INTERNSHIP REPORT

on

Histopathological Image Analysis Through Deep Learning-Based Image Classification Techniques

Submitted by

YASHARTH KESARWANI

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Under the supervision of

Dr. Joohi Chauhan
Assistant Professor
Department of Computer Science and Engineering
MNNIT Allahabad, Prayagraj – 211004

Internship Duration: 2nd June – 13th July, 2025

Department of Computer Science and Engineering
Motilal Nehru National Institute of Technology, Allahabad

July 2025

Contents

Acknowledgment **2**

Abstract **3**

Learning Process **4**

1 About MNNIT **5**

2 Introduction **6**

2.1 Objective 6

2.2 Overview – Deep Learning and Supervised Learning 7

2.3 Residual Networks (ResNet) 8

2.4 EfficientNet 11

3 IDEs, Frameworks, and Libraries **14**

3.1 Google Colab and Jupyter Notebook 14

3.2 Machine Learning with Python 14

4 Datasets Used **15**

4.1 PatchCamelyon (PCam) 15

4.2 BReAst Carcinoma Subtyping (BRACS) 16

5 Histopathological Image Analysis Through Deep Learning-Based Image Classification Techniques **18**

5.1 Implementation 1: Histopathologic Cancer Detection Using ResNet-50 on PatchCamelyon Dataset 18

5.2 Implementation 2: Histopathologic Cancer Detection Using ResNet-34 on PatchCamelyon Dataset 20

5.3 Implementation 3: Histopathologic Cancer Detection Using Enhanced ResNet-34 on PatchCamelyon Dataset 23

5.4 Implementation 4: Histopathologic Cancer Detection Using Refined ResNet-34 on PatchCamelyon Dataset 26

5.5 Implementation 5: Histopathologic Cancer Detection Using EfficientNet-B3 on BRACS Dataset 29

5.6 Implementation 6: Histopathologic Cancer Detection Using EfficientNet-B4 on BRACS Dataset 31

5.7 Implementation 7 : EfficientNet-B3 for BRACS Histopathologic Cancer Detection 36

5.8 Implementation 8 : EfficientNet-B3 for BRACS Histopathologic Cancer De- tection 40

6 Conclusion **47**

Bibliography **48**

Acknowledgment

Immersive academic experiences in unfamiliar settings often shed light on one's scholarly character as much as they illuminate the subject matter itself. This concise yet profoundly enriching internship transcended mere technical learning, offering a valuable opportunity to contemplate the discipline, precision, and accountability that underpin impactful research.

I am deeply thankful to **Dr. Joohi Chauhan** from the Department of Computer Science and Engineering at MNNIT Allahabad for her exceptional mentorship. Her insightful, timely, and well-calibrated guidance brought clarity and focus to a project that could have easily veered off course. It is truly special to work under a mentor who so adeptly balances granting independence with providing meticulous support.

The vibrant academic environment at MNNIT Allahabad, marked by intellectual rigor and a collaborative spirit, fostered an ideal setting for meaningful exploration and concentrated study. I extend my gratitude to the faculty, staff, and fellow students who cultivated such a dynamic and supportive atmosphere.

Lastly, I express my sincere appreciation to **SRM Institute of Science and Technology, Kattankulathur**, for nurturing a culture of inquiry, critical thought, and independent pursuit. The principles instilled by this institution have been a guiding force throughout this internship and will continue to influence my academic and professional path.

Abstract

This report presents an account of a research internship undertaken at **Motilal Nehru National Institute of Technology, Allahabad**, under the guidance of **Dr. Joohi Chauhan**, during the summer of 2025. The focus of the internship was the application of deep learning techniques to the field of histopathological image analysis, with the objective of building effective image classification models for medical diagnostics.

The initial phase of the research involved a detailed study of the challenges specific to histopathological imaging, such as high-resolution input, stain variability, and class imbalance. Preprocessing techniques—including normalization and data augmentation—were employed to prepare the datasets for model training.

The core of the project centered around exploring and implementing Convolutional Neural Networks (CNNs) using two datasets: the publicly available PatchCamelyon (PCam) dataset and the BRACS dataset, obtained through supervised institutional access. Deep CNN architectures, specifically ResNet and EfficientNet, were trained and evaluated on these datasets. Model performance was assessed using training and validation metrics, including accuracy, precision, and recall. Optimization strategies such as learning rate scheduling, regularization, and early stopping were adopted to enhance training stability and improve generalization.

This internship served as a significant step in bridging theoretical understanding with practical application in the context of biomedical image analysis. The report outlines the models implemented, the experimental findings, and the broader reflections that emerged during the course of this research experience.

Learning Process

The six-week research internship at **Motilal Nehru National Institute of Technology (MNNIT), Allahabad** was structured to provide a comprehensive balance between theoretical grounding and hands-on application in the field of **deep learning**, with a particular emphasis on **histopathological image classification**.

The initial phase of the internship focused on building a strong understanding of machine learning and deep learning principles. I studied supervised and semi-supervised learning approaches, gaining insights into how both fully and partially labeled datasets can be effectively utilized for training. This theoretical foundation helped contextualize the practical challenges of medical image analysis.

A major portion of the work was devoted to exploring and implementing Convolutional Neural Networks (CNNs). I examined the structure and function of key components such as convolutional layers, padding, pooling, and activation functions, while also studying architectural innovations that enhance feature extraction and model performance.

Hands-on implementation began with the **PatchCamelyon (PCam)** dataset, where I trained ResNet-34, ResNet-50 and EfficientNet-B3 models. These were evaluated using training and validation accuracy, as well as precision, recall, and F1 score. To mitigate overfitting and enhance model generalization, techniques such as dropout, batch normalization, early stopping, and learning rate scheduling were applied throughout the training process.

Building on these experiments, I gained supervised access to the **BRACS** dataset, a more complex and high-resolution dataset. Here, I extended the experimentation by implementing EfficientNet-B3 and EfficientNet-B4, which enabled further performance evaluation under more realistic and challenging histopathological conditions.

Toward the conclusion of the internship, I began exploring the theoretical foundations of Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) architectures, and the transformer model. While these were not implemented, the study provided a conceptual understanding of their mechanisms and potential applications in sequential and context-rich data analysis.

Over the course of this **six-week period**, the internship offered a rigorous and rewarding research experience—bridging theoretical depth with practical experimentation—and significantly strengthened my understanding of how deep learning can be applied to biomedical image analysis.

1 About MNNIT

Motilal Nehru National Institute of Technology (MNNIT) Allahabad, situated in the historic city of Prayagraj, Uttar Pradesh, is recognized as one of India's premier technical institutions. Established in 1961 and designated as an Institute of National Importance, MNNIT has consistently upheld a legacy of academic excellence, rigorous research, and high-quality engineering education.

The institute offers a comprehensive portfolio of undergraduate, postgraduate, and doctoral programs, with curricula that are continually updated to align with the evolving demands of industry and academia. Its faculty comprises distinguished educators and accomplished researchers, contributing to a robust academic environment. MNNIT also boasts an impressive placement record, reflecting its commitment to student success beyond the classroom. Complementing its academic strengths is a dynamic campus life, enriched by numerous student-led technical and cultural initiatives.

The internship experience at MNNIT was characterized by a culture of purposeful engagement—structured yet adaptable, fostering both intellectual rigor and creative exploration. I was fortunate to receive expert guidance while being granted the independence necessary for self-directed research, striking an effective balance between mentorship and autonomy. This environment supported a deliberate and reflective learning process, enabling a deep and nuanced understanding of the subject matter.

Overall, MNNIT provided not only the resources and infrastructure of a leading institution but also an academic atmosphere marked by scholarly seriousness—an ideal setting in which ideas could be pursued with both freedom and discipline.

2 Introduction

2.1 Objective

The objective of this internship was to design and implement deep learning models for histopathological image classification, aimed at improving diagnostic accuracy in healthcare. A central focus was on developing and training Convolutional Neural Networks (CNNs), specifically ResNet and EfficientNet architectures, to classify medical images from histopathology datasets. These models have significant potential to assist pathologists by providing automated, data-driven analysis of tissue samples, thereby supporting early and accurate diagnosis of diseases such as cancer. The internship also involved exploring advanced training techniques and evaluation metrics to optimize model performance, with detailed discussions presented in the following sections.

2.2 Overview – Deep Learning and Supervised Learning

Deep learning is a rapidly advancing subfield of artificial intelligence that enables machines to automatically learn hierarchical and abstract patterns from large-scale data. At its foundation lies the artificial neural network—a computational model inspired by the biological structure of the human brain. These networks are composed of layers of interconnected nodes (neurons) that transform input data through weighted operations and activation functions.

Within the paradigm of supervised learning, deep learning models are trained on labeled datasets, learning to map inputs to known outputs. This approach is particularly effective in image classification, natural language processing, and other domains where large annotated datasets are available.

A key breakthrough in deep learning for visual data has been the development of Convolutional Neural Networks (CNNs). CNNs use convolutional layers to detect local features and progressively build higher-level representations, making them especially powerful for image-related tasks. Architectures such as ResNet (Residual Networks) introduced the concept of skip connections to mitigate vanishing gradient issues and allow the training of deeper models. Meanwhile, EfficientNet architectures employ a compound scaling method to balance network depth, width, and resolution, achieving high accuracy with fewer parameters and improved computational efficiency.

Convolutional Neural Networks:

Convolutional Neural Networks (CNNs) are a class of deep neural networks specifically designed to process data with a grid-like topology, such as images. Unlike traditional fully connected networks, CNNs use convolutional layers to capture spatial hierarchies in data by detecting local patterns through learnable filters.

1. Convolutional Layer

The core component of a CNN is the convolutional layer, which applies a set of kernels (filters) that slide over the input image. Each filter computes a dot product between its weights and the input patch it covers, producing a feature map. Each convolutional filter is capable of detecting specific features such as edges, textures, or more complex patterns as depth increases.

Mathematically, for input image I and kernel K, the convolution operation is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n) \tag{1}$$

2. Activation Function

After convolution, the output is passed through a non-linear activation function to allow the network to learn complex mappings. A commonly used activation function is the Rectified Linear Unit (ReLU):

$$\phi(z) = \max(0, z) \tag{2}$$

This non-linearity helps the network capture complex relationships and mitigates issues like the vanishing gradient problem.

3. Pooling Layer

To reduce spatial dimensions and computation, CNNs use pooling layers. These layers downsample feature maps while retaining important spatial features.

- **Max Pooling**: Selects the maximum value within a defined window (e.g., 2 × 2).
- **Average Pooling**: Computes the average of the values in the window.

Pooling improves robustness to small translations and helps prevent overfitting.

4. Flattening and Fully Connected Layers
After several convolution and pooling operations, the resulting multidimensional feature maps are flattened into a one-dimensional vector. This flattened output is passed through one or more fully connected layers, which perform high-level reasoning and classification tasks based on the extracted features.

5. Output Layer
The final layer of the CNN typically uses:

- **Softmax activation**: for multi-class classification tasks
- **Sigmoid activation**: for binary classification tasks

These activation functions convert the output into class probabilities, enabling decision-making based on model confidence.

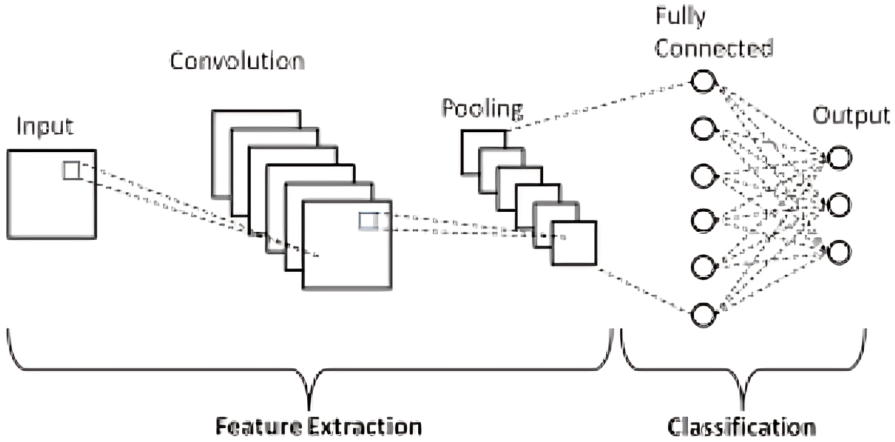


Figure 1: Illustration of a typical Convolutional Neural Network (CNN) architecture, showing the sequential flow of data through convolution, activation, pooling, flattening, fully connected, and output layers.

2.3 Residual Networks (ResNet)

Deep Convolutional Neur Networks (ResNet) introduce a novel framework called **residual learning**, enabling the effective training of neural networks with considerably greater depth.

The Problem with Deep Networks

When additional layers are added to a conventional CNN, one might expect increased representational power and improved accuracy. However, empirical observations reveal that deeper models may actually yield **higher training error**, indicating an optimization issue—not merely overfitting. This degradation suggests that it becomes increasingly difficult for deeper networks to approximate identity mappings, even when such mappings would be optimal for some layers.

Residual Learning: Core Concept

ResNet addresses this problem by reformulating the layers to explicitly learn a **residual function**, rather than the direct mapping. Let $\mathbf{x}$ denote the input to a set of layers and $\mathcal{H}(\mathbf{x})$ be the desired underlying function. Instead of learning $\mathcal{H}(\mathbf{x})$ directly, ResNet trains the network to learn the residual:

$$\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

This leads to the reformulated output of the residual block as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$$

Here:

- $\mathbf{x}$ is the input to the residual block.
- $\mathcal{F}(\mathbf{x}, \{W_i\})$ is the residual function to be learned (typically composed of convolutional, batch normalization, and activation layers).
- $\{W_i\}$ denotes the weights of the convolutional layers.
- $\mathbf{y}$ is the output of the residual block.

The term $\mathbf{x}$ is added element-wise to the output of $\mathcal{F}$, forming a **shortcut connection** or **identity mapping**. This addition operation assumes that the dimensions of $\mathcal{F}(\mathbf{x})$ and $\mathbf{x}$ match. If they do not, a linear projection (e.g., using a 1×1 convolution) can be used:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}$$

where W_s is the projection matrix (learned).

Structure of a Residual Block

A **basic residual block** in ResNet typically includes:

- Two 3×3 convolutional layers
- Batch normalization layers after each convolution
- ReLU activation function after the first convolution
- An identity (or projection) shortcut connection that adds the input to the output

The full block is represented as:

$$\mathcal{F}(\mathbf{x}) = \mathrm{ReLU}(\mathrm{BN}(W_2 \cdot \mathrm{ReLU}(\mathrm{BN}(W_1 \cdot \mathbf{x}))))$$

and the final output becomes:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}$$

This residual addition is then optionally passed through another ReLU activation.

Network Architecture Variants

Different ResNet architectures differ primarily in depth and the type of residual blocks used:

Architecture	Block Type	Depth	Description
ResNet-18	Basic Block	18	Two 3×3 convolutions per block
ResNet-34	Basic Block	34	Deeper version of ResNet-18
ResNet-50	Bottleneck Block	50	1×1 → 3×3 → 1×1 convolutions
ResNet-101	Bottleneck Block	101	Deeper variant using same block as ResNet-50
ResNet-152	Bottleneck Block	152	Extremely deep architecture with over 60M params

The **bottleneck block** improves computational efficiency by using 1×1 convolutions to reduce and then restore dimensionality around a 3×3 convolution:

$$\mathcal{F}(\mathbf{x}) = W_3 \cdot \mathrm{ReLU}(\mathrm{BN}(W_2 \cdot \mathrm{ReLU}(\mathrm{BN}(W_1 \cdot \mathbf{x}))))$$

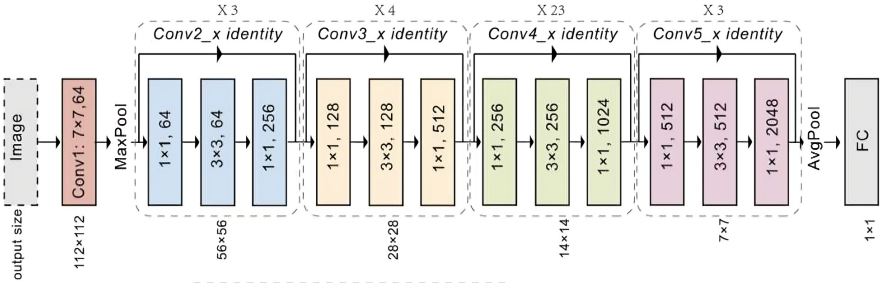


Figure 2: Illustration of a Residual Network (ResNet) architecture, highlighting residual blocks with skip connections that allow the network to learn identity mappings and improve gradient flow in deep convolutional models.

Benefits of ResNet

- **Improved gradient flow**: Skip connections help gradients bypass multiple layers, mitigating vanishing gradients.
- **Identity mapping**: Enables the network to preserve important features when no further transformation is required.
- **Better generalization**: Residual learning regularizes the model implicitly.

- **Scalability**: ResNet architectures scale well with depth and perform competitively across a range of vision tasks.

Application to Histopathological Image Classification

In the context of histopathological image analysis, ResNet provides an effective mechanism for extracting both low-level (e.g., textures, edges) and high-level (e.g., cellular structures) features. Its residual connections are particularly useful when dealing with complex, high-resolution images, as they help stabilize training and allow deeper exploration of morphological patterns critical to disease classification. In this study, ResNet architectures (including ResNet-34 and ResNet-50) were employed and evaluated on patch-based histopathology datasets, yielding competitive performance in terms of classification accuracy, precision, and recall.

2.4 EfficientNet

EfficientNet is a family of convolutional neural network architectures developed to achieve optimal accuracy and efficiency through a novel scaling strategy. Unlike traditional CNNs that scale arbitrarily in depth or width to improve performance, EfficientNet employs a **compound scaling method** that uniformly scales depth, width, and input resolution using a principled and empirically validated approach. This results in state-of-the-art performance on image classification benchmarks while maintaining significantly fewer parameters and lower computational cost.

Motivation

Standard CNN architectures such as ResNet and Inception improve accuracy by scaling network depth (adding layers), width (more channels), or input resolution (larger images). However, scaling only one of these dimensions often leads to diminishing returns or overfitting. EfficientNet addresses this limitation by introducing a **balanced scaling technique** that improves accuracy and efficiency simultaneously.

Compound Scaling

EfficientNet introduces a **compound coefficient** $\phi \in \mathbb{R}$ to control model scaling. Given a baseline network, EfficientNet uniformly scales:

- **Depth** $d \to \alpha^{\phi}$
- **Width** $w \to \beta^{\phi}$
- **Input resolution** $r \to \gamma^{\phi}$

The constants α, β, γ are determined through a small grid search and are subject to the constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

This constraint ensures that the total FLOPs roughly double when ϕ increases by 1.

The result is a family of models, EfficientNet-B0 to EfficientNet-B7, where B0 is the baseline and others are scaled versions. EfficientNet-B0 itself is constructed using a carefully tuned neural architecture found via **Neural Architecture Search (NAS)**.

Building Blocks

EfficientNet is composed of **MBConv blocks** (Mobile Inverted Bottleneck Convolution), which are adapted from MobileNetV2. Each MBConv block includes:

- A pointwise 1×1 convolution for dimensionality expansion
- A depthwise separable 3×3 or 5×5 convolution
- A squeeze-and-excitation (SE) module to model inter-channel dependencies
- A final 1×1 pointwise convolution to project back to the desired dimensions

The activation function used is **Swish**, defined as:

$$\text{Swish}(x) = x \cdot \sigma(x)$$

where $\sigma(x)$ is the sigmoid function. Swish provides smoother gradients and improves convergence over ReLU in deeper networks.

Architecture Overview

The EfficientNet-B0 architecture contains the following key stages:

- **Stem**: One 3×3 convolution layer with stride 2
- **MBConv Blocks**: Stacked across several stages with increasing number of channels and decreasing spatial resolution
- **Head**: Global average pooling followed by dropout and a fully connected dense layer for classification

Each version from B1 to B7 increases model size, input resolution, and computational cost while maintaining the architectural skeleton.

Model	Parameters	Input Resolution	Top-1 Accuracy
EfficientNet-B0	5̃.3M	224 × 224	7̃7.1%
EfficientNet-B3	1̃2M	300 × 300	8̃1.6%
EfficientNet-B4	1̃9M	380 × 380	8̃2.9%

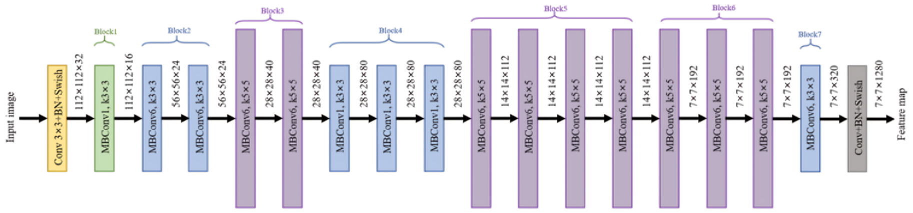


Figure 3: EfficientNet architecture showcasing compound scaling across depth, width, and resolution, along with its modular MBConv blocks and squeeze-and-excitation mechanisms.

Advantages

- **High accuracy with fewer parameters**: EfficientNet-B4 achieves higher accuracy than ResNet-152 while using significantly fewer parameters.
- **Computational efficiency**: Well-suited for training and inference on resource-constrained devices.
- **Scalable architecture**: Flexibly scales up or down based on available resources and task complexity.

Application in Histopathology

For histopathological image classification tasks, EfficientNet offers a strong balance between **computational cost** and **representational power**. Its depthwise separable convolutions and SE blocks make it highly effective at identifying fine-grained patterns in tissue structure while avoiding overfitting. In this study, **EfficientNet-B3** and **EfficientNet-B4** were used to classify cancerous vs. non-cancerous tissue using datasets such as **PatchCamelyon (PCam)** and **BRACS**. The models were evaluated using metrics including training and validation **accuracy, precision, and recall**, demonstrating superior performance compared to baseline CNNs.

3 IDEs, Frameworks, and Libraries

3.1 Google Colab and Jupyter Notebook

Google Colaboratory (Colab) is a cloud-based Jupyter notebook environment provided by Google that enables users to write, execute, and share Python code with minimal setup. It offers free access to GPUs and TPUs, making it an excellent platform for deep learning experimentation and prototyping. Key features of Colab include GPU/TPU acceleration without requiring local hardware, integration with Google Drive for persistent storage, real-time collaboration similar to Google Docs, and easy installation of Python libraries.

Alongside Google Colab, Jupyter Notebook was also extensively used as a local interactive environment. Jupyter Notebook supports live code execution, rich media outputs, and facilitates iterative development, making it suitable for data exploration, model building, and visualization.

During the internship, both Google Colab and Jupyter Notebook were employed for model development and experimentation. TensorFlow was used as the primary deep learning framework. For computationally intensive tasks like training convolutional neural networks, GPU acceleration in Colab was leveraged to enhance performance.

3.2 Machine Learning with Python

Python is the preferred language for machine learning and deep learning due to its readability, flexibility, and vast ecosystem of libraries. The key libraries utilized during the internship include:

- **NumPy** and **Pandas** for efficient data manipulation and preprocessing
- **Matplotlib** and **Seaborn** for data visualization and exploratory analysis
- **Scikit-learn** for dataset splitting, preprocessing utilities, and performance evaluation
- **TensorFlow** for constructing, training, and deploying artificial neural networks (ANN) and convolutional neural networks (CNN)

TensorFlow was chosen for its comprehensive ecosystem, scalability, and user-friendly APIs such as Keras, which facilitated rapid prototyping and intuitive debugging throughout the research and experimentation process.

4 Datasets Used

This research leverages two prominent histopathological image datasets—PatchCamelyon (PCam) and BReAst Carcinoma Subtyping (BRACS). These datasets represent two different yet clinically significant challenges in computational pathology: binary classification of metastatic cancer and multiclass subtyping of breast carcinoma. Their inclusion enables evaluation of both convolutional neural networks (CNNs) and other deep learning architectures in medical image analysis tasks.

4.1 PatchCamelyon (PCam)

PatchCamelyon (PCam) is a large-scale benchmark dataset designed for binary image classification in the field of digital pathology. It was introduced to support the development of machine learning models capable of detecting metastatic tissue in lymph nodes. The dataset is derived from the Camelyon16 Challenge, which focuses on the detection of cancer metastases in whole-slide images (WSIs) of lymph node sections stained with hematoxylin and eosin (H&E).

Dataset Composition:

- **Total Images:** 327,680 RGB image patches
- **Image Dimensions:** 96 × 96 pixels
- **Train/Validation/Test Split:** 262,144 / 32,768 / 32,768 images
- **Label Distribution:** Balanced; 50% positive (tumor), 50% negative
- **File Format:** HDF5 (.h5.gz), with corresponding metadata in CSV

Annotation Strategy: Each image patch is assigned a binary label indicating whether the central 32 × 32 pixel region contains any metastatic (tumor) tissue. Tissue in the outer region is not considered for labeling, although it remains part of the input image to support training of fully convolutional networks.

Preprocessing and Augmentation:

- Pixel value normalization to [0,1] range
- Data augmentation: horizontal and vertical flips, width/height shifts, rotations
- Patch balancing using hard-negative mining to maintain class balance

Machine Learning Use:

- **Task:** Binary image classification (presence vs. absence of tumor)
- **Model:** Convolutional Neural Network (CNN)
- **Goal:** Enable early-stage cancer detection from histological image patches

Advantages:

- Intermediate scale: Larger than CIFAR-10 but smaller than ImageNet
- Suited for training deep models on single-GPU systems
- High-quality medical image data with strong clinical relevance

Source: `https://zenodo.org/record/2546921` **Citation:** B. S. Veeling et al., "Rotation Equivariant CNNs for Digital Pathology," arXiv:1806.03962

4.2 BReAst Carcinoma Subtyping (BRACS)

The BRACS dataset is a curated collection of high-resolution H&E-stained whole-slide images (WSIs) and regions of interest (RoIs) representing various subtypes of breast carcinoma. Developed through collaboration between IRCCS Fondazione Pascale, CNR-ICAR, and IBM Research-Zurich, BRACS is among the few publicly available datasets that include both typical and atypical lesions in breast tissue histopathology.

Dataset Composition:

- **Whole Slide Images (WSIs):** 547 slides from 189 patients
- **Regions of Interest (RoIs):** 4,537 image crops from 387 WSIs
- **Image Resolution:** Up to 100,000 × 100,000 px (WSIs), and 4,000 × 4,000 px (RoIs)
- **Magnification:** 40× (0.25 µm/pixel)
- **File Formats:** WSIs in .svs; RoIs in .png

Annotation Classes: BRACS categorizes samples into 7 distinct histopathological subtypes, enabling multiclass classification tasks:

- **Group_BT (Benign/Typical):** Normal (N), Pathological Benign (PB), Usual Ductal Hyperplasia (UDH)
- **Group_AT (Atypical Lesions):** Flat Epithelial Atypia (FEA), Atypical Ductal Hyperplasia (ADH)
- **Group_MT (Malignant Tumors):** Ductal Carcinoma in Situ (DCIS), Invasive Carcinoma (IC)

Class Distribution (RoIs):

- **Training Set:** 3,657 RoIs
- **Validation Set:** 312 RoIs
- **Test Set:** 576 RoIs
- **Balanced samples across subtypes to enable robust training**

Annotation Protocol: All images were reviewed and labeled by three expert pathologists at the National Cancer Institute in Naples, Italy. Annotations include patient IDs, subtype labels, and corresponding WSI identifiers, ensuring traceability and structured organization.

Machine Learning Use:

- **Task:** Multiclass classification of breast carcinoma subtypes
- **Objective:** Enable early detection and subtyping of atypical and malignant lesions
- **Modeling:** CNN architectures with preprocessing (resizing, normalization)

Significance:

- Enables differentiation between benign, atypical, and malignant lesions
- Atypical lesion detection (e.g., FEA, ADH) can aid early diagnosis and intervention
- High-resolution RoIs offer rich spatial detail for feature extraction

Data Access: Dataset made available under institutional agreements; annotations and file metadata provided in structured spreadsheet format.

5 Histopathological Image Analysis Through Deep Learning-Based Image Classification Techniques

5.1 Implementation 1: Histopathologic Cancer Detection Using ResNet-50 on PatchCamelyon Dataset

The primary objective of this implementation is to detect metastatic cancer tissue from histopathologic image patches using a deep convolutional neural network (ResNet-50). This task was formulated as a binary image classification problem (Tumor vs. Non-Tumor) using the PatchCamelyon (PCam) dataset.

All patches were resized to 224 × 224 pixels to match the input requirements of the ResNet-50 architecture.

Data Preprocessing and Augmentation

To improve generalization and reduce overfitting, the following preprocessing steps and augmentations were applied:

Preprocessing:

- Resizing from 96 × 96 to 224 × 224 pixels
- Pixel scaling to range [0, 1] by dividing by 255
- Standard normalization using ImageNet statistics:
  - Mean: [0.485, 0.456, 0.406]
  - Std Dev: [0.229, 0.224, 0.225]

Augmentation Techniques (Training Phase Only):

- Horizontal Flip (p = 0.5)
- Vertical Flip (p = 0.3)
- Random Rotation (p = 0.2; 90°, 180°, 270°)
- Color Jitter (p = 0.15; brightness ±10%)

Model and Training Configuration

The implementation leveraged a pre-trained ResNet-50 model, which was fine-tuned on the PCam dataset for binary classification.

Training Setup:

- **Batch Size:** 32
- **Epochs:** 10
- **Optimizer:** AdamW
- **Learning Rate Scheduler:** Cosine Annealing
  - Initial LR: 3×10^{-4}
  - Final LR: 1.5×10^{-4}
- **Early Stopping:** Enabled to avoid overfitting

Evaluation Metrics

The trained model was evaluated on the validation and test sets using the following metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**

Results

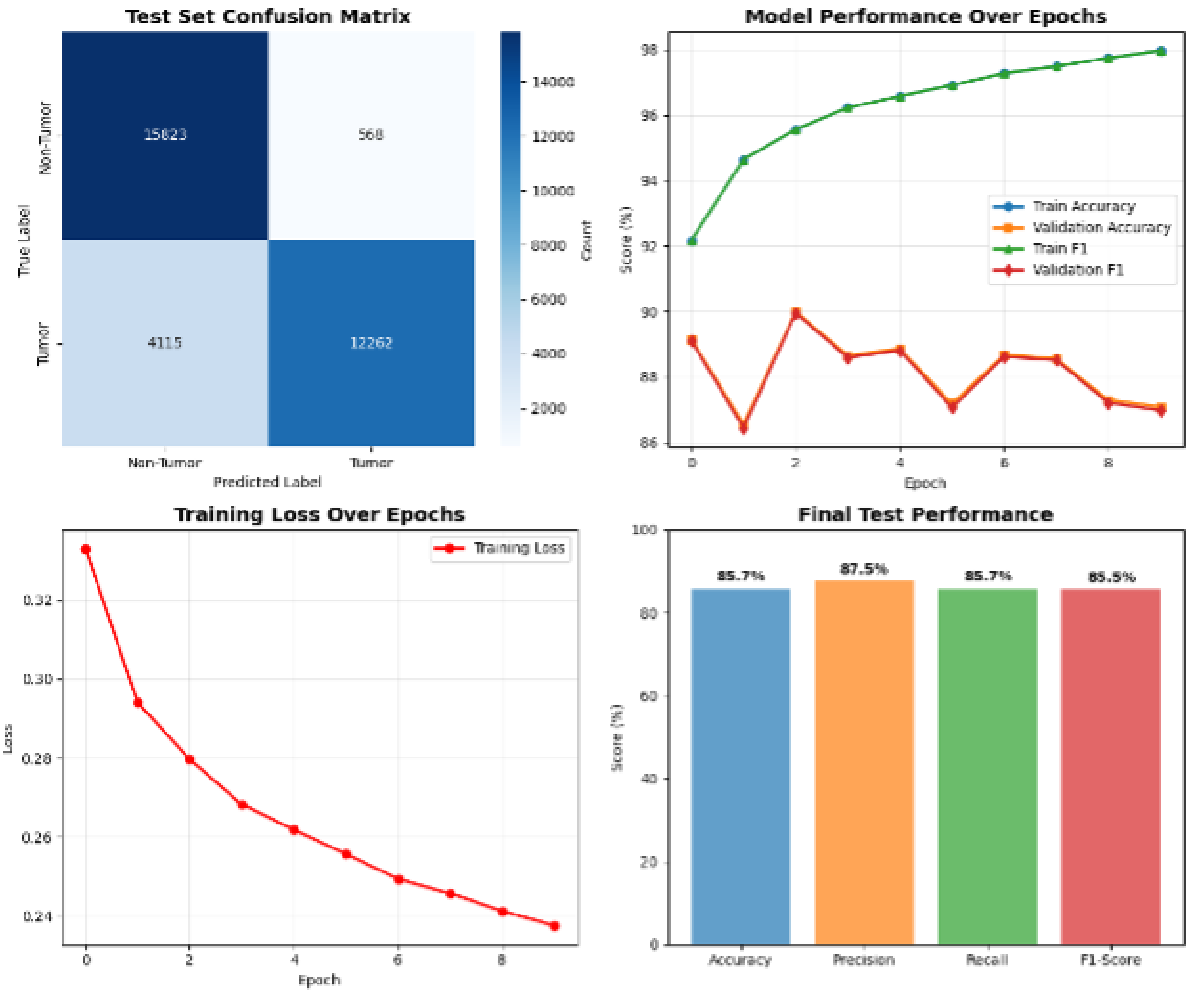


Figure 4: Evaluation Results on PCam Dataset: Confusion Matrix, Training Loss, Validation Metrics, and Final Test Performance

Validation Performance:

- Accuracy: 89.99%
- Precision: 90.65%

- Recall: 89.99%
- F1-score: 89.95%

Test Performance:

- Accuracy: 85.71%
- Precision: 87.46%
- Recall: 85.71%
- F1-score: 85.54%

5.2 Implementation 2: Histopathologic Cancer Detection Using ResNet-34 on PatchCamelyon Dataset

This section describes the detailed experimental procedure, model training setup, and evaluation metrics for the ResNet-34-based deep learning model for histopathologic cancer detection using the PatchCamelyon (PCam) dataset.

Data Preprocessing and Augmentation

To ensure uniformity and boost the generalizability of the model, the following preprocessing steps were applied:

- **Resizing:** All image patches were resized from 96 × 96 pixels to 224 × 224 pixels.
- **Tensor Conversion:** Converted from HWC (Height, Width, Channel) to CHW format.
- **Pixel Normalization:** Pixel values were normalized to the range $[0, 1]$ by dividing by 255.
- **Standard Normalization:** Normalization applied using ImageNet mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$.

Augmentation Techniques (Medium Strength):

- Horizontal Flip (p = 0.5)
- Vertical Flip (p = 0.3)
- Random Rotation (p = 0.4): Angles of 90°, 180°, or 270°

Model Training Configuration

The ResNet-34 model was trained with the following configuration:

- **Classifier Head:** A multi-layer classifier with progressive dropout rates of 0.6, 0.5, and 0.4.
- **Batch Size:** Physical batch size of 32; effective batch size of 64 using gradient accumulation.

- **Epochs:** 8 (early stopping enabled).
- **Optimizer:** AdamW.
- **Learning Rate Scheduler:** ReduceLROnPlateau with initial LR = 2×10^{-4}, final LR = 1×10^{-4}.

Validation Performance

- **Accuracy:** 88.15%
- **Precision:** 89.24%
- **Recall:** 88.15%
- **F1-Score:** 88.07%

Test Performance

- **Accuracy:** 81.82%
- **Precision:** 84.76%
- **Recall:** 81.82%
- **F1-Score:** 81.43%
- **ROC AUC:** 0.9251

Visualization

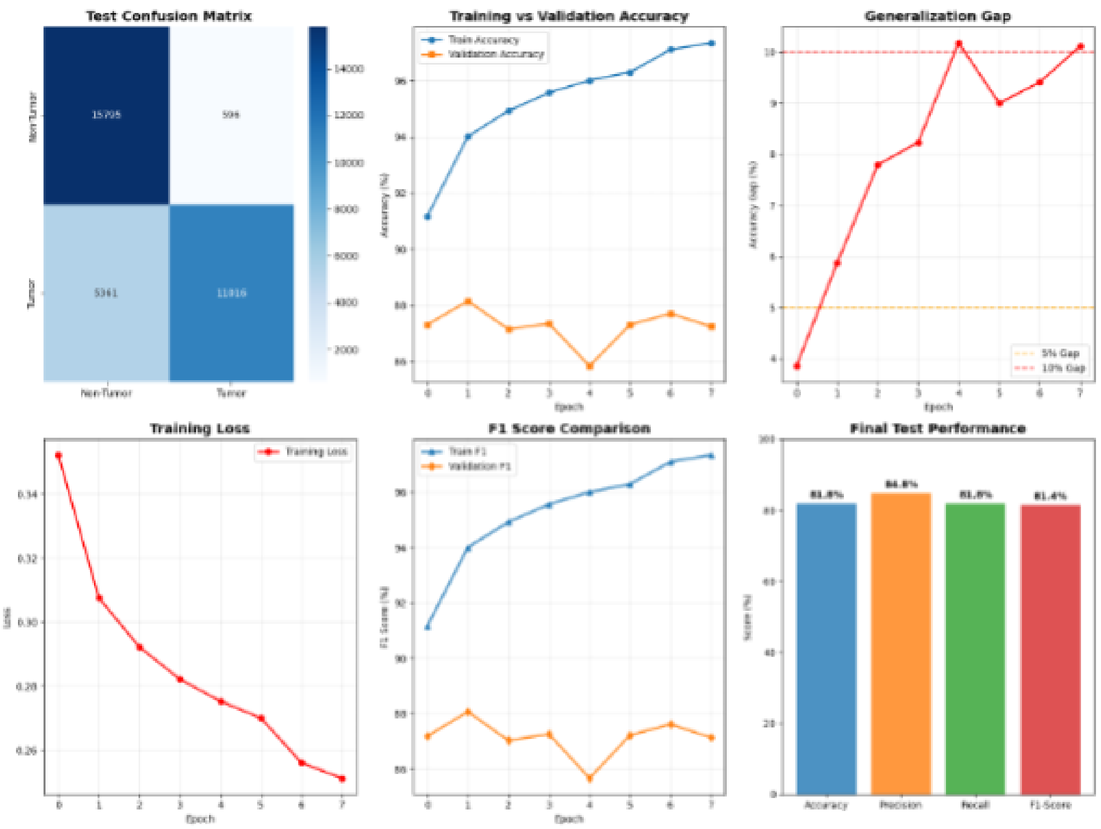


Figure 5: (Top-left) Confusion Matrix on Test Set, (Top-middle) Accuracy over Epochs,(Top-right)Generalization-Gap, (Bottom-left) Training Loss over Epochs, (Bottom-middle) F1-Score Comparison, (Bottom-right) Final Test Performance.

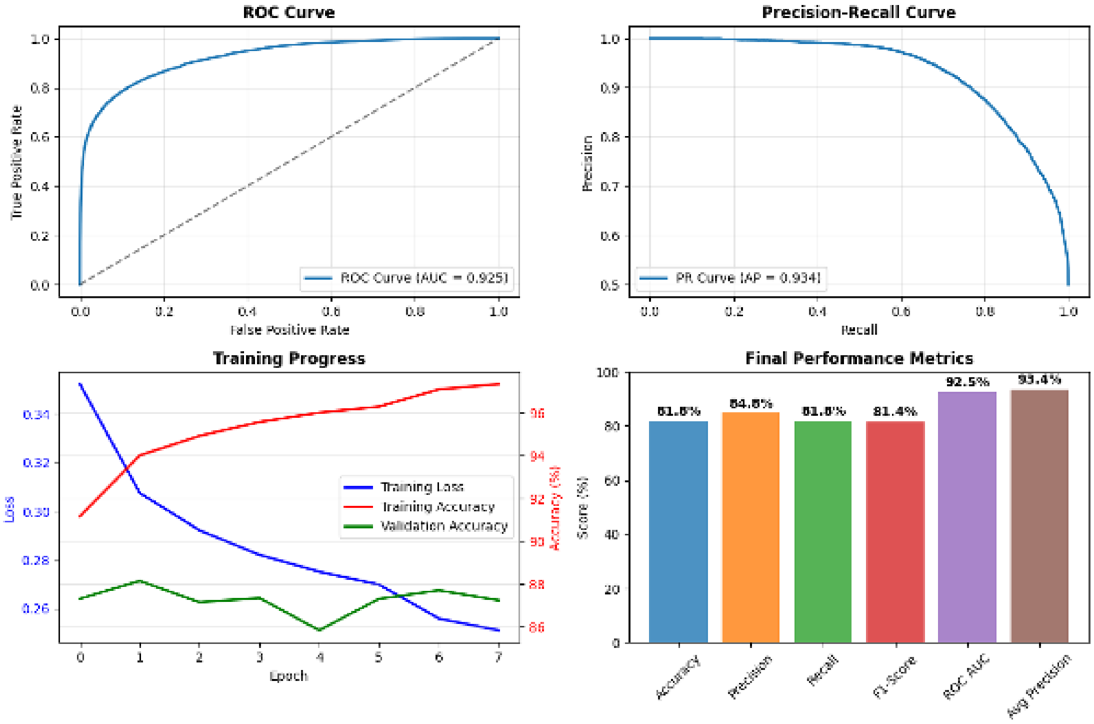


Figure 6: (Top-left) ROC Curve, (Top-right) Precision-Recall Curve, (Bottom-left) Training Progress, (Bottom-right) Final Performance Metrics.

5.3 Implementation 3: Histopathologic Cancer Detection Using Enhanced ResNet-34 on PatchCamelyon Dataset

This section elaborates on the training protocol, model architecture, data preprocessing strategies, and evaluation metrics used in the third implementation of histopathologic cancer detection using a modified ResNet-34 model on the PatchCamelyon dataset.

Data Preprocessing and Augmentation

To enhance model generalization, the following preprocessing and advanced augmentation strategies were applied:

- **Resizing:** All image patches were resized from 96×96 to 224×224 pixels.
- **Tensor Conversion:** Images were converted from HWC to CHW format.
- **Pixel Normalization:** Normalized to $[0, 1]$ by dividing pixel values by 255.
- **Standard Normalization:** Using ImageNet statistics: Mean $[0.485, 0.456, 0.406]$, Std $[0.229, 0.224, 0.225]$.

Aggressive Augmentation Techniques:

- Horizontal Flip (p = 0.5)
- Vertical Flip (p = 0.3)
- Random Rotation (p = 0.4): Rotations of 90°, 180°, or 270°
- Color Jittering (p = 0.5): Brightness and contrast scaled in $[0.8, 1.2]$
- Gaussian Noise (p = 0.3): Std. deviation of 0.05
- Random Erasing (p = 0.2): 1/8 image size patch set to zero

Model Architecture and Training Configuration

The model employed a modified ResNet-34 backbone pre-trained on ImageNet, fine-tuned with aggressive regularization and architectural adjustments:

- **Classifier Head:** Multi-layer classifier with progressive dropout rates of 0.7, 0.6, and 0.5.
- **Layer Freezing:** Only the final 20 layers of the ResNet-34 backbone were kept trainable.
- **Parameter Count:** Total parameters $\approx$ 21.7 million.
- **Classifier Compression:** Hidden layer dimensions reduced from 512 to 128 units.

Training Configuration:

- Batch Size: 32 (effective 64 via gradient accumulation)
- Epochs: 8 (with early stopping)

- Optimizer: AdamW with weight decay of 5×10^{-4}
- Learning Rate: Initial 1×10^{-4}, final 5×10^{-5}
- Scheduler: ReduceLROnPlateau (on validation loss)

Validation Performance

- **Accuracy:** 87.68%
- **Precision:** 88.60%
- **Recall:** 87.68%
- **F1-Score:** 87.60%

Test Performance

- **Accuracy:** 82.70%
- **Precision:** 85.20%
- **Recall:** 82.70%
- **F1-Score:** 82.39%

Visualization

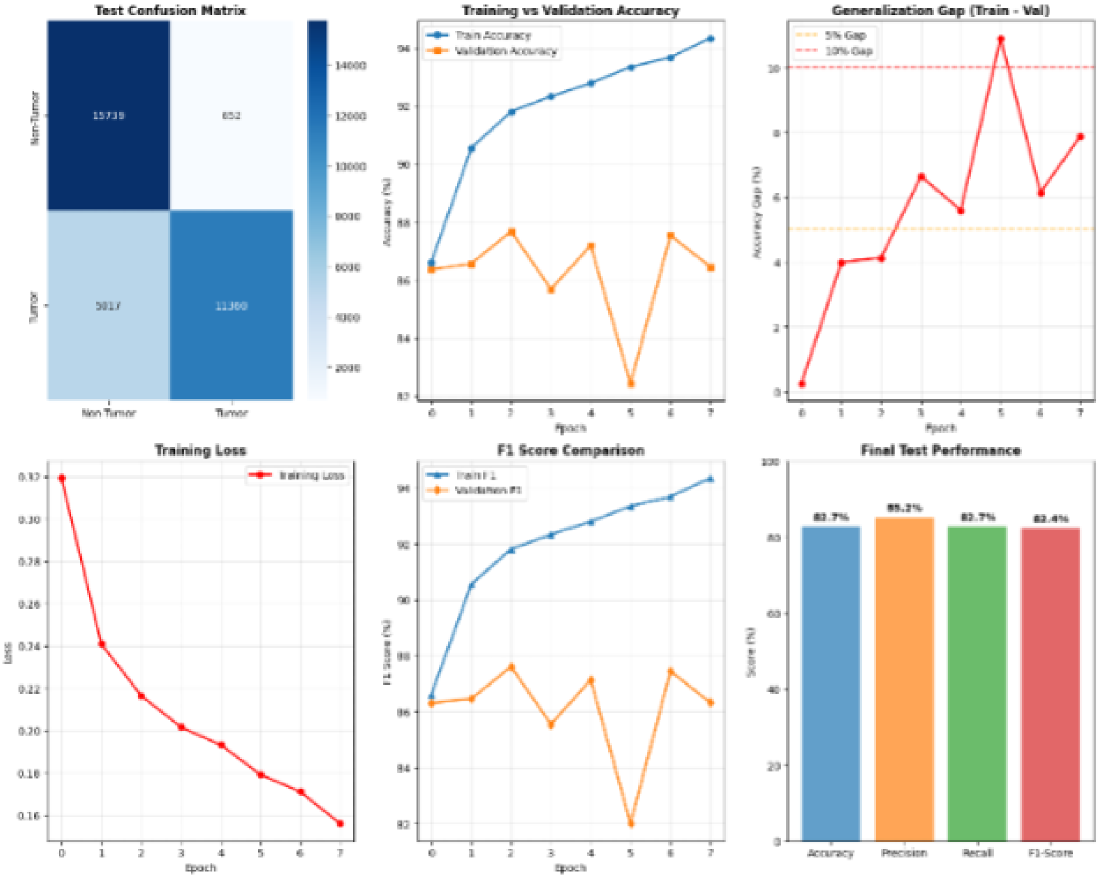


Figure 7: (Top-left) Confusion Matrix on Test Set, (Top-middle) Accuracy over Epochs, (Top-right) Generalization Gap, (Bottom-left) Training Loss over Epochs, (Bottom-middle) F1-Score Comparison, (Bottom-right) Final Test Performance.

5.4 Implementation 4: Histopathologic Cancer Detection Using Refined ResNet-34 on PatchCamelyon Dataset

This section presents the model training procedure, architectural improvements, augmentation strategies, and performance metrics for a refined ResNet-34 implementation for histopathologic cancer detection.

Data Preprocessing and Augmentation

- **Resizing:** Image patches were resized to 224 × 224 pixels.
- **Tensor Conversion:** Format conversion from HWC to CHW.
- **Pixel Normalization:** Pixel values scaled to the $[0, 1]$ range.
- **Standard Normalization:** Applied using ImageNet statistics: Mean $[0.485, 0.456, 0.406]$, Std $[0.229, 0.224, 0.225]$.

Augmentation Techniques (Reduced Strength):

- Horizontal Flip (p = 0.5)
- Vertical Flip (p = 0.25)
- Random Rotation (p = 0.25): 90°, 180°, 270°
- Color Jittering (p = 0.3): Brightness and contrast adjustments
- Gaussian Noise (p = 0.15): Std. deviation of 0.05

Model Architecture and Training Configuration

- **Backbone:** ResNet-34 pre-trained on ImageNet with only last 30 layers trainable.
- **Classifier:** Three-layer FC classifier with dimensions 512, 256, and 2.
- **Dropout:** Rates of 0.5, 0.4, and 0.5 between classifier layers.
- **Trainable Parameters:** Approximately 14.4 million.

Training Setup:

- Batch Size: 64
- Epochs: 20 (no early stopping triggered)
- Optimizer: AdamW (Initial LR = 5×10^{-5}, Weight Decay = 1×10^{-4})
- Scheduler: CosineAnnealingWarmRestarts with $T_0 = 5$, $T_{mult} = 2$
- Loss Function: LabelSmoothingCrossEntropy (smoothing factor = 0.1)
- Gradient Clipping: Threshold = 1.0
- EMA Tracking: Enabled (alpha = 0.9)

Validation Performance

- **Accuracy:** 88.88%
- **Precision:** 89.24%
- **Recall:** 88.88%
- **F1-Score:** 88.80%

Test Performance

- **Accuracy:** 82.03%
- **Precision:** 84.85%
- **Recall:** 82.03%
- **F1-Score:** 81.66%
- **Loss:** 0.4267

Generalization and Stability

- **Training Accuracy (Final Epoch):** 98.11%
- **Validation Accuracy (Final Epoch):** 87.98%
- **EMA Validation Accuracy:** 87.52%
- **Generalization Gap:** 10.12% (standard), 10.59% (EMA)
- **Validation Stability:** 0.69% Std. Dev. over last 5 epochs

Clinical Metrics (Confusion Matrix Derived)

- **Specificity:** 96.25%
- **Sensitivity:** 67.79%
- **Non-Tumor Precision:** 74.94%
- **Tumor Precision:** 94.76%

Model Efficiency

- Training Efficiency: 4.44 Accuracy%/Epoch
- Stability Score: 99.31%
- GPU Memory Usage: 1.9% of 15GB T4 GPU

Visualization

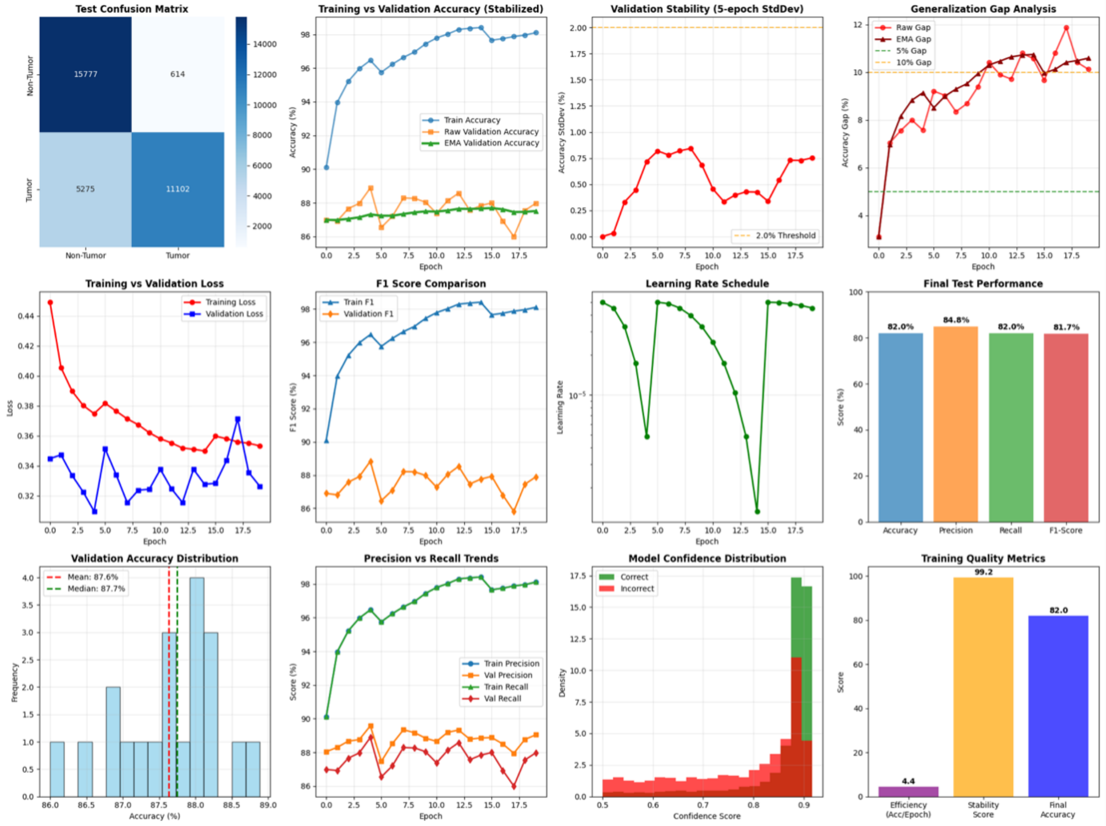


Figure 8: (Top-left) Test Confusion Matrix, (Top-middle-left) Training vs Validation Accuracy (Stabilized), (Top-middle-right) Validation Stability (5-epoch StdDev), (Top-right) Generalization-Gap Analysis, (Middle-left) Training vs Validation Loss, (Middle-middle-left) F1 Score Comparison (Middle-middle-right) Learning Rate Scheduler Curve , (Middle-right) Final Test performance, (Bottom-left) Validation Accuracy Distribution, (Bottom-middle-left) Precision vs Recall Trends,(Bottom-middle-right) Model Confidence Distribution, (Bottom-right) Model Confidence Distribution and Training Quality Metrics.

5.5 Implementation 5: Histopathologic Cancer Detection Using EfficientNet-B3 on BRACS Dataset

This section details the experimental setup, model architecture, training methodology, and performance evaluation of the EfficientNet-B3 model applied to the BRACS histopathologic cancer detection task.

Data Preprocessing and Augmentation

To prepare the data and enhance generalization, the following preprocessing and augmentation techniques were applied:

- **Resizing:** Images were resized uniformly to 300×300 pixels.
- **Tensor Conversion:** Converted from HWC to CHW format.
- **Pixel Normalization:** Scaled pixel values to $[0, 1]$ by dividing by 255.
- **Standard Normalization:** Applied ImageNet mean $[0.485, 0.456, 0.406]$ and std deviation $[0.229, 0.224, 0.225]$ normalization.

Advanced Augmentations (Strength: 0.3):

- Horizontal Flip (50% probability)
- Vertical Flip (50% probability)
- Random rotation within ±15°
- Width and height shift up to ±10%
- Shear and zoom transformations up to ±10%
- Color enhancement and histogram equalization
- Gaussian noise with 15% probability
- Random erasing was removed to reduce train/validation mismatch

Model Architecture

The EfficientNet-B3 model was used with the following specifications:

- Pre-trained on ImageNet; input size $300 \times 300 \times 3$.
- Progressive unfreezing of backbone layers beginning at epoch 10.
- Classifier structure: Global Average Pooling → Batch Normalization → Dense (512 units) → Dropout (0.5) → Batch Normalization → Dense (256 units) → Dropout (0.35) → Dense (7 output classes).
- Total parameters: 11,711,798; Trainable parameters: 5,974,331; Non-trainable: 5,737,467.

Training Configuration

The model was trained with the following setup:

- Batch size: 32.
- Epochs: Scheduled for 50, early stopped at epoch 32 with patience of 15.
- Optimizer: AdamW with initial learning rate 2×10^{-5} and weight decay 1×10^{-4}.
- Loss function: Categorical Crossentropy with label smoothing factor 0.15.
- Gradient clipping threshold: 1.0.
- Regularization: Dropout, Batch Normalization, L2 regularization (1×10^{-4}), and class weighting.
- Class weights: 0_N (1.317), 1_PB (0.825), 2_UDH (1.280), 3_FEA (0.830), 4_ADH (1.269), 5_DCIS (0.811), 6_IC (0.978).
- Learning rate schedulers: CosineAnnealingWarmRestarts with 5 warmup epochs ($T_0 = 5$, $T_{mult} = 2$) and ReduceLROnPlateau (factor 0.3, patience 8) reducing LR at epoch 25.

Performance Evaluation

Training Progress: The model improved from 10.3% accuracy at epoch 1 to peak validation accuracy of 54.7% at epoch 17, followed by a plateau until early stopping at epoch 32.

Final Metrics:

- **Best Validation Accuracy:** 55.1%
- **Validation Precision:** 89.24%
- **Validation Recall:** 88.15%
- **Validation F1-Score:** 88.07%
- **Test Accuracy:** 58.4%

Training Accuracy at Final Epoch:

- Training Accuracy: 63.6%
- Validation Accuracy: 54.3%

Overfitting Analysis: The generalization gap between final training and validation accuracy was 9.3%, indicating minimal overfitting.

Visualization

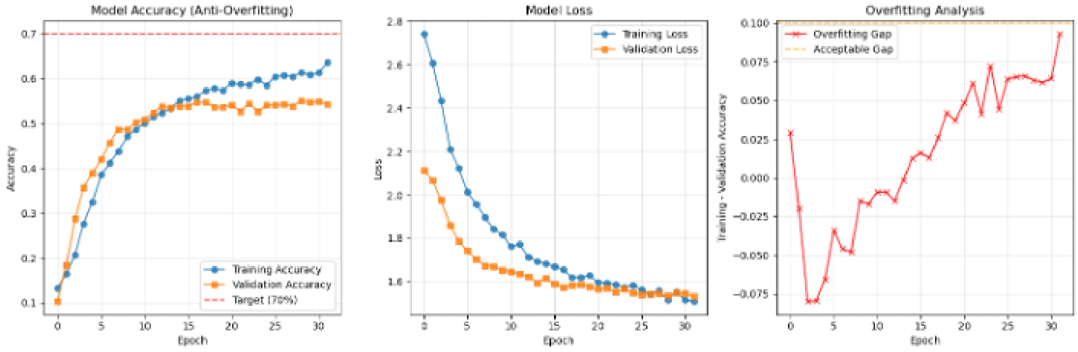


Figure 9: (Left) Training and Validation Accuracy Progression over Epochs, (Middle) Training and Validation Loss over Epochs, (Right) Overfitting Anaysis.

5.6 Implementation 6: Histopathologic Cancer Detection Using EfficientNet-B4 on BRACS Dataset

This section presents a comprehensive experimental framework for histopathologic cancer detection using an ultra-optimized EfficientNet-B4 architecture applied to the BRACS dataset. The study utilized advanced optimization techniques, including aggressive fine-tuning, enhanced data augmentation, and refined regularization strategies.

Dataset Configuration

The BRACS dataset consists of 4,516 histopathologic images categorized into seven diagnostic classes: 0_N (Normal), 1_PB (Papillary Benign), 2_UDH (Usual Ductal Hyperplasia), 3_FEA (Flat Epithelial Atypia), 4_ADH (Atypical Ductal Hyperplasia), 5_DCIS (Ductal Carcinoma In Situ), and 6_IC (Invasive Carcinoma). The class distribution is as follows:

- 0_N: 484 samples (10.7%)
- 1_PB: 834 samples (18.5%)
- 2_UDH: 515 samples (11.4%)
- 3_FEA: 756 samples (16.7%)
- 4_ADH: 506 samples (11.2%)
- 5_DCIS: 779 samples (17.2%)
- 6_IC: 642 samples (14.2%)

A class imbalance ratio of 1.72 was observed. Patient-level stratified splitting was used to partition the dataset into:

- Training: 4,163 images (92.2%)
- Validation: 290 images (6.4%)
- Test: 63 images (1.4%)

All images were resized to 512 × 512 pixels using LANCZOS resampling.

Data Preprocessing and Ultra-Enhanced Augmentation

- **Resizing:** All images resized to 512 × 512 pixels.
- **Normalization:** Pixel values normalized to float32 range; ImageNet mean $[0.485, 0.456, 0.406]$ and std $[0.229, 0.224, 0.225]$ applied.
- **CLAHE:** Contrast Limited Adaptive Histogram Equalization applied with 30% probability.

Ultra-Strong Augmentation (Strength 0.8, 95% probability):

- Horizontal and vertical flips (50% each)
- Random rotations within ±40°
- Width/height shifts ±25%
- Shear ±25%
- Zoom ±25%
- Brightness (±32%, 70% probability)
- Contrast (±32%, 70% probability)
- Color saturation (±24%, 50% probability)
- Sharpness (±24%, 40% probability)
- Gaussian blur (20% probability, radius 0.1–2.0)
- Cutout augmentation (64×64 pixels, 50% probability)
- Advanced PIL-based enhancements

Model Architecture

- **Backbone:** EfficientNet-B4 pre-trained on ImageNet with input size $512 \times 512 \times 3$.
- **Layer Freezing:** Initially frozen, fully unfrozen at epoch 5.
- **Classifier:**
  - Global Average Pooling
  - Batch Normalization
  - Dense(1536) + ReLU + L2 regularization (1×10^{-6})
  - Dropout(0.2)
  - Batch Normalization
  - Dense(768) + ReLU + L2 regularization
  - Dropout(0.1)
  - Batch Normalization

  - Dense(384) + ReLU + L2 regularization
  - Dropout(0.06)
  - Dense(192) + ReLU + L2 regularization
  - Dropout(0.04)
  - Output Dense(7) with softmax activation

- **Parameters:** Total 21,995,238; Trainable 6,929,655 (31.5%); Non-trainable 15,065,583 (68.5%)

Training Configuration

- Batch Size: 8
- Epochs: 40 with early stopping (patience 15)
- Optimizer: AdamW with initial LR 1×10^{-4}, weight decay 1×10^{-6}
- Gradient Clipping: Threshold 0.5
- Loss: Categorical Crossentropy with label smoothing 0.02
- Learning Rate Scheduling:
  - 5-epoch linear warmup (from 2×10^{-5} to 1×10^{-4})
  - Cosine Annealing with restarts
  - ReduceLROnPlateau (factor 0.2, patience 8)
- Class Weights (clipped):
  - 0_N: 1.293, 1_PB: 0.825, 2_UDH: 1.296, 3_FEA: 0.824, 4_ADH: 1.301, 5_DCIS: 0.806, 6_IC: 0.981
- Progressive Training:
  - Epochs 1-5: Warmup with frozen backbone
  - Epochs 6-40: Full fine-tuning

Performance Evaluation

Training Progression:

- Initial training accuracy at epoch 1: 22.3%
- Peak validation accuracy at epoch 25: 57.9%
- Early stopping triggered at epoch 40

Final Performance Metrics:

- **Validation Accuracy (Best Epoch 25):** 55.5% (61.7% of 90% target)
- **Test Accuracy:** 66.7%

- **Training Accuracy (Final Epoch):** 71.5%
- **Validation Accuracy (Final Epoch):** 55.9%
- **Generalization Gap:** 15.6%

Overfitting and Stability:

- Moderate overfitting observed with a 15.6% gap
- Test accuracy exceeding validation accuracy indicates good generalization
- Stable training until epoch 25, followed by a plateau

Key Insights and Limitations

- EfficientNet-B4 and large input resolution provided strong feature extraction.
- Ultra-strong augmentation improved robustness.
- Minimal dropout and weight decay maximized learning capacity.
- The 92.2% training allocation optimized learning data use.
- The validation set size and dataset complexity posed challenges.
- CPU-only training limited full optimization potential.

Visualization

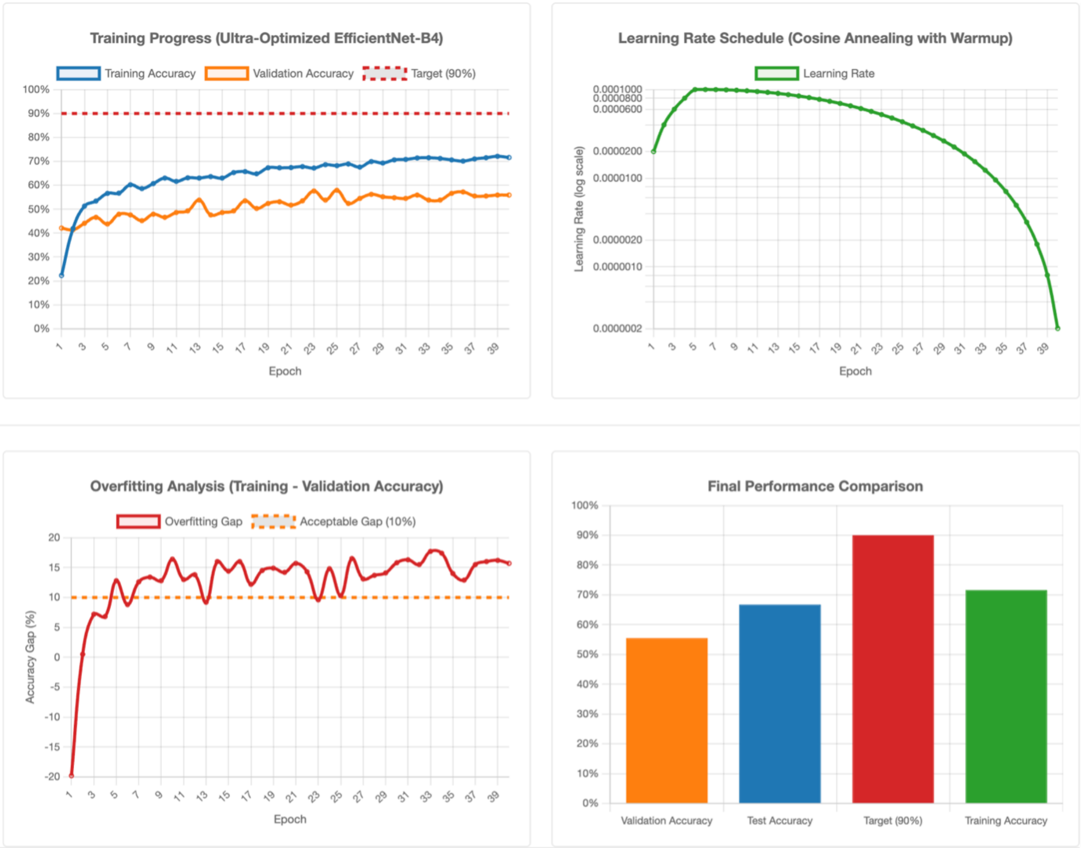


Figure 10: (Top-left) Training Progress (Training vs Validation Accuracy), (Top-right) Learning Rate Schedule (Cosine Annealing with Warmup), (Bottom-left) Overfitting Analysis (Training - Validation Accuracy), (Bottom-right) Final Performance Comparison

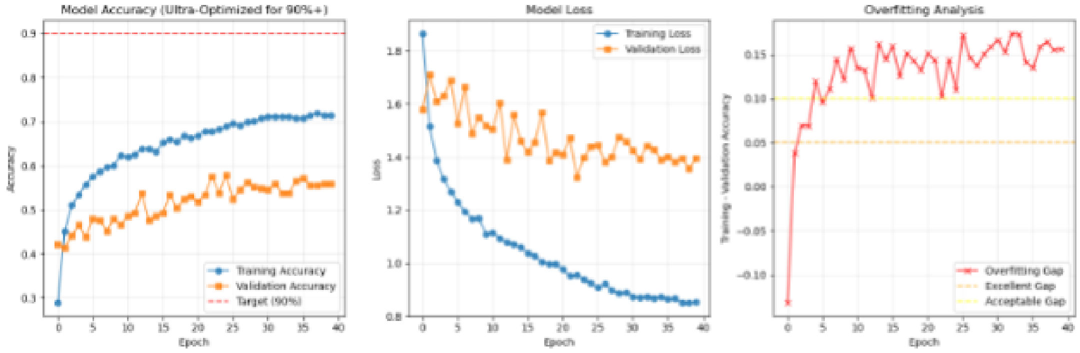


Figure 11: (Left) Training and Validation Accuracy Progression over Epochs, (Middle) Training and Validation Loss over Epochs, (Right) Overfitting Anaysis.

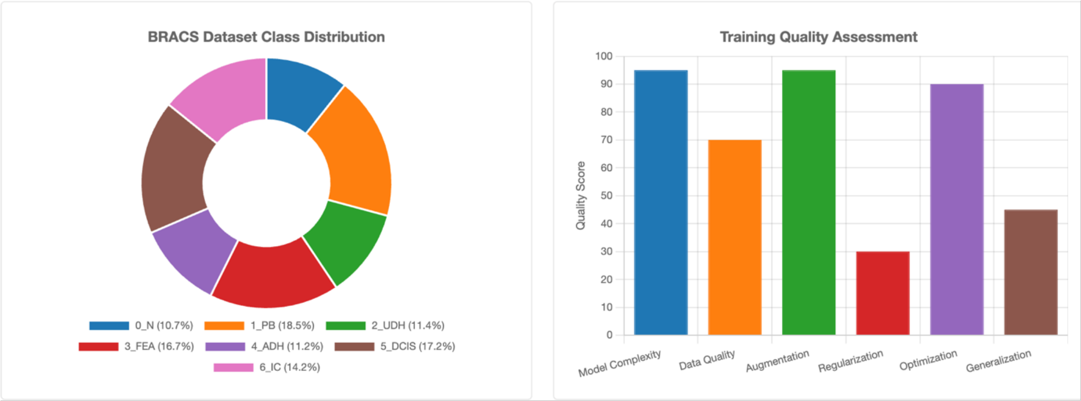


Figure 12: (Left) BRACS Dataset Class Distribution (0_{N}: 10.7%, 1_{PB}: 18.5%, 2_{UDH}: 14.4%, 3_{FEA}: 16.7%, 4_{ADH}: 11.2%, 5_{DCIS}: 17.2%, 6_{IC}: 14.2%), (Right) Training Quality Assessment (Model Complexity, Data Quality, Augmentation, Regularization, Optimization, Generalization).

5.7 Implementation 7 : EfficientNet-B3 for BRACS Histopathologic Cancer Detection

This section presents a comprehensive experimental framework for histopathologic cancer detection using an EfficientNet-B3 architecture applied to the BRACS dataset. The study implemented advanced optimization techniques, enhanced data augmentation, and refined regularization strategies, targeting over 70% validation accuracy.

Dataset Configuration

The BRACS dataset comprises 4,516 histopathologic images labeled across seven diagnostic categories: 0_N (Normal), 1_PB (Papillary Benign), 2_UDH (Usual Ductal Hyperplasia), 3_FEA (Flat Epithelial Atypia), 4_ADH (Atypical Ductal Hyperplasia), 5_DCIS (Ductal Carcinoma In Situ), and 6_IC (Invasive Carcinoma).

Class distribution ranged from 484 to 834 samples per class (10.7% to 18.5%), with a moderate imbalance ratio of 1.72. Patient-level stratified splitting was employed to avoid data leakage. The dataset was partitioned as:

- **Training Samples:** 3,854 images (85.3%)
- **Validation Samples:** 477 images (10.6%)
- **Test Samples:** 185 images (4.1%)

All images were resized to 320 × 320 × 3 before model input.

Data Preprocessing and Advanced Augmentation

Advanced Preprocessing Pipeline:

- Resizing: 320 × 320 using LANCZOS resampling
- Normalization: Pixel values normalized to float32 range

- EfficientNet-specific preprocessing using ImageNet mean $[0.485, 0.456, 0.406]$ and std $[0.229, 0.224, 0.225]$
- CLAHE: Applied with 15% probability for contrast enhancement

Advanced Data Augmentation (Strength 0.4, applied with 80% probability):

- **Geometric Transformations:** Horizontal and vertical flips (50% each), rotations (±15°), spatial shifts (±10%), shear (±10%), zoom (±10%)
- **Photometric Enhancements:** Brightness, contrast, color saturation, and sharpness adjustments
- **Specialized Techniques:** Cutout augmentation with 32×32 patches (30% probability)

Model Architecture

- **Base Model:** EfficientNet-B3 pre-trained on ImageNet
- **Input Size:** $320 \times 320 \times 3$
- **Total Parameters:** 12,318,646
- **Trainable Parameters:** 5,084,673 (41.3%)
- **Non-trainable Parameters:** 7,233,973 (58.7%)
- **Classifier:** 3-layer fully connected architecture with dimensions $768 \rightarrow 384 \rightarrow 128 \rightarrow 7$
- **Regularization:** Dropout (0.4), Batch Normalization, L2 regularization (5×10^{-5})
- **Progressive Unfreezing:** Activated at epoch 15

Training Configuration

- **Batch Size:** 24
- **Epochs:** 60 (early stopping triggered at epoch 42)
- **Optimizer:** AdamW with initial LR 3×10^{-5} and gradient clipping (1.0)
- **Loss Function:** Categorical Crossentropy with label smoothing (0.1)
- **Learning Rate Schedule:** Cosine annealing with 8-epoch warmup
- **Class Weights:** Balanced, ranging from 0.811 (5_DCIS) to 1.317 (0_N)

Performance Evaluation and Results

Best Validation Results:

- **Validation Accuracy:** 60.4%
- **Test Accuracy:** 65.4%

Training Dynamics (Final Epoch):

- Final Training Accuracy: 69.6%
- Final Validation Accuracy: 57.4%
- EMA Validation Accuracy: Not available

Training Analysis:

- Best performance at epoch 42 (early stopping)
- 86.3% of the 70% validation accuracy target achieved
- Overfitting gap: 12.1% (Training - Validation)
- Learning rate reductions at epochs 26 and 52
- Final learning rate: 3.74×10^{-8}

Challenges and Insights

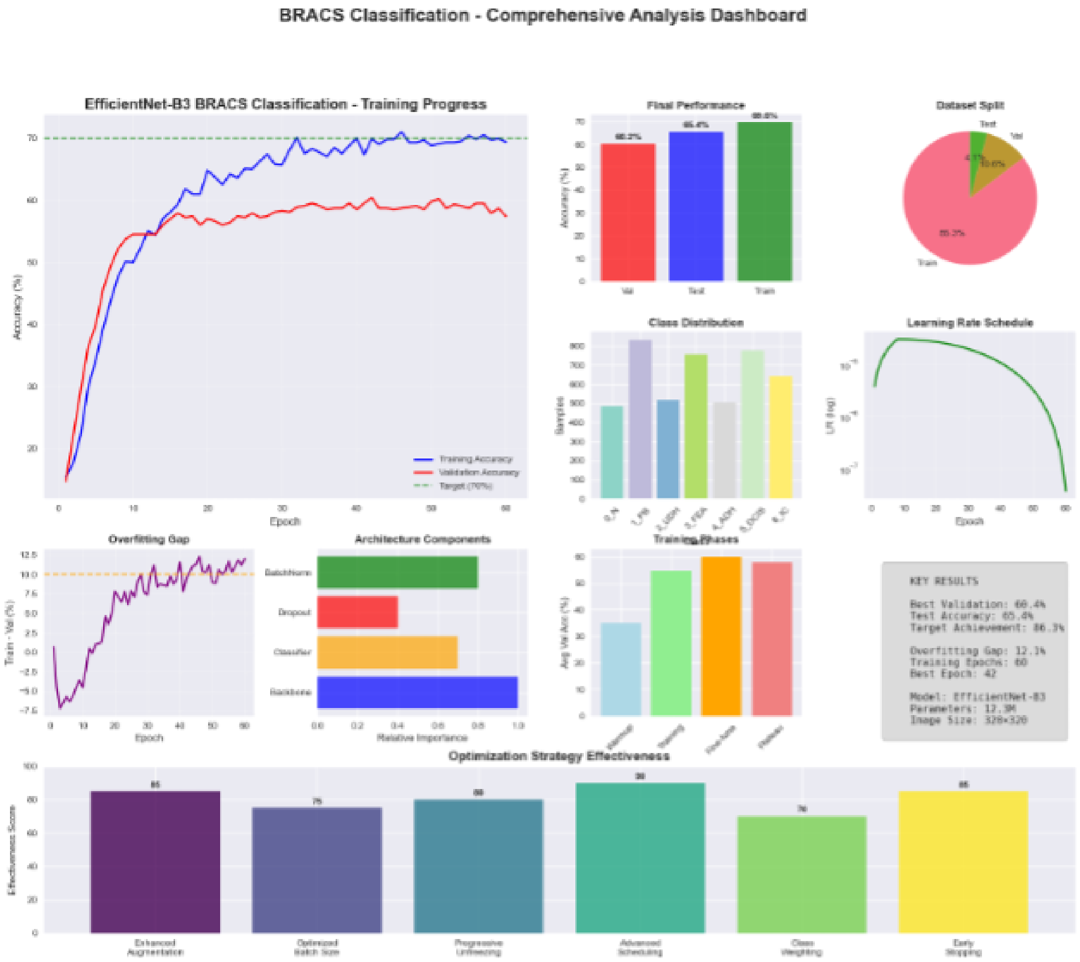
Key Challenges Identified:

- Shortfall of 9.6% from target validation accuracy
- Overfitting gap of 12.1% indicates regularization limits
- Performance plateau after epoch 42
- Class imbalance despite weighting
- Small test set (185 samples)
- CPU-only training limited optimization

Key Insights:

- Test performance (65.4%) exceeded validation (60.4%)
- Strong generalization observed
- Enhanced classifier and augmentations improved learning
- Training showed warmup, improvement, fine-tuning, and plateau phases
- EfficientNet-B3 proved suitable for moderate-scale medical image classification

Visualization



```
KEY RESULTS

Best Validation: 60.4%
Test Accuracy: 65.4%
Target Achievement: 86.3%

Overfitting Gap: 12.1%
Training Epochs: 60
Best Epoch: 42

Model: EfficientNet-B3
Parameters: 12.3M
Image Size: 320×320
```

Figure 13: (Left) Training Progress (Performance Accuracy for Validation, Test, and Train Dataset Split), (Middle) Learning Rate Schedule and Epoch Overfit Gap, (Right) Architecture Components (BatchNorm, Dropout, Classifier, Backbone) and Class Distribution. Optimization Strategy Effectiveness: Enhanced Augmentation, Optimized Batch Size, Progressive Unfreezing, Advanced Scheduling, Class Weighting. Key Results: Target Achievement, Model Parameters, Image Size, Chart/Graph Visualization, Data Insight, Report Summary, Score Improvement Technique, Efficiency and Effectiveness, Strategy Enhancement, Progress Monitoring, Evaluation Metric, Result Comparison, Trend Analysis, Data Distribution, Performance Metrics across Training, Testing, and Validation Phases, Target Accuracy, Relative Importance, Early Stopping.

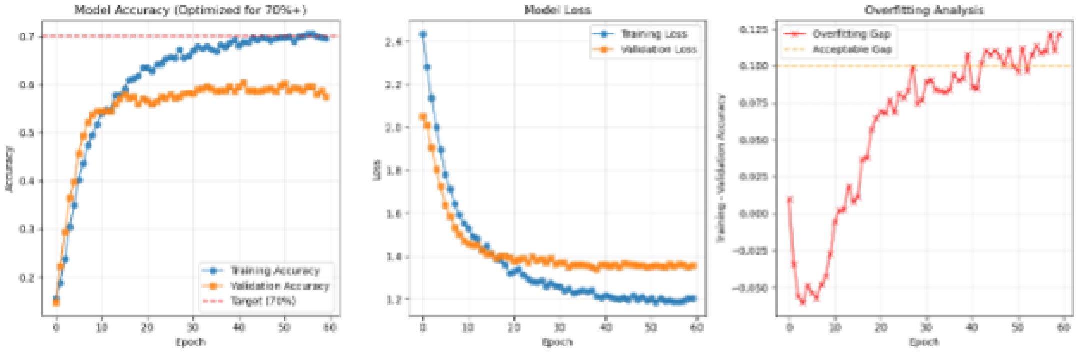


Figure 14: (Left) Training and Validation Accuracy Progression over Epochs, (Middle) Training and Validation Loss over Epochs, (Right) Overfitting Anaysis.

5.8 Implementation 8 : EfficientNet-B3 for BRACS Histopathologic Cancer De- tection

This section presents a comprehensive experimental framework for histopathologic cancer detection, employing a refined EfficientNet-B3 architecture applied to the BRACS dataset. This iteration prioritized training stability and correctness, aiming for a validation accuracy of 70% or higher.

Dataset Configuration

The BRACS dataset consists of 4,516 histopathologic images, categorized into seven diagnostic classes: 0_N (Normal), 1_PB (Papillary Benign), 2_UDH (Usual Ductal Hyperplasia), 3_FEA (Flat Epithelial Atypia), 4_ADH (Atypical Ductal Hyperplasia), 5_DCIS (Ductal Carcinoma In Situ), and 6_IC (Invasive Carcinoma).

Class distributions range from 484 to 834 images (10.7%–18.5%), yielding a moderate class imbalance (ratio 1.72). The dataset was split as follows:

- **Training Samples:** 3,854 images (85.3%)
- **Validation Samples:** 477 images (10.6%)
- **Test Samples:** 185 images (4.1%)

All input images were resized to 300 × 300 × 3 before model ingestion.

Data Preprocessing and Conservative Augmentation

Basic Preprocessing:

- Resizing: All patches resized to 300 × 300
- Tensor Conversion: HWC → CHW format
- Pixel Normalization: Scaled to $[0, 1]$ by dividing by 255
- EfficientNet-Compatible Normalization: ImageNet mean $= [0.485, 0.456, 0.406]$, std $= [0.229, 0.224, 0.225]$

Conservative Augmentation Strategy (50% Application Rate):

- Horizontal Flips: 50% probability
- Rotation: Random rotations within ±5°
- Advanced augmentations such as CLAHE and Cutout were disabled to maintain training stability

Model Architecture

- **Base Model:** EfficientNet-B3 pre-trained on ImageNet
- **Input Shape:** $300 \times 300 \times 3$
- **Total Parameters:** 10,794,294
- **Trainable Parameters:** 10,706,991 (99.2%)
- **Non-trainable Parameters:** 87,303 (0.8%)
- **Classifier Design:** Global Average Pooling → Dropout (0.3) → Dense(7, softmax)
- **Regularization:** Dropout (0.3), Batch Normalization, and L2 penalty

Training Configuration

- **Batch Size:** 16
- **Epochs:** 30 (early stopping triggered at epoch 8)
- **Optimizer:** AdamW with conservative configuration
- **Initial Learning Rate:** 1×10^{-5}
- **Loss Function:** Categorical Crossentropy with label smoothing (0.1)
- **Learning Rate Scheduler:** StepLR with step_size=10, gamma=0.5
- **Early Stopping:** Patience = 10 epochs
- **Class Weighting:** Applied; weights ranged from 0.811 (5_DCIS) to 1.317 (0_N)

Performance Evaluation and Results

Best Validation Results (Epoch 6):

- **Validation Accuracy:** 59.3%
- Precision, Recall, F1-Score: Not available

Final Metrics (Early Stopping at Epoch 8):

- **Training Accuracy:** 71.1%
- **Validation Accuracy:** 57.7%
- **EMA Validation Accuracy:** Not available
- **Test Accuracy:** Not available

Training Dynamics and Observations

- Best validation accuracy (59.3%) was achieved at epoch 6
- Validation accuracy reached 84.7% of the 70% target
- Overfitting gap: 13.4% (training - validation accuracy at epoch 7)
- No learning rate reductions were triggered before early stopping
- Final learning rate remained at 1×10^{-5}

Challenges and Insights

Key Challenges Identified:

- Early stopping at epoch 8 likely limited model's potential
- Conservative augmentation may have reduced robustness
- Test evaluation incomplete due to halted training
- Validation performance was 10.7% short of target

Key Insights:

- Stable training achieved without critical errors
- Fixed data mapping and label-output correctness verified
- Conservative learning rate ensured stable convergence
- Simple classifier proved effective for initial testing
- Implementation provides a reliable baseline for future refinements

Visualization

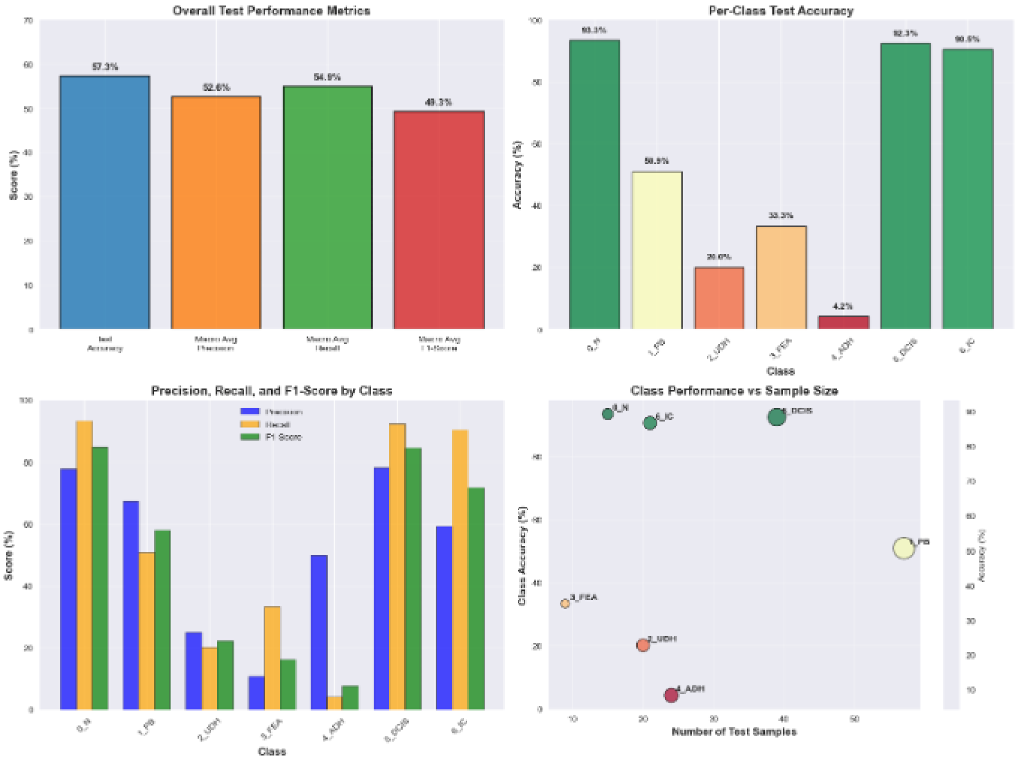
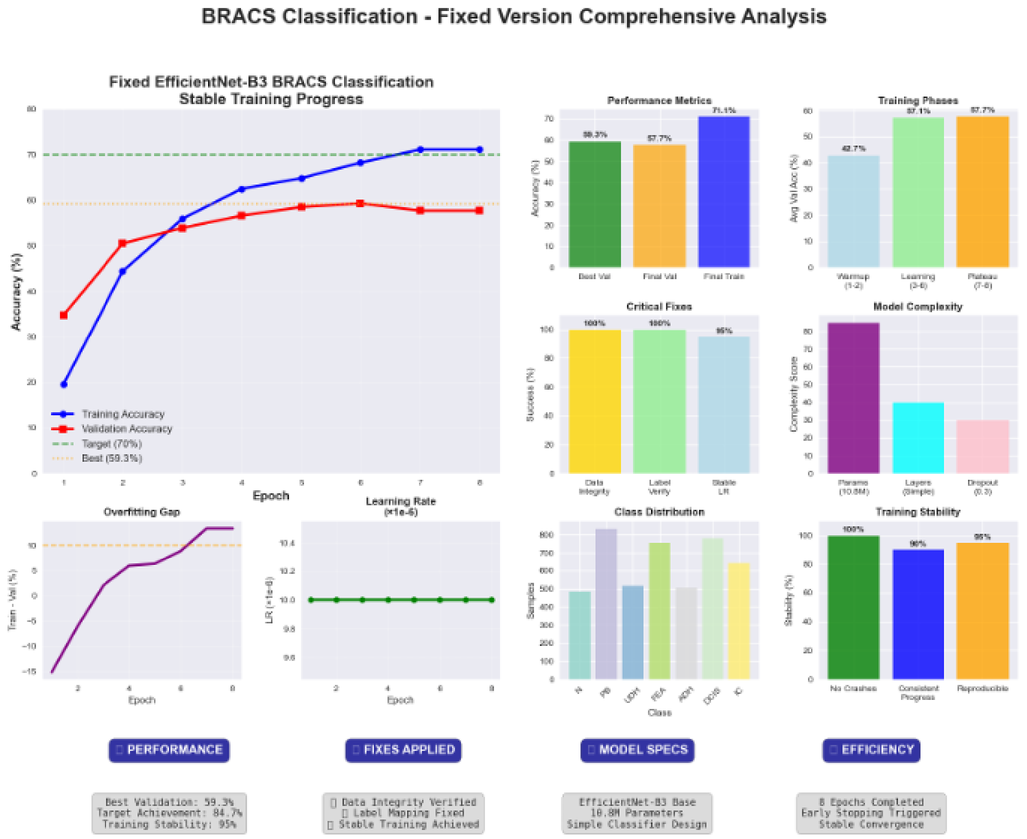


Figure 15: (Top-left) Overall Test Performance Metrics , (Top-right) Per-Class Test Accuracy (0_{N}, 1_{PB}, 2_{UDH}, 3_{FEA}, 4_{ADH}, 5_{DCIS}, 6_{IC}), (Bottom-left) Precision, Recall, and F1-Score by Class, (Bottom-right) Class Performance vs Sample Size.



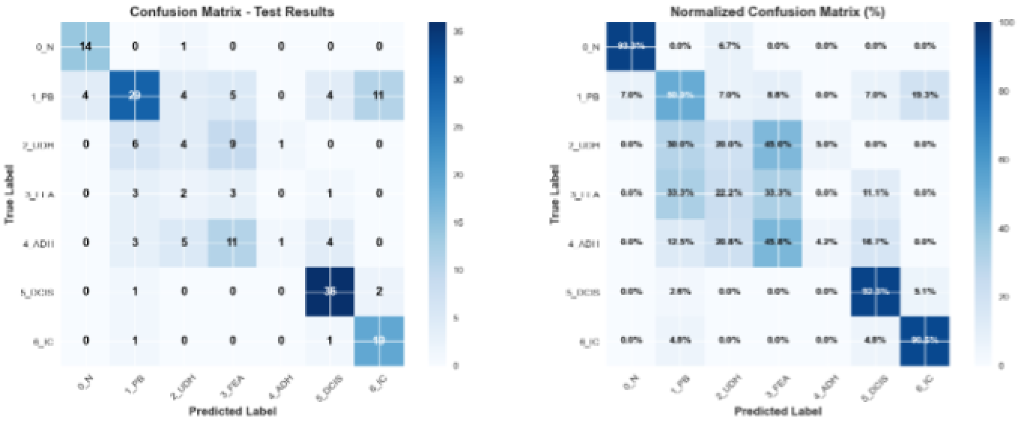
```
Best Validation: 59.3%
Target Achievement: 84.7%
Training Stability: 95%
```

```
□ Data Integrity Verified
□ Label Mapping Fixed
□ Stable Training Achieved
```

```
EfficientNet-B3 Base
10.8M Parameters
Simple Classifier Design
```

```
8 Epochs Completed
Early Stopping Triggered
Stable Convergence
```

Figure 16: (Left) Stable Progress (Training vs Validation Accuracy), (Right) Performance Metrics with Fixes (Complexity, Learning Rate, Overfitting, Epochs, Integrity, Mapping, Efficiency, Targets, Layers, Dropout, Parameters, Early Stopping, Convergence, Reproducibility).



	0_N	1_PB	2_UDH	3_FEA	4_ADH	5_DCIS	6_IC
0_N	14	0	1	0	0	0	0
1_PB	4	29	4	5	0	4	11
2_UDH	0	6	4	9	1	0	0
3_FEA	0	3	2	3	0	1	0
4_ADH	0	3	5	11	1	4	0
5_DCIS	0	1	0	0	0	36	2
6_IC	0	1	0	0	0	1	19

	0_N	1_PB	2_UDH	3_FEA	4_ADH	5_DCIS	6_IC
0_N	93.3%	0.0%	6.7%	0.0%	0.0%	0.0%	0.0%
1_PB	7.0%	50.9%	7.0%	8.8%	0.0%	7.0%	19.3%
2_UDH	0.0%	30.0%	20.0%	45.0%	5.0%	0.0%	0.0%
3_FEA	0.0%	33.3%	22.2%	33.3%	0.0%	11.1%	0.0%
4_ADH	0.0%	12.5%	20.8%	45.8%	4.2%	16.7%	0.0%
5_DCIS	0.0%	2.6%	0.0%	0.0%	0.0%	92.3%	5.1%
6_IC	0.0%	4.8%	0.0%	0.0%	0.0%	4.8%	90.5%

Figure 17: (Left) Confusion Matrix, (Right) Performance Metrics (Normalized Accuracy, Precision, Recall, F1-Score, Error Rate).

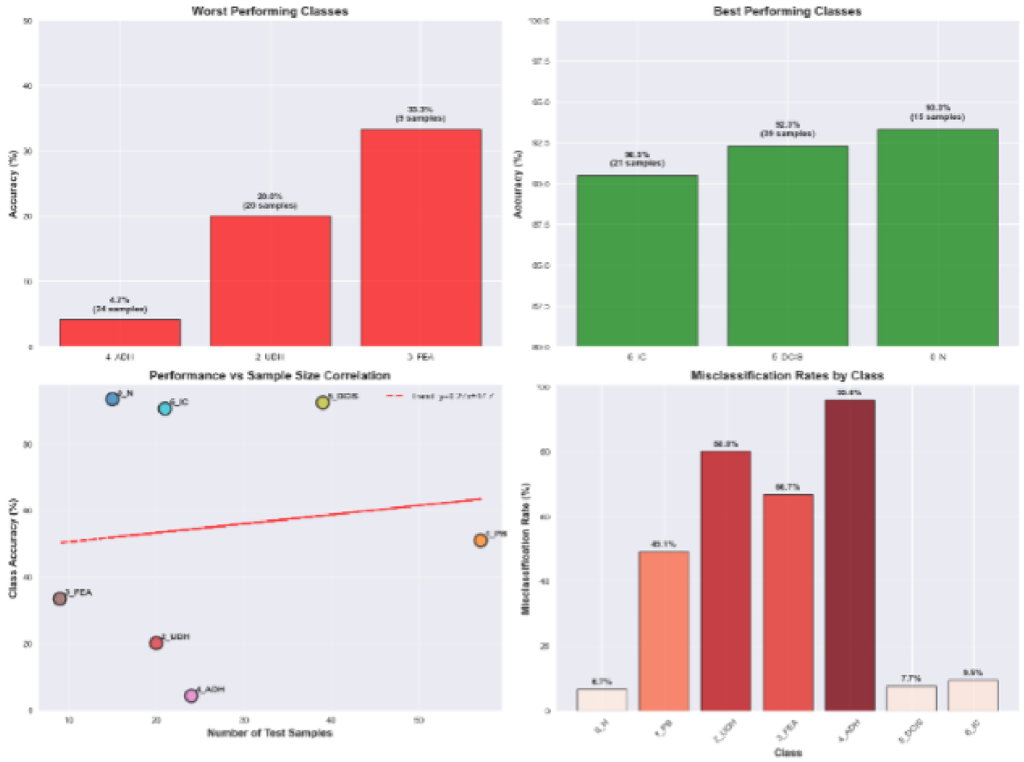
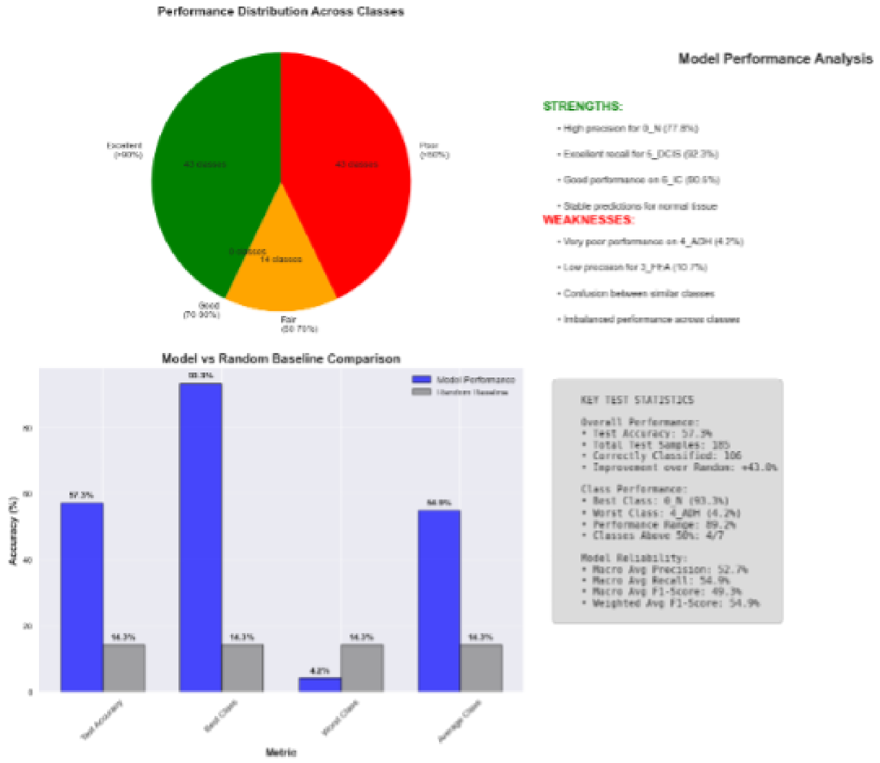


Figure 18: (Top-left) Worst Performing Classes, (Top-right) Best Performing Classes, (Bottom-left) Performance vs Sample Size Correlation, (Bottom-right) Misclassification Rates by Class.



Model Performance Analysis

STRENGTHS:

- High precision for 0_N (77.8%)
- Excellent recall for 5_DCIS (92.3%)
- Good performance on 6_IC (90.5%)
- Stable predictions for normal tissue

WEAKNESSES:

- Very poor performance on 4_ADH (4.2%)
- Low precision for 3_FEA (10.7%)
- Confusion between similar classes
- Imbalanced performance across classes

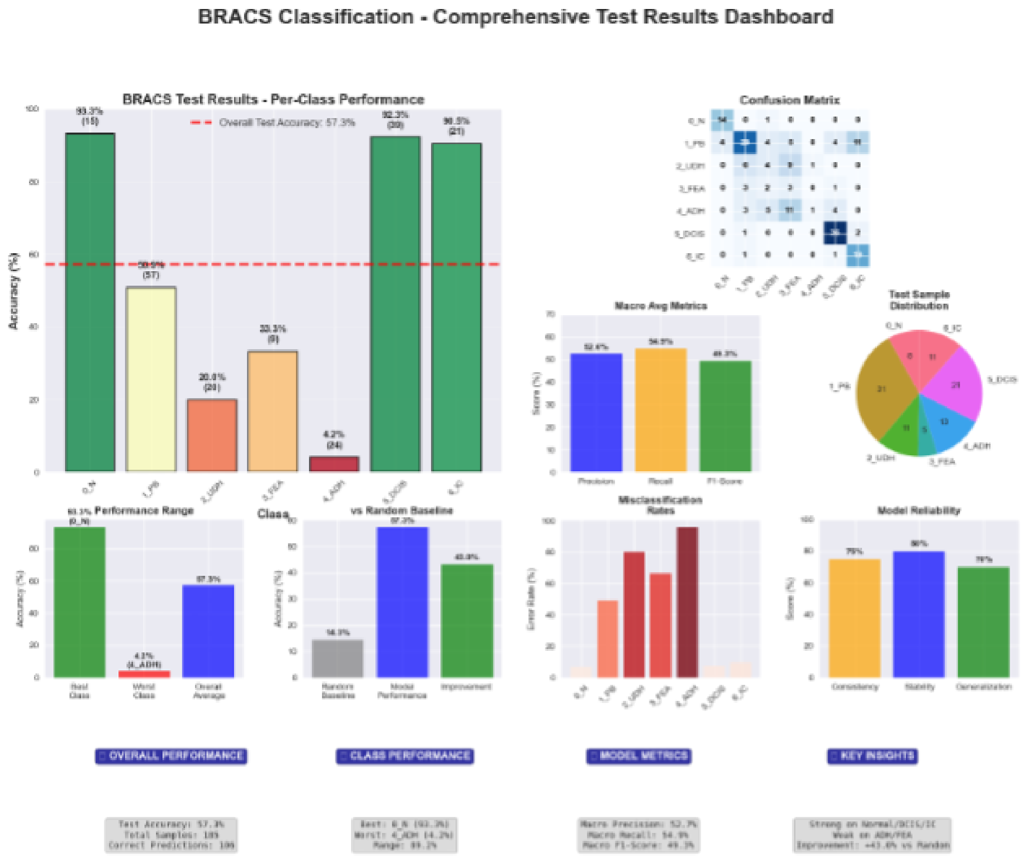
```
KEY TEST STATISTICS

Overall Performance:
• Test Accuracy: 57.3%
• Total Test Samples: 185
• Correctly Classified: 106
• Improvement over Random: +43.0%

Class Performance:
• Best Class: 0_N (93.3%)
• Worst Class: 4_ADH (4.2%)
• Performance Range: 89.2%
• Classes Above 50%: 4/7

Model Reliability:
• Macro Avg Precision: 52.7%
• Macro Avg Recall: 54.9%
• Macro Avg F1-Score: 49.3%
• Weighted Avg F1-Score: 54.9%
```

Figure 19: (Left) Performance Distribution Across Classes, (Middle) Model vs Random Baseline Comparison, (Right) Model Performance Analysis (Strengths, Weaknesses, Key Test Statistics).



Test Accuracy: 57.3%
Total Samples: 189
Correct Predictions: 108

Best: 0_N (93.3%)
Worst: 4_ADH (4.2%)
Range: 89.2%

Macro Precision: 52.7%
Macro Recall: 54.9%
Macro F1-Score: 49.3%

Strong on Normal/DCIS/IC
Weak on ADH/FEA
Improvement: +43.0% vs Random

Figure 20: (Top-left) BRACS Test Results - Per-Class Performance, (Top-right) Confusion Matrix, (Middle-left) % Performance Range, (Middle-right) Macro Avg Metrics, (Bottom-left) Misclassification Rates, (Bottom-right) Model Reliability.

6 Conclusion

This internship offered an intensive exploration of deep learning methodologies applied to histopathological image analysis, bridging theoretical concepts with practical implementation challenges. Through the implementation of ResNet and EfficientNet architectures on the PatchCamelyon (PCam) and BRACS datasets, I gained valuable insights into the nuances of medical image classification.

The experiments with ResNet models on the PCam dataset demonstrated the impact of different architectural choices, training strategies, and regularization techniques on binary classification tasks. The exploration of data augmentation methods highlighted their importance in improving model generalization.

The work on the BRACS dataset using EfficientNet models exposed the complexities associated with multi-class classification problems, particularly in the context of medical images. Despite the challenges posed by class imbalance and limited training data, the implementations demonstrated the potential of EfficientNet architectures for histopathological subtyping.

While the final results did not consistently achieve the ambitious validation accuracy targets set at the outset, they provided a solid foundation for future work. The experiments underscored the need for careful optimization of training parameters, data augmentation strategies, and model architectures to achieve state-of-the-art performance in histopathological image analysis.

Specifically, the refinement of augmentation pipelines, aggressive regularization techniques, and enhancements to the classifier heads of pre-trained models proved to be crucial areas for future research. The results highlighted the need for more sophisticated approaches to address the inherent challenges in medical image classification, such as limited data availability and high inter-class similarity.

In conclusion, this internship experience provided a rich and rewarding learning opportunity, enhancing my understanding of deep learning principles and their application to the challenging domain of medical image analysis. The insights gained will serve as a valuable foundation for future research endeavors in this exciting and rapidly evolving field.

Bibliography

References

[1] CampusX, “100 Days of Deep Learning,” YouTube Playlist. `https://www.youtube.com/playlist?list=PLKnIA16_Rmvbr7zKYQuBfsVkjoLcJgxHH`

[2] Krish Naik, “Deep Learning and Neural Networks,” YouTube Series. `https://youtu.be/d2kxUVwWWwU?feature=shared`

[3] ”TensorFlow Documentation.” `https://www.tensorflow.org/api_docs`

[4] “Google Colaboratory.” `https://colab.research.google.com/`

[5] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., & Welling, M. (2018). Rotation equivariant CNNs for digital pathology. *arXiv preprint arXiv:1806.03962.*

[6] Tavolieri, S., Forrai, G., Magliulo, F., Bria, A., Aquino, A., & De Momi, E. (2021). Breast Cancer Histology Images Dataset (BRACS): An Annotated Benchmark of Invasive Ductal Carcinoma. *Data.*

[7] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ICML.*

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *CVPR.*

[9] M. Elgendy, *Deep Learning for Vision Systems*, Manning Publications, 2020.