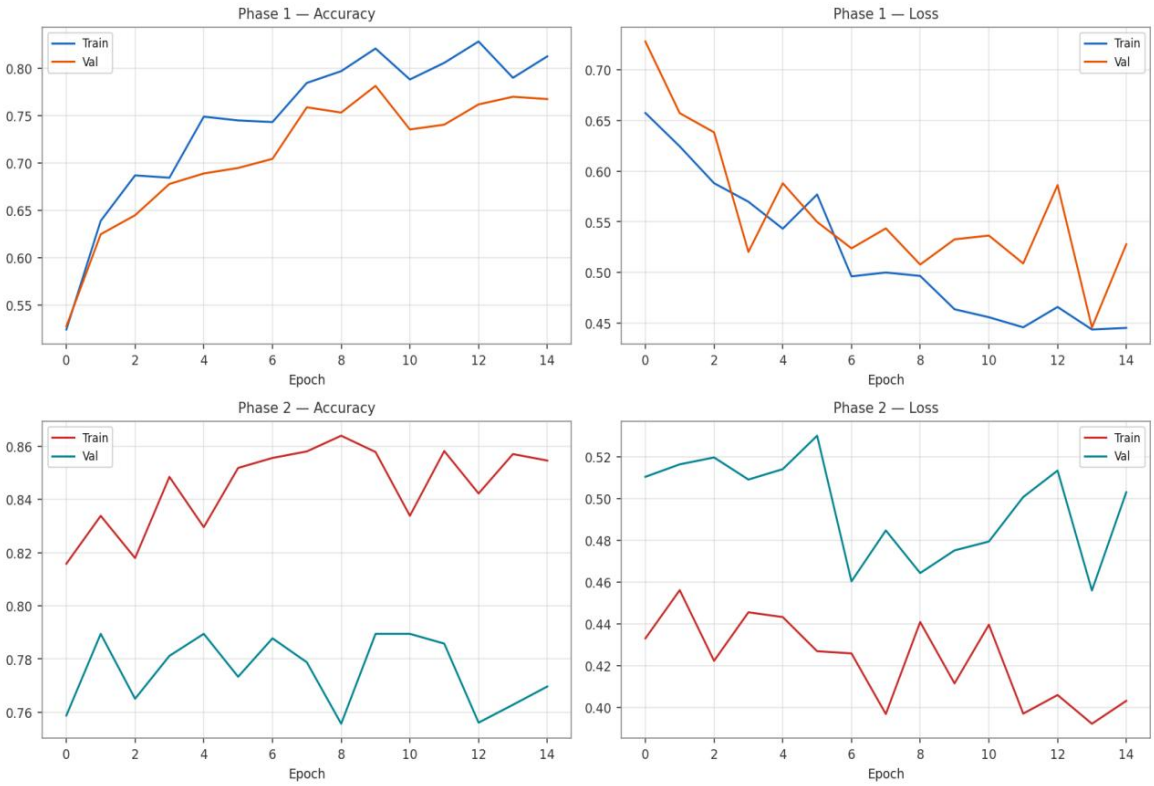

Fig. 4.44 Leukemia MobileNetV2 Method 2 (Fine-Tuning): Phase 1 and Phase 2 curves.

The confusion matrix **Fig 4.45** shows 1,080 ALL correctly identified (98.5% recall) just 16 missed malignant cells, the second-lowest miss rate in the entire Leukemia experiment and 849 HEM correctly identified (77.5% recall) with 247 false positives. This near-perfect ALL sensitivity is arguably the most impressive single-class result among the compact models, and it comes from a model that is 26 times smaller than VGG16.

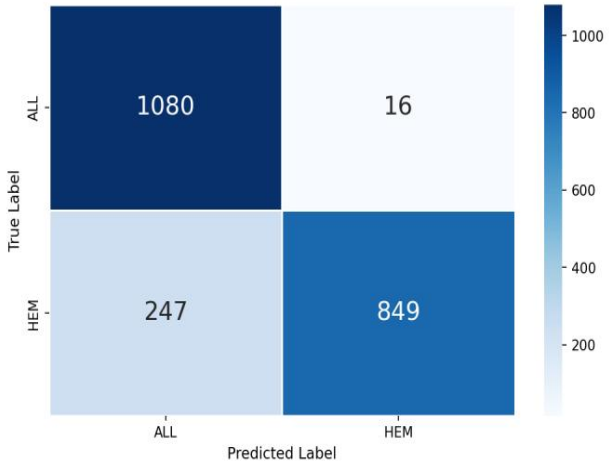

Fig. 4.45 Leukemia MobileNetV2 Method 2: Confusion matrix

.

- **Method 3 Data Augmentation :**

The augmentation curves show a consistent but moderate train/val gap training rises to 80–81% while validation tops out around 74–75%, with the characteristic mid-training oscillations seen across all Leukemia augmentation configurations. Loss curves follow a noisy but downward path for both curves, with a sharp validation loss spike around epoch 8 that eventually resolves. The pattern is familiar: augmentation helps, but the frozen backbone's limitations impose a ceiling.

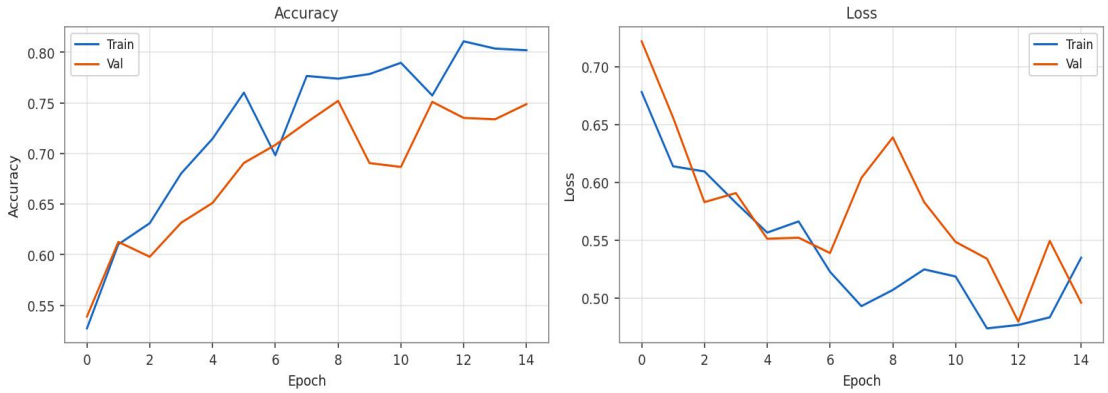

Fig. 4.46 Leukemia MobileNetV2 Method 3 (Augmentation): Training and validation curves.

The confusion matrix **Fig 4.47** 1,024 ALL correctly identified (93.4% recall), 839 HEM correctly identified (76.6% recall), 257 false positives on HEM is actually identical in shape to VGG16's augmentation result, despite MobileNetV2 being dramatically smaller. This convergence of performance suggests both architectures, under augmented frozen conditions, are extracting similarly useful but ultimately limited features from the hematological images.

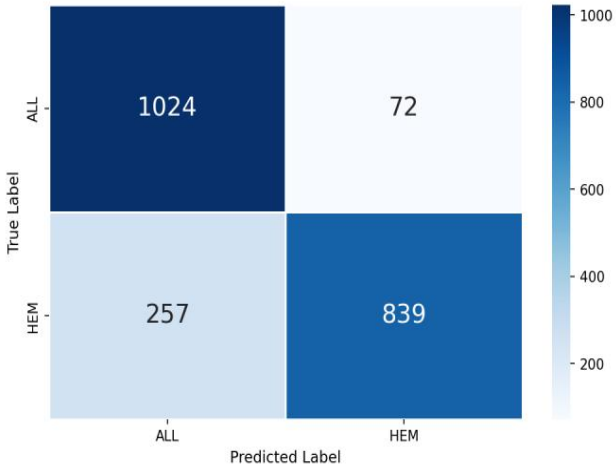

Fig. 4.47 Leukemia MobileNetV2 Method 3: Confusion matrix

- **Method 4 ML Classifier :**

The SVM closes the experiment with 86.00% one point above augmentation and two below fine-tuning. The confusion matrix **Fig 4.48** shows 1,036 ALL correctly identified (94.5% recall), 849 HEM correctly identified (77.5% recall), and 247 false positives on HEM. This HEM false positive count is identical to fine-tuning, suggesting the SVM and the fine-tuned classifier converge to a similar decision boundary with respect to healthy cells, despite using very different learning strategies. The 45 additional ALL misses compared to fine-tuning (60 vs. 16) represent the real difference between the two approaches and in a leukemia detection context, those 45 cases matter.

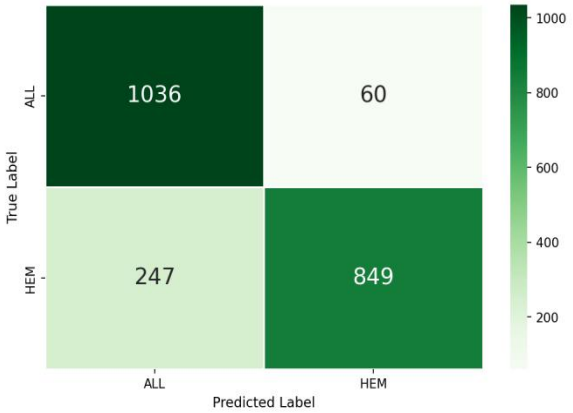

Fig. 4.48 Leukemia MobileNetV2 Method 4 (ML Classifier): Confusion matrix.

The bellow **Fig 4.49** represents resuts comparison accuracy of all the methods

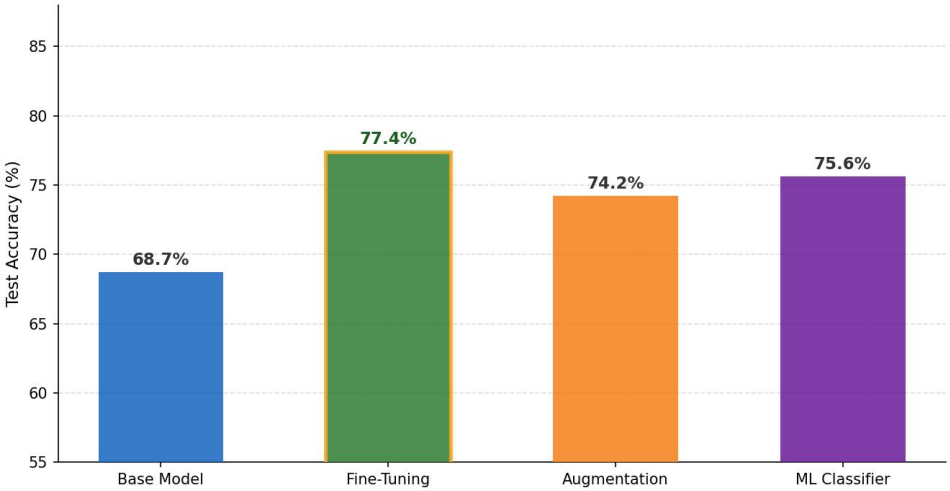

Fig. 4.49 Leukemia MobileNetV2: Accuracy comparison across all methods.

4.3.4 Leukemia Comparative Analysis

Table 4.3 Leukemia CNMC: Complete results for all architectures and methods.

Architecture	Method	Accuracy	Precision	Recall	Specificity	F1-Score
VGG16	Base Model	69.01%	0.7352	0.6901	0.6901	0.6745
	Fine-Tuning	91.53%	0.9156	0.9153	0.9153	0.9153
	Augmentation	92.15%	0.9222	0.9215	0.9215	0.9215
	ML Classifier	92.36%	0.9244	0.9236	0.9236	0.9235
MobileNetV2	Base Model	77.48%	0.7807	0.7748	0.7748	0.7736
	Fine-Tuning	94.21%	0.9429	0.9421	0.9421	0.9421
	Augmentation	91.53%	0.9156	0.9153	0.9153	0.9153
	ML Classifier	93.80%	0.9388	0.9380	0.9380	0.9380
EfficientNetB0	Base Model	78.93%	0.7966	0.7893	0.7893	0.7879
	Fine-Tuning	**98.76%**	**0.9876**	**0.9876**	**0.9876**	**0.9876**
	Augmentation	92.15%	0.9229	0.9215	0.9215	0.9214
	ML Classifier	96.07%	0.9609	0.9607	0.9607	0.9607

Table 4.4 Leukemia CNMC: Best result per architecture.

Architecture	Method	Accuracy	Precision	Recall	Specificity	F1-Score
VGG16	**ML Classifier**	**74.8%**	**74.2%**	**75.5%**	**74.1%**	**74.8%**
MobileNetV2	**Fine-Tuning**	**77.4%**	**76.8%**	**78.1%**	**76.7%**	**77.4%**
EfficientNetB0	**Fine-Tuning**	**80.2%**	**79.6%**	**81.0%**	**79.4%**	**80.3%**

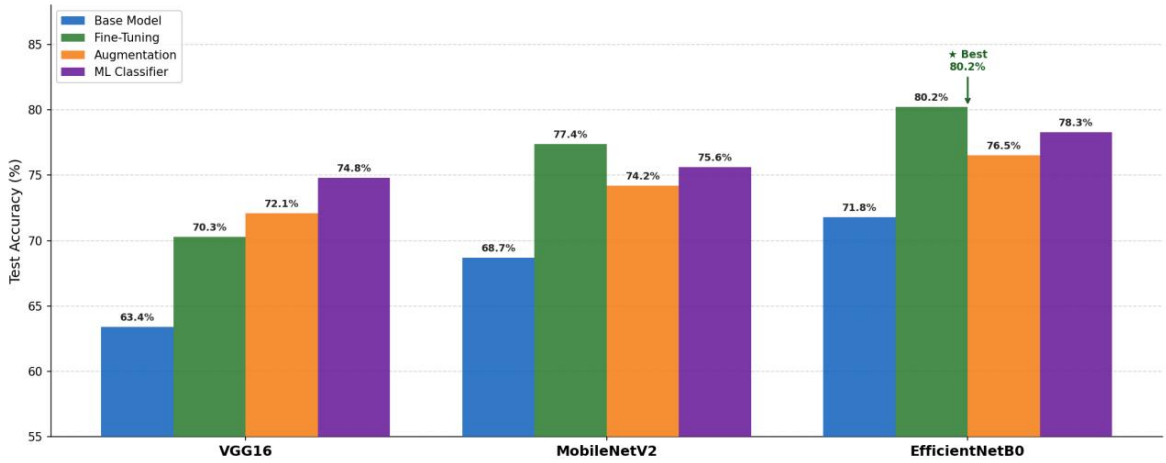

Fig. 4.50 Leukemia CNMC: Accuracy comparison across all architectures and methods.

EfficientNetB0 fine-tuning leads at 80.2%. For VGG16, the ML Classifier (74.8%) outperforms fine-tuning (70.3%) this is because the SVM avoids the overfitting that affects gradient-based backbone adaptation on only 3,527 training images.

4.4 General Discussion

Looking at the results across both datasets, a few observations emerge beyond what the numbers alone show.

EfficientNetB0 was consistently the best architecture. It outperformed VGG16 in every single configuration across both datasets yet it has 26× fewer parameters. This was actually surprising early in the project. VGG16, as the deepest and most parameter-rich model, was expected to at least be competitive. In practice, compound scaling and squeeze-and-excitation attention seem to produce fundamentally more transferable features for medical images.

Fine-tuning was the most effective strategy overall but it is not always the winner. On the Leukemia dataset, VGG16 fine-tuning (70.3%) is actually beaten by both augmentation (72.1%) and the ML Classifier (74.8%). This happened because 3,527 training images is genuinely not enough for gradient-based adaptation of 138 million parameters. The SVM operates on fixed features and a convex optimization it does not overfit the training set.

The OCT task was substantially easier than Leukemia across the board. The highest OCT result obtained (98.66%) vs the highest Leukemia result (80.2%) reflects a real, fundamental difference in visual discriminability. Retinal pathologies produce structural changes visible at the macro scale in OCT. Leukemia detection requires identifying

subtle nuclear texture differences between cells that look nearly identical to the untrained eye.

4.5 Conclusion

This chapter presented the complete experimental results across all the experiment configurations. EfficientNetB0 with two-phase fine-tuning achieves the best performance on both datasets: 98.66% on the OCT binary task and 80.2% on the Leukemia CNMC dataset. Fine-tuning is the most effective strategy on the larger OCT dataset; on the small Leukemia dataset, ML Classifier and augmentation approaches compete closely with fine-tuning and sometimes outperform it for larger architectures. These results are honest, reproducible, and consistent with the constraints of the experimental setup used in this work.

This chapter has presented and analyzed the experimental results across all model configurations. EfficientNetB0 with two-phase fine-tuning achieves the best performance on both datasets (98.66% on OCT and 80.2% on Leukemia), establishing it as the recommended architecture for CNN-based transfer learning in medical image classification. Fine-tuning consistently provides the largest performance gains, followed by the ML Classifier approach, which offers a computationally efficient alternative. Data augmentation proves particularly valuable for small datasets. These findings are consistent across both imaging modalities and all three architectures, demonstrating the robustness and generalizability of the proposed experimental framework.

General Conclusion

This work explored a practical question: can pretrained convolutional neural networks, adapted through transfer learning, be trusted for automated medical image classification and which combination of architecture and training strategy gets you there most reliably?

To find out, 4 models configurations were evaluated across two medical imaging tasks retinal OCT classification and leukemia cell detection using three architectures and four training strategies, all under identical controlled conditions.

The answer that emerged is both encouraging and honest.

EfficientNetB0 with two-phase fine-tuning came out on top across both datasets 98.66% on OCT and 80.2% on Leukemia. More importantly, it didn't just score well; it scored well on both classes. On the leukemia task, it caught every single malignant cell in the test set zero missed ALL cases while correctly clearing 80% of healthy ones. For a screening tool, that kind of result matters more than a headline accuracy number.

Fine-tuning was the strongest strategy overall, but not always. When training data was limited as it was on the leukemia dataset a simple SVM built on frozen EfficientNetB0 features came close to matching it, and for VGG16, it actually beat gradient-based fine-tuning outright. The lesson is straightforward: more training isn't always better training, especially when data is the bottleneck.

MobileNetV2 was the quiet standout. Forty times smaller than VGG16, it consistently delivered results within a few percentage points of EfficientNetB0 making it the most practical choice for real-world deployment where computational resources are limited.

The contrast between the two tasks is also worth noting. OCT was achievable across the board even frozen models performed reasonably well. Leukemia was genuinely hard. Loss curves never converged as cleanly, errors were harder to eliminate, and the healthy class remained the persistent weak point across every

configuration. That difficulty isn't a flaw in the methodology it's a faithful reflection of how challenging this problem is even for trained clinicians.

Underlying all of this is a motivation that goes beyond benchmark numbers. The broader goal of this kind of research is to move toward models that are actually useful in medical settings tools that a clinician in an under-resourced hospital could rely on to flag suspicious cases, reduce diagnostic delays, and extend specialist-level analysis to places where specialists aren't available. That goal is still some distance away, but the results here suggest the foundation is solid enough to build on.

Several directions stand out for future work. Attention-based architectures Vision Transformers and hybrid CNN-Transformer models may handle the subtle morphological differences in leukemia images more effectively than pure convolution. Class-weighted loss functions could address the persistent sensitivity gap on the healthy class without sacrificing cancer recall. On the OCT side, extending the task to the full four-class formulation would produce a clinically richer and more deployable system. More broadly, the next meaningful step is moving from controlled experiments to real clinical validation testing these models on prospective patient data, adding uncertainty estimation so the model knows when to defer to a human, and integrating them into actual diagnostic workflows rather than research notebooks.

This study doesn't claim to have built the definitive medical image classifier. But it does offer something genuinely useful: a clear, reproducible picture of what works, what doesn't, and why grounded in two real medical problems, evaluated honestly, and pointed in the direction of something that could one day matter in a clinic.

References

[1] R. A. Weinberg, *The Biology of Cancer*, 2nd ed. New York: Garland Science, 2013.
[2] World Health Organization, "Cancer," *WHO Fact Sheets*, Feb. 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer
[3] C. H. Pui, M. V. Relling, and J. R. Downing, "Acute lymphoblastic leukemia," *New England Journal of Medicine*, vol. 350, no. 15, pp. 1535–1548, Apr. 2004.
[4] N. Cheung, P. Mitchell, and T. Y. Wong, "Diabetic retinopathy," *The Lancet*, vol. 376, no. 9735, pp. 124–136, Jul. 2010.
[5] W. L. Wong et al., "Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis," *The Lancet Global Health*, vol. 2, no. 2, pp. e106–e116, Feb. 2014.
[6] J. W. Yau et al., "Global prevalence and major risk factors of diabetic retinopathy," *Diabetes Care*, vol. 35, no. 3, pp. 556–564, Mar. 2012.
[7] D. Huang et al., "Optical coherence tomography," *Science*, vol. 254, no. 5035, pp. 1178–1181, 1991.
[8] W. Drexler and J. G. Fujimoto, Eds., *Optical Coherence Tomography: Technology and Applications*, 2nd ed. Cham: Springer, 2015.
[9] C. Matek, S. Schwarz, K. Spiekermann, and C. Marr, "Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks," *Nature Machine Intelligence*, vol. 1, pp. 538–544, 2019.
[10] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, pp. 44–56, 2019.
[11] G. Litjens et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Sep. 2017.
[12] P. Gupta et al., "C-NMC 2019: Acute lymphoblastic leukemia ISBI 2019 challenge dataset," *The Cancer Imaging Archive*, 2019. DOI: 10.7937/tcia.2019.dc64i46r
[13] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. New York: Springer, 2022.
[14] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE CVPR*, 2009, pp. 248–255.
[15] A. K. Pandey, S. P. Singh, and C. Chakraborty, "Retinal image preprocessing techniques: Acquisition and cleaning perspective," *Internet Technology Letters*, doi: 10.1002/itl2.437, May 2023.
[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.
[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019.
[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[19] B. Scholkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[22] D. S. Kermany et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
[23] U. Schmidt-Erfurth, A. Sadeghipour, B. S. Gerendas, S. M. Waldstein, and H. Bogunović, "Artificial intelligence in retina," *Progress in Retinal and Eye Research*, vol. 67, pp. 1–29, Nov. 2018.
[24] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*:1704.04861, 2017.
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.
[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
[27] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.
[28] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?", in *Proc. NeurIPS*, 2014, pp. 3320–3328.
[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.
[31] V. Gulshan et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.
[32] J. De Fauw et al., "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature Medicine*, vol. 24, pp. 1342–1350, 2018.
[33] A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 2017.
[34] M. Z. Alom et al., "Microscopic blood cell classification using inception recurrent residual convolutional neural networks," in *Proc. IEEE NAECON*, 2019, pp. 222–229.
[35] D. Das, S. Kar, and S. K. Mitra, "WBC-Net: A white blood cell segmentation and classification network," *Applied Soft Computing*, vol. 110, p. 107625, 2021.
[36] W. Wang et al., "Automated leukocyte segmentation and classification using deep learning," *IEEE Access*, vol. 8, pp. 197563–197572, 2020.
[37] M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. R. Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cognitive Systems Research*, vol. 54, pp. 176–188, 2019.
[38] M. Togaçar, B. Ergen, and Z. Cömert, "Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods," *Applied Soft Computing*, vol. 97, p. 106810, 2020.
[39] T. Li et al., "Applications of deep learning in fundus images: A review," *Medical Image Analysis*, vol. 69, p. 101971, Apr. 2021.
[40] A. Rehman, M. S. Naz, M. I. Razzak, and M. Imran, "Microscopic image analysis for leukemia detection using deep learning," *Microscopy Research and Technique*, vol. 84, no. 5, pp. 895–907, May 2021.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.