MANIPAL
ACADEMY of HIGHER EDUCATION
(Institution of Eminence Deemed to be University)

MANIPAL SCHOOL OF INFORMATION SCIENCES

(A Constituent unit of MAHE, Manipal)

Development of Deep Learning Approach for Grading Squamous Cell Carcinoma from Histopathology Images

Reg. Number	Name	Branch
251100610012	Adithya Rao Kalathur	M.E. Computer Science and Engineering
251100610013	Vaishnav P S	M.E. Computer Science and Engineering
251100610017	Sriram K	M.E. Computer Science and Engineering
251100610040	Vaibhav V Acharya	M.E. Computer Science and Engineering

Under the guidance of

Dr Keerthana Prasad
Professor & Director,
Manipal School of Information Sciences,
MAHE, MANIPAL

17/11/2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
SCC	Squamous Cell Carcinoma
CNN	Convolutional Neural Network
AI	Artificial Intelligence
GPU	Graphics Processing Unit
WSI	Whole Slide Image
RGB	Red Green Blue
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
API	Application Programming Interface
UI	User Interface
GUI	Graphical User Interface
PDF	Portable Document Format
CAM	Class Activation Mapping
Grad-CAM	Gradient-weighted Class Activation Mapping
LC25000	Lung and Colon Cancer Dataset (25,000 images)
L1	L1 Regularization (Lasso)
L2	L2 Regularization (Ridge)
LR	Learning Rate
NSCLC	Non-Small Cell Lung Cancer
ECOC	Error-Correcting Output Codes
SVM	Support Vector Machine
ViT	Vision Transformer
MCC	Matthews Correlation Coefficient
KD	Knowledge Distillation
MLP	Multi-Layer Perceptron
ISIC	International Skin Imaging Collaboration
BUSI	Breast Ultrasound Images
CIFAR	Canadian Institute for Advanced Research
ImageNet	Large Visual Database for Visual Object Recognition
AdamW	Adam with Weight Decay
RMSprop	Root Mean Square Propagation
CLAHE	Contrast Limited Adaptive Histogram Equalization
PIL	Python Imaging Library
JPEG	Joint Photographic Experts Group
WSIs	Whole Slide Images
GAP	Global Average Pooling

NaN	Not a Number
RGBA	Red Green Blue Alpha

1. INTRODUCTION

1.1. Background

Squamous Cell Carcinoma (SCC) is an abnormal tissue growth that develops from squamous epithelial cells. These cells are flat, thin structures that cover the surface of the skin, mouth, throat, and several internal organs. When these cells begin to grow in an uncontrolled manner, they form irregular structures that disturb normal tissue patterns. Grading this abnormal growth is important because it helps in understanding the severity of the condition and supports doctors in deciding the most suitable treatment approach.

1.2. Importance of Histopathology

Histopathology, which is the microscopic study of tissue samples, is the most trusted and widely used method for identifying and grading Squamous Cell Carcinoma. A histopathology image shows the arrangement, shape, and texture of the cells. Based on how similar or different the cells appear compared to normal tissue, the sample is graded as well-differentiated, moderately differentiated, or poorly differentiated. However, manual grading through microscopic observation is time-consuming and depends greatly on the experience and judgment of the pathologist. This subjectivity can sometimes lead to variations in diagnosis.

1.3. Challenges in Manual Grading

Manual grading presents several difficulties in real-world conditions. It requires high expertise, consistent concentration, and considerable time to examine each tissue slide carefully. Differences in staining, lighting, or image clarity can make grading even more difficult. In hospitals and diagnostic centers that handle a large number of samples daily, these challenges create delays and increase the workload for medical professionals. Therefore, there is a strong need for automated systems that can perform grading quickly, consistently, and objectively.

1.4. Advancements in Digital Pathology

The introduction of digital pathology has brought major changes to medical image analysis. Modern scanners can convert traditional glass slides into high-resolution digital images. These digital histopathology images can be easily stored, shared, and processed by computer systems. This development has opened new opportunities for automated image analysis using computer-based techniques, allowing for faster and more reliable examination of tissue structures.

1.5. Emergence of Deep Learning

Deep learning, a specialized area within artificial intelligence, has shown remarkable success in analyzing and interpreting complex image data. It uses neural networks with multiple layers that automatically learn useful patterns from raw images. Convolutional Neural Networks (CNNs), in particular, are highly effective for image-related tasks. They can recognize detailed visual patterns such as edges, shapes, and textures, which are essential for understanding histopathology images. Unlike traditional image processing techniques that rely on manually designed features, deep learning models can learn features directly from the data itself.

1.6. Deep Learning for Grading Squamous Cell Carcinoma

Using deep learning methods for grading Squamous Cell Carcinoma can significantly improve the reliability and efficiency of the diagnostic process. A CNN model can be trained on labeled histopathology images to identify patterns that correspond to different grades of SCC. Once trained, the model can classify new images accurately without human intervention. This approach reduces errors, provides consistency, and saves valuable time for pathologists. It also supports better decision-making by combining computational accuracy with medical expertise. The development of a deep learning approach for grading Squamous Cell Carcinoma from histopathology images represents an important step toward intelligent and automated medical diagnosis. It strengthens the role of technology in assisting experts, enhances the precision of image-based evaluation, and contributes to making medical analysis faster, more objective, and more dependable.

2. OBJECTIVES

The main objective of this project is to develop, train, and evaluate a set of deep learning models for grading Squamous Cell Carcinoma (SCC) from histopathology images. The work focuses on exploring different model architectures and fine-tuning their parameters to achieve better accuracy, reliability, and consistency.

1. To collect and preprocess histopathology images of Squamous Cell Carcinoma for training and testing the models. This includes resizing images, normalizing pixel values, and applying augmentation techniques to improve image variety and prevent model overfitting.
2. To design and implement multiple deep learning architectures, including basic Convolutional Neural Networks (CNNs) and transfer learning models such as VGG, ResNet, and Inception. Each model will be trained and tested to study how its structure affects performance.
3. To experiment with different parameters and hyperparameter tuning techniques, such as adjusting the learning rate, batch size, number of epochs, optimizers, and activation functions, to observe how these changes influence model learning and accuracy.
4. To train, validate, and test each model on the prepared dataset and analyze how different hyperparameter settings impact performance, convergence, and stability.
5. To compare the performance of all trained models using evaluation metrics like accuracy, precision, recall, and F1-score, and identify the model configuration that gives the best and most consistent grading results.

3. LITERATURE SURVEY

Recent research in cancer histopathology classification has shown significant advances through deep learning, hybrid architectures, and knowledge-distillation techniques. Early work by Musulin et al. [1] applied CNN-based multiclass grading for oral squamous cell carcinoma, demonstrating that convolutional features effectively capture morphological variations needed for cancer grading. Lung carcinoma studies expanded this direction—Patharia and Sethy [2] combined EfficientNet-B0 with SVM to build a lightweight yet accurate grading system, while Sethy et al. [3] merged wavelet texture features with AlexNet, improving robustness across lung histopathological variations. EfficientNet-based approaches continue to excel: Kumar and Nelson [4] showed that EfficientNet-B3 captures fine structural patterns in oral SCC, and Lathakumari et al. [8] demonstrated EfficientNet-B2's strong performance for NSCLC despite dataset limitations. Fusion-based methods, such as those by Kumar and Kumar [5], further reveal that combining multiple CNNs enhances the discriminative power and improves generalization on challenging SCC datasets.

More advanced architectures leverage global attention mechanisms. Kang et al. [6] proposed EsccNet, a hybrid CNN-Transformer model that achieved strong performance on whole slide images, while Park et al. [7] demonstrated that Vision Transformers outperform CNNs in margin classification for SCC, achieving high accuracy and AUC by capturing long-range tissue dependencies. Beyond SCC, Akella et al. [12] achieved near-perfect colorectal cancer detection by integrating CNN ensembles, Transformers, SVMs, and unsupervised segmentation, improving both accuracy and interpretability.

Methodological innovations have also strengthened model efficiency and reliability. Multi-teacher knowledge distillation approaches from Yang et al. [9], coded KD from Salamah et al. [10], and lightweight student architectures from Song et al. [11] demonstrate how performance can be improved while drastically reducing computational demands—critical for point-of-care deployment. Survey works [13] reinforce that ensembles and hybrid fusion strategies consistently improve accuracy and robustness across histopathology tasks. Foundational contributions such as EfficientNet's compound scaling [14] and Hinton et al.'s original KD framework [15] provide the theoretical foundation for many recent medical imaging models. Collectively, these studies highlight that combining efficient CNNs, Transformer-based

attention, ensemble fusion, and advanced distillation techniques leads to improved grading accuracy, interpretability, and clinical reliability across diverse histopathology datasets.

SL NO	Paper Title	Authors	Year	Dataset	Methods / Models Used	Detailed Key Findings / Results
1	Automated Grading of Oral Squamous Cell Carcinoma into Multiple Classes Using Deep Learning Methods	J. Musulin et al.	2021	Oral SCC histopathology images	CNN-based multiclass grading	Demonstrated successful multiclass grading (well, moderate, poor). The CNN-based pipeline achieved high accuracy and strong F1-scores and showed that deep learning models can reliably distinguish subtle morphological differences in SCC tissue. This work established a foundation for automated grading workflows.[1]
2	LungCarcinoGrade-EffNetSVM: A Novel Approach to Lung Carcinoma Grading using EfficientNet-B0 and SVM	P. Patharia, P. K. Sethy	2024	Kaggle lung CT dataset (1000 images)	EfficientNet-B0 feature extractor + SVM (ECOC)	EfficientNet-B0 was used as a compact but powerful feature extractor, and SVM improved decision boundaries, achieving 86.88% accuracy. Results showed high specificity and strong agreement metrics (MCC, Kappa), proving that hybrid DL + classical ML can achieve competitive grading performance with reduced computation time (~9s per image). [2]
3	Lung cancer histopathological image classification using wavelets and AlexNet	P. K. Sethy et al.	2023	Lung carcinoma histopathology	Wavelet features + AlexNet + SVM	Combined wavelet texture features with deep CNN embeddings to enhance feature richness. Achieved ~88.5% accuracy, demonstrating that hybrid spatial-frequency features improve discrimination across carcinoma subtypes. Demonstrated robustness across multiple histopathological variations. [3]
4	Enhancing Oral SCC Detection using EfficientNet-B3 from	A. Kumar, L. Nelson	2025	Oral SCC histopathology	EfficientNet-B3	EfficientNet-B3 effectively captured fine structural details in tissue morphology. Provided higher accuracy and stability compared to baseline CNNs due to compound scaling. Showed strong capability for detecting

	Histopathologic Images					early-stage SCC patterns, proving suitability for clinical screening pipelines. [4]
5	Histopathological Image Based Oral SCC Classification Using Deep Network Fusion	A. Kumar, V. Kumar	2023	Oral SCC histopathology	Deep network fusion of multiple CNNs	Introduced fusion of multiple CNN architectures to aggregate complementary features. Improved classification accuracy over individual models, especially for borderline cases. Demonstrated that feature-level and decision-level fusion reduces misclassification and improves generalization on challenging histopathology datasets. [5]
6	EsccNet: Hybrid CNN + Transformers Model for Whole Slide Images of Esophageal SCC	Z. Kang et al.	2024	Esophageal SCC WSIs	Hybrid CNN + Vision Transformer	Introduced a hybrid model combining CNN local feature extraction with Transformer global attention to handle gigapixel WSIs. Achieved superior accuracy over traditional CNNs, especially in capturing long-range contextual patterns. Demonstrated strong robustness across large-scale pathology slides. [6]
7	Squamous Cell Carcinoma Margin Classification Using Vision Transformers	S.-Y. Park et al.	2025	828 SCC margin images	ViT-B16/ViT-B32/ViT-L32	Vision Transformers outperformed CNN models due to their ability to process global contextual cues. ViT-B16 achieved ~0.906 accuracy and ~0.905 AUC. Showed excellent boundary-margin understanding, proving ViTs' reliability for margin assessment in surgical pathology. [7]
8	Classification of Non-Small Cell Lung Cancer using EfficientNet-B2	K. R. Lathakumari et al.	2023	Lung CT dataset (1000 images)	EfficientNet-B2	Achieved strong training accuracy (95%) but lower testing accuracy (~83%), highlighting the challenges of limited datasets and domain variability. Demonstrated EfficientNet-B2's ability to learn multi-class lung cancer patterns efficiently, though generalization requires more diverse data. [8]
9	Multi-Teacher Knowledge Distillation with Reinforcemen	C. Yang et al.	2025	Standard vision datasets	Multi-teacher KD +	Introduced a reinforcement-learning agent to dynamically assign weights to multiple teacher models. Significantly improved student performance compared to standard KD.

	t Learning for Visual Recognition				reinforcement learning	Demonstrated robustness against noisy labels, model bias, and heterogeneous teacher architectures. [9]
10	Coded Knowledge Distillation with Adaptive JPEG Compression	A. H. Salamah et al.	2025	CIFAR-10/100, ImageNet	Coded KD + adaptive JPEG encoding	Proposed compressing teacher outputs via adaptive JPEG to remove overconfident logits. Softened probability distributions improved student accuracy and reduced overfitting. Achieved consistent improvements across multiple benchmarks at low computational cost. [10]
11	Multiple Teachers Are Beneficial: Lightweight & Noise-Resistant Student Model for Point-of-Care Imaging	Y. Song et al.	2025	ISIC, BUSI, Dermnet	Multi-teacher KD + Shift-MLP student	Designed a lightweight student model for point-of-care applications. The model achieved large reductions in parameters and computation (up to 38× fewer parameters). Multi-teacher distillation enabled strong performance even in noisy clinical imaging environments. [11]
12	Colorectal Cancer Detection using Hybrid Supervised + Unsupervised Learning	S. Akella et al.	2025	CVC-ClinicDB (1,650 images)	CNN ensemble + Transformer + SVM + K-means segmentation	Achieved ~99% accuracy and AUC ≈ 0.99. Combined CNN and Transformer features with SVM classification to enhance reliability. Added unsupervised K-means segmentation for lesion localization, improving interpretability and clinical usability. [12]
13	Deep Network Fusion / Ensemble Methods for Histopathology Classification (Survey)	Various	2022 – 2024	Multiple histopathology datasets	CNN + Transformer Ensembles	Surveyed and validated that fusing CNNs and Transformers improves robustness, reduces variance, enhances generalization, and performs better on noisy, heterogeneous pathology datasets. [13]
14	EfficientNet: Rethinking Model Scaling for CNNs	M. Tan, Q. Le	2019	ImageNet	EfficientNet family (compound scaling)	Introduced compound scaling, enabling high accuracy with significantly fewer parameters. Highly influential in medical imaging due to efficiency and

						strong feature extraction capabilities. [14]
15	Distilling the Knowledge in a Neural Network	G. Hinton et al.	2015	ImageNet / general	Knowledge Distillation	Introduced the foundational concept of teacher-student distillation. Soft targets improved generalization and compression. Widely adopted in medical imaging where lightweight models are essential. [15]

4. SPECIFICATIONS

4.1 Dataset

Dataset 1: LC25000 (Lung and Colon Cancer Histopathological Images)

- Source:Kaggle(https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images)
- Classes Used (for preliminary experiments):
  - Adenocarcinoma
  - Squamous Cell Carcinoma (Binary classification)
- Image Count: 25,000 patches across 5 categories (subset used for binary task)
- Preprocessing:
  - Resized to 224 × 224 pixels
  - Pixel normalization to [0, 1]
  - Data augmentation: rotation, flipping, zoom, brightness/contrast jitter
  - Removal of noisy or blank patches

Dataset 2: Institutional SCC Grading Dataset

(Provided by the College/Institute for primary research work)

- Classes:
  - Well-differentiated SCC
  - Moderately differentiated SCC
  - Poorly differentiated SCC (Three-class grading task)
- Dataset Size:
  - Number of images per class based on institutional dataset distribution
  - Patch extraction used if Whole Slide Images (WSIs) were provided
- Preprocessing:
  - Patch generation from WSIs (if applicable)
  - Removal of white/blank/uninformative patches
  - Resized all patches/images to 224 × 224 pixels
  - Augmentation to handle class imbalance and tissue variability:
    - Random rotation, horizontal/vertical flips
    - Zoom, shift, random crop

4.2 Hardware and Software Setup

Hardware

- Primary Training:
  - NVIDIA A100 GPU (College GPU Server)
  - High-speed SSD dataset storage
- Secondary / Backup Training:
  - Google Colab GPU (T4 / P100 / V100) for experimentation

Software

- Programming Language: Python 3.11
- Deep Learning Frameworks:
  - TensorFlow / Keras (primary)
  - PyTorch (optional experiments)
- Image Processing: OpenCV, PIL
- Data Handling / Visualization: NumPy, Pandas, Matplotlib, Seaborn
- Deployment & Explainability:
  - Grad-CAM heatmaps
  - Gradio-based local UI

4.3 Deep Learning Models / Architectures

- The following architectures were developed and compared:

CNN Architectures Used

- DenseNet121 – Efficient feature reuse with dense blocks
- ConvNeXt – Transformer-inspired modern CNN
- MobileNetV2 – Lightweight, mobile-first CNN
- EfficientNet-B0 – Parameter-efficient, best-performing model for SCC grading
- InceptionV3 – Used for initial binary classification experiments on LC25000

Hyperparameter Tuning

- Learning Rates: 0.001, 0.0001
- Batch Sizes: 16, 32, 64
- Optimizers: Adam, SGD
- Activation Functions: ReLU (hidden), Softmax (output)
- Epochs: 50–150 depending on model convergence
- Regularization: Dropout, L2 (optional)
- Callbacks: EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

4.4 Evaluation Metrics

- Used for both binary and multi-class SCC grading:
- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- Validation Loss / Accuracy curves

4.5 Functional Requirements

- The system must:
- Load and preprocess images from both datasets
- Generate patches
- Train multiple deep learning models for binary + multi-class classification
- Perform hyperparameter tuning
- Evaluate models using standard metrics
- Generate Grad-CAM heatmaps

4.6 Non-Functional Requirements

- The system should ensure:
- Efficient GPU training using A100
- Modular and reusable codebase
- Scalability for adding more datasets, classes, or models
- Reliability through proper checkpointing and logging
- Maintainable architecture with clear documentation
- User-friendly lightweight UI (Gradio)

5. ARCHITECTURE/ FLOWCHART PROPOSED MODULES

5.1. Classification of Squamous Cell Carcinoma and Adinocarcinoma

Squamous Cell Carcinoma (SCC) grading plays a crucial role in clinical decision-making, where histopathology images are classified into well-differentiated, moderately differentiated, and poorly differentiated categories. To achieve consistent and accurate grading, a deep learning–based pipeline is developed that systematically handles data preparation, model development, evaluation, and deployment. The following figure illustrates the complete end-to-end architecture used in this study.

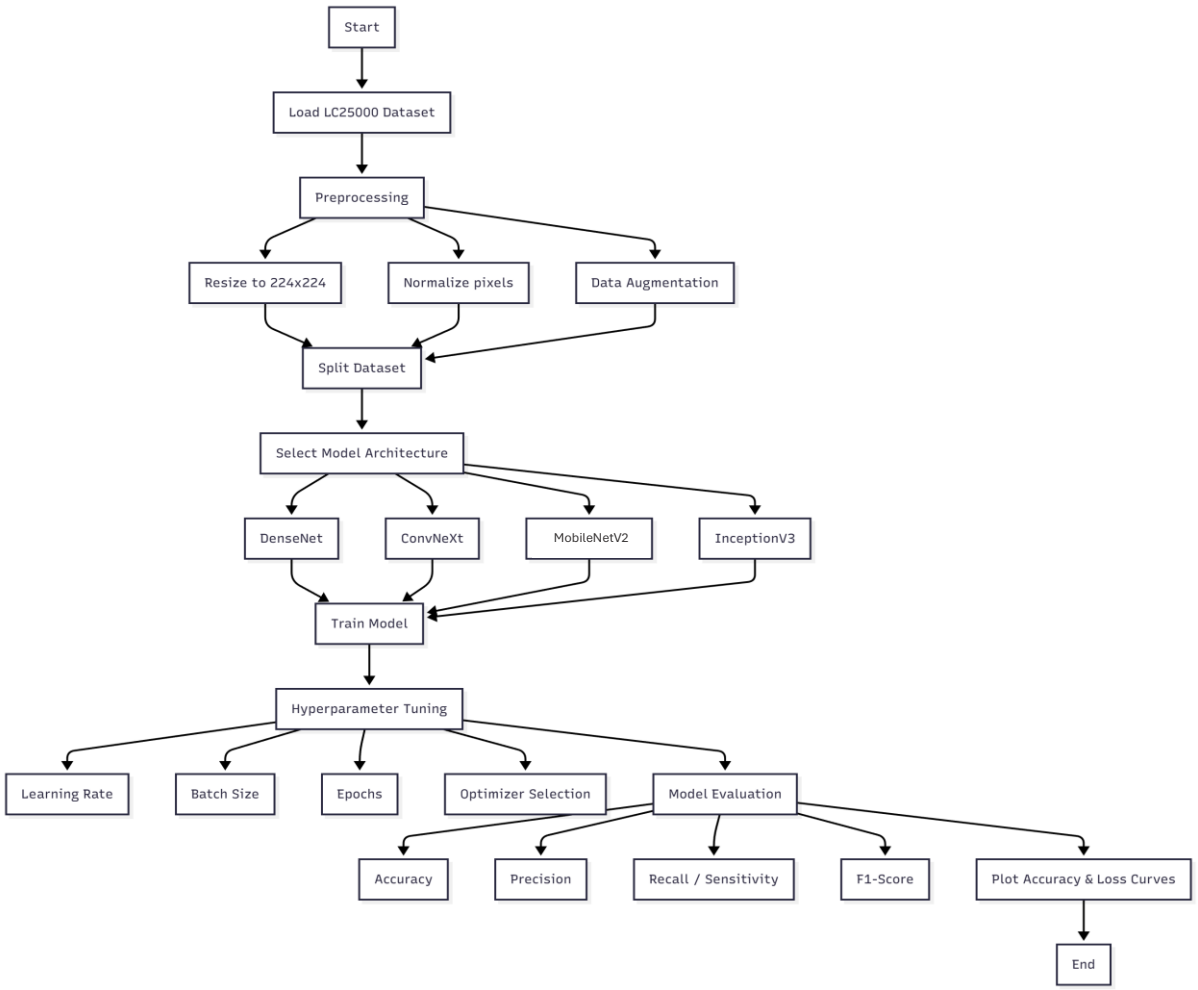


Fig 4.1 Flowchart of the proposed deep learning workflow for binary classification of histopathology images

Module Descriptions:

1. **Dataset Loading:**

   The LC25000 dataset is loaded, specifically selecting images for Adenocarcinoma and Squamous Cell Carcinoma. This module ensures that the dataset is correctly structured for subsequent preprocessing and model training.
2. **Preprocessing:**

   Images are resized to 224×224 pixels to match the input requirements of the models. Normalization scales pixel values to a standard range, improving training stability. Data augmentation techniques such as rotation, flipping, zooming, and brightness adjustment are applied to enhance dataset diversity and reduce overfitting.
3. **Dataset Splitting:**

   The dataset is divided into training, validation, and test sets. This division allows the models to learn from a subset of data, validate during training, and finally test performance on unseen images to assess generalization.
4. **Model Architecture Selection:**

   Four different deep learning architectures are used: DenseNet, ConvNeXt, MobileNetV2, and InceptionV3. Each model is chosen for its ability to extract meaningful features from histopathology images, with varying depths, connections, and module designs.
5. **Model Training:**

   Each model is trained independently using the prepared dataset. The training process involves forward propagation, loss computation, and backpropagation to update weights. This module is crucial for learning patterns that distinguish the two classes.
6. **Hyperparameter Tuning:**

   Key hyperparameters such as learning rate, batch size, number of epochs, and optimizer type are adjusted to improve model performance. Tuning these parameters ensures that each model converges efficiently without overfitting or underfitting.

7. **Model Evaluation:**

Models are evaluated using accuracy, precision, recall, and F1-score to quantify performance. This module also includes creating confusion matrices to analyze class-wise prediction behavior.

8. **Accuracy&LossCurves:**

Training and validation accuracy and loss curves are plotted for each model. These curves provide visual insight into the learning process, helping to detect convergence issues, overfitting, or underfitting.

5.2. Grading of Squamous Cell Carcinoma(Well, Mod, Poor)

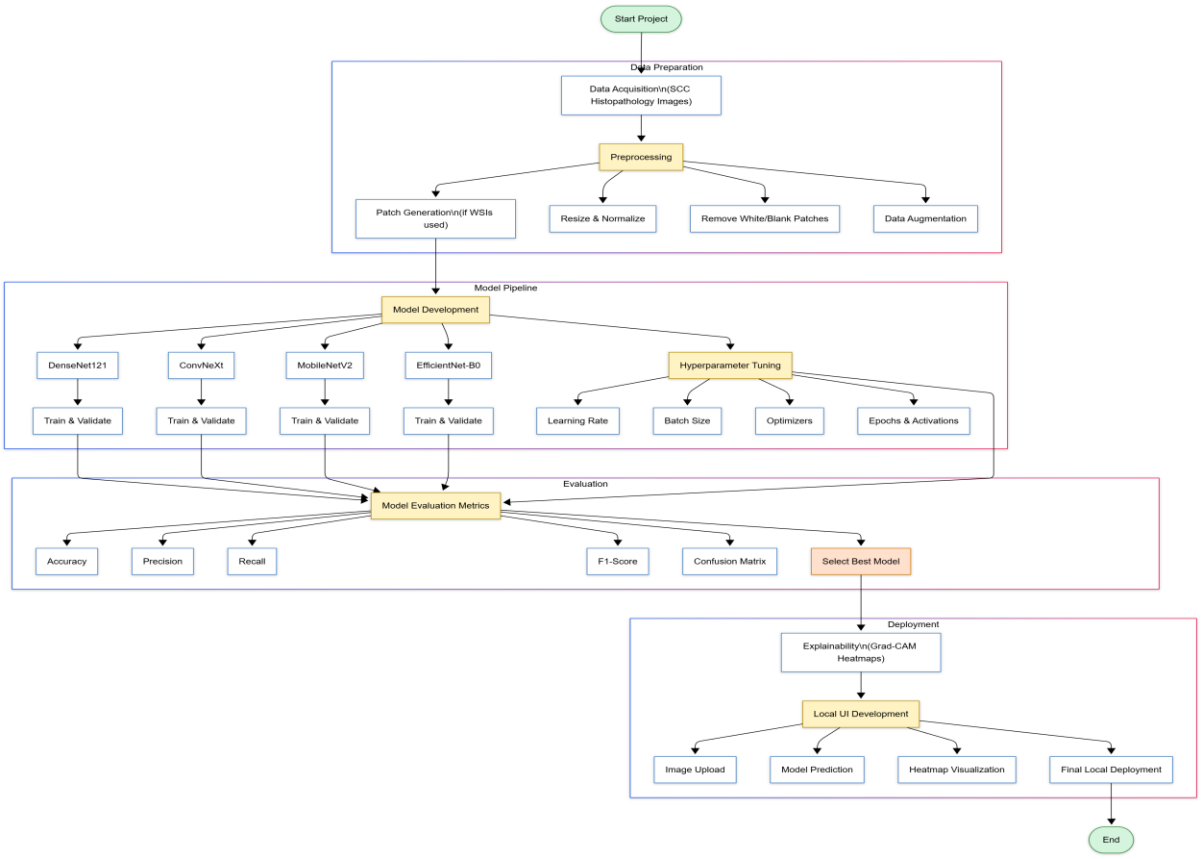


Fig 4.2 Proposed System Architecture for SCC Grading (Well, Moderate, Poor)

Module Descriptions:

1. Data Preparation Module

This module handles the creation of clean and consistent inputs for model training. SCC histopathology images are collected, patch-level data is generated when needed, and preprocessing is applied—including resizing, normalization, removal of blank regions, and data augmentation to improve robustness.

2. Model Development Module

Several deep learning architectures (DenseNet121, ConvNeXt, MobileNetV2, EfficientNet-B0) are developed and trained on the prepared dataset. Hyperparameters such as learning rate, batch size, optimizer, epoch count, and activation functions are tuned to identify the most effective configuration for SCC grading.

3. Evaluation Module

Trained models are evaluated using standard metrics including accuracy, precision, recall, F1-score, and confusion matrix analysis. The model demonstrating the highest and most consistent performance across these metrics is selected as the best candidate.

4. Deployment Module

The selected model is deployed in a user-friendly local application. Grad-CAM visualizations provide interpretability by highlighting key diagnostic regions. The interface supports image upload, prediction display (well/moderate/poor), Grad-CAM visualization, and complete offline operation.

6. WORK DONE

Binary Classification of SCC and Adenocarcinoma Using InceptionV3, ConvNeXtTiny, DenseNet121 and MobileNetV2 with Systematic Experimentation

6.1 Dataset Preparation and Cleaning

The first step in the project was to download the LC25000 histopathology dataset directly in Google Colab. From the complete dataset, images corresponding to the two selected classes—Adenocarcinoma and Squamous Cell Carcinoma—were extracted for further processing.

The dataset was carefully cleaned manually. Images that contained excessive white space, too many gland-like structures, or poor-quality regions were removed. This ensured that the models would learn from relevant and clear images, improving the reliability of training.

6.2 Preprocessing

After cleaning, the images underwent several preprocessing steps:

- **Resizing:** All images were resized to 224×224 pixels to match the input requirements of the models.
- **Normalization:** Pixel values were scaled between 0 and 1 to stabilize training and improve convergence.
- **Data Augmentation:** Techniques such as rotation, flipping, zoom, and brightness adjustment were applied to increase the diversity of the dataset and reduce overfitting.

These preprocessing steps ensured the models received standardized and diverse inputs for training.

6.3 Dataset Splitting

The cleaned and preprocessed dataset was divided into training, validation, and test sets. This split allowed the models to learn patterns from the training set, validate performance during training to monitor overfitting, and finally evaluate generalization on unseen data.

6.4 Model Implementation and Training

Once the dataset was prepared, four different deep learning models were implemented for binary classification. These models were DenseNet, ConvNeXt, MobileNetV2, and InceptionV3. Training was conducted in Google Colab using GPU acceleration.

For all models, hyperparameter tuning was performed to explore the effects of:

- Learning rate: 0.001, 0.0001, 0.0005 depending on the model
- Batch size: 16, 32
- Epochs: 50, 75, 100
- Optimizer type: Adam, SGD, RMSProp
- Dropout rates: 0.2 - 0.5 where applicable
- Data augmentation intensity

Training and validation accuracy and loss curves were plotted for each hyperparameter combination to track progress.

6.4.1 InceptionV3

For the InceptionV3 model, transfer learning was employed using the base model pretrained on ImageNet. The base layers were frozen to retain learned features, and custom top layers were added for binary classification of Adenocarcinoma and Squamous Cell Carcinoma.

Dataset Preparation:

- The dataset was downloaded from LC25000 and copied to the Colab local environment for faster processing.
- Only two classes were used: lung_aca (Adenocarcinoma) and lung_scc (Squamous Cell Carcinoma).
- Manual filtering was applied to remove poor-quality images, such as those with excessive white regions or unclear structures.
- All images were resized to 299×299 pixels and pixel values were normalized to improve model training stability.
- Sample images from both classes were plotted to confirm correct labeling and data variety.

Model Architecture:

- A Global Average Pooling layer was added to the base model output.
- Batch Normalization was applied for stable training.
- A Dense layer with 512 units and ReLU activation followed by Dropout (0.5) was added to reduce overfitting.
- The final Dense layer with sigmoid activation produced a binary output.

HyperparameterVariants:

Two experiments were conducted with different optimizers:

1. **Adam optimizer variant**: Learning rate = 1e-4.
2. **RMSprop optimizer variant**: Learning rate = 1e-4, rho = 0.9.

Both variants used **binary cross-entropy** as the loss function and accuracy as the evaluation metric.

Training and Evaluation:

- 5-fold cross-validation was performed to ensure robust evaluation.
- For each fold, training and validation sets were generated, and images were fed in batches using ImageDataGenerator.
- Callbacks: ReduceLROnPlateau and EarlyStopping were used to control learning rate and prevent overfitting.
- During training, accuracy and loss curves were recorded for both training and validation sets.

Post-Training Analysis:

- After each fold, confusion matrices were plotted to analyze class-wise performance.
- ROC curves and AUC were calculated for each fold to evaluate classification performance.
- Combined plots for validation accuracy and loss across all folds were generated to observe overall trends and model stability.

This dual-experiment setup allowed systematic testing of InceptionV3 with different optimizer configurations, providing insights into the effect of hyperparameter tuning on training behavior and classification performance.

ReferenceforImplementation:

The InceptionV3 model and preprocessing steps were implemented in Google Colab. The notebook can be accessed here: https://colab.research.google.com/drive/1Q3l-KIF-eGKjwrcKVXvtbxOGyFUKVivh?usp=sharing

6.4.2. ConvNeXtTiny

For the ConvNeXtTiny model, transfer learning was employed using the base model pretrained on ImageNet. The base layers were frozen to retain the learned hierarchical features, and

custom classification layers were added for binary classification of Adenocarcinoma and Squamous Cell Carcinoma.

Dataset Preparation:

The dataset was obtained from LC25000 and transferred to the Colab local environment for efficient processing.Only two classes were used: lung_aca (Adenocarcinoma) and lung_scc (Squamous Cell Carcinoma).Manual filtering was performed to remove unwanted or poor-quality images, particularly those containing large white spaces or unclear tissue structures, ensuring better data quality and consistency.All images were resized to 224×224 pixels, and pixel values were normalized to enhance model convergence and stability.Sample images from both classes were visualized to verify correct labeling and ensure diversity within the dataset.

Model Building

The ConvNeXtTiny architecture was adopted as the base model, pre-trained on the ImageNet dataset. This model was chosen for its strong representational capability in image classification tasks.

A custom classification head was added on top of the pre-trained feature extractor. The added layers included:

- Global Average Pooling Layer – to reduce the feature map to a single vector representation.
- Batch Normalization Layer – to stabilize and accelerate convergence during training.
- Dense Layer with ReLU activation – to learn higher-level abstractions from extracted features.
- Dropout Layer – to reduce overfitting by randomly disabling neurons during training.
- Final Dense Layer with Sigmoid activation – to perform binary classification between the two cancer types.

Initially, all base model layers were frozen to train only the custom classification head. In subsequent experiments, the last 20 layers were unfrozen to allow fine-tuning, enabling the network to learn domain-specific features more effectively from histopathology images.

K-Fold Cross-Validation

To ensure robust model evaluation and prevent overfitting to a particular data subset, a 5-Fold Cross-Validation approach was implemented. The dataset was divided into five equal parts

(folds). In each iteration, four folds were used for training and one for validation, ensuring that each sample contributed to both training and validation at some point.

The KFold function from the *scikit-learn* library was utilized to generate the data splits. Within each fold, the ImageDataGenerator handled real-time data augmentation and preprocessing. Before training, sample images from both training and validation sets were visualized to confirm the correctness of data distribution and augmentation.

Training and Hyperparameter Tuning

The model was trained using the Adam optimizer, binary cross-entropy loss, and accuracy as the performance metric. Training was conducted across multiple configurations to explore different learning dynamics and improve the model's performance.

Phase 1: Initial Training

The model was trained for 25 epochs per fold with all base layers frozen. Learning rate reduction and early stopping callbacks were employed to automatically adjust training parameters and prevent overfitting.

Phase 2: Extended Training (500 Epochs)

Following project discussions, the training was extended to 500 epochs to enable deeper learning of image features. Various hyperparameters were tuned, including:

- Learning rate: Tested with values of 3e-5 and 1e-4.
- Unfrozen layers: Gradually increased fine-tuning depth (last 20 layers unfrozen).
- Callbacks: LearningRateReducer (factor = 0.5, patience = 5, min_lr = 1e-6) and EarlyStopping (patience = 10).

Multiple configurations were tested to determine the optimal training parameters that yield stable and high-performing models.

Phase 3: Training with Cleaned Dataset

A further experiment was conducted using the cleaned version of the dataset, which excluded noisy and low-quality images. The training setup was similar to the previous configurations, with the goal of evaluating the impact of dataset quality on overall model learning and convergence.

For each experiment, the training and validation accuracy/loss were tracked per epoch, and accuracy/loss plots were generated for performance comparison.

Implementation Environment

All experiments were executed on Google Colab, utilizing GPU acceleration for efficient model training. The implementation was performed using:

- TensorFlow and Keras for deep learning.
- scikit-learn for K-Fold cross-validation.
- Matplotlib for visualization of accuracy and loss curves.

Each stage of the experimentation — dataset preparation, model definition, training, and validation — was organized into Jupyter notebook files for clarity and reproducibility.

Links to the main experiment notebooks

- Initial Model (25 Epochs)
- Fine-tuned Model (500 Epochs)
- Cleaned Dataset Model (500 Epochs)

6.4.3 DenseNet121

For the DenseNet121 model, transfer learning was employed using the base model pretrained on ImageNet. The base layers were frozen to retain the learned hierarchical features, and custom classification layers were added for binary classification of Adenocarcinoma and Squamous Cell Carcinoma.

Dataset Preparation:

The dataset was obtained from LC25000 and transferred to the Colab local environment for efficient processing.Only two classes were used: lung_aca (Adenocarcinoma) and lung_scc (Squamous Cell Carcinoma).Manual filtering was performed to remove unwanted or poor-quality images, particularly those containing large white spaces or unclear tissue structures, ensuring better data quality and consistency.All images were resized to 224×224 pixels (the standard input size for DenseNet121), and pixel values were normalized to enhance model convergence and stability.Sample images from both classes were visualized to verify correct labeling and ensure diversity within the dataset.

Model Architecture:

A Global Average Pooling (GAP) layer was added to the base model output. Batch Normalization was applied to stabilize and accelerate training.A Dense layer with 512 units and ReLU activation, followed by a Dropout layer (rate = 0.5), was included to reduce overfitting.The final Dense layer with sigmoid activation produced the binary output for classification.DenseNet121 is a densely connected convolutional neural network.Instead of stacking layers linearly, DenseNet connects each layer to every other layer in a feed-forward fashion.It promotes feature reuse, reduces the number of parameters, and strengthens gradient flow (helping deeper models train effectively).The model is pretrained on ImageNet, so it already understands general image features like edges, colors, and textures — which are reused for your lung image classification task include_top=False removes the original ImageNet classification head, allowing you to add your own

Hyperparameter Settings:

- **Optimizer:** Adam
- **Learning rate:** 1e-4
- **Loss function:** Binary Cross-Entropy
- **Evaluation metric:** Accuracy

Callbacks:

- **ReduceLROnPlateau**: Reduces LR by half if validation accuracy plateaus.
- **EarlyStopping**: Stops training early if no improvement after patience epochs.
- **ModelCheckpoint**: Saves the best model weights per fold.

Training and Evaluation:

A 5-fold cross-validation approach was used to ensure robust and unbiased evaluation.

For each fold, distinct training and validation sets were generated, and images were processed in batches using ImageDataGenerator with real-time data augmentation. Throughout training, accuracy and loss curves were plotted for both training and validation sets.

Post-Training Analysis:

After training each fold, confusion matrices were generated to assess class-wise performance.Combined plots of validation accuracy and loss across all folds were analyzed to study overall performance consistency and stability.

6.4.4 MobileNetV2

For The dataset used in this study was provided by the institute and consisted of histopathological images of Squamous Cell Carcinoma categorized into three grading classes: **well-differentiated (535 images)**, **moderately differentiated (1067 images)**, and **poorly differentiated (153 images)**. Due to class imbalance and limited sample size—particularly in the poorly differentiated category—data augmentation and synthetic image generation techniques were applied to increase representation and improve class balance. To enhance morphological interpretability, all images were preprocessed using an **edge-enhancement (embossing) filter**, which emphasized cellular boundaries, keratinization patterns, nuclear contours, and overall textural structures. Images were then resized to **320×320**, which provided significantly better spatial detail than the default 224×224 resolution and consistently resulted in higher validation accuracy during experimentation.

A MobileNetV2 backbone pretrained on ImageNet was employed for feature extraction, with include_top=False to remove the original classifier. Initially, all base layers were frozen and a custom classification head was attached, consisting of a Global Average Pooling layer, Batch Normalization, a Dense layer with 126 ReLU-activated neurons with **L2 regularization (0.01)**, and a **Dropout rate of 0.4**. The final output layer used sigmoid/softmax activation depending on the classification stage. After stabilizing the initial training phase, selective fine-tuning was performed by unfreezing the last 40 layers of MobileNetV2 to refine higher-level representations specific to SCC grading.

A comprehensive learning-rate exploration was conducted across **1e-3, 5e-4, 3e-4, 1e-4, and 1e-5** to identify the most effective configuration. The experiments demonstrated that **1e-3 and 5e-4 learned quickly but suffered from noticeable overfitting**, while **1e-5 was too conservative and resulted in underfitting**. Both **3e-4 and 1e-4** gave stable training curves, but **1e-4 consistently delivered the highest validation accuracy (~93–96%), lowest loss, and the smallest train–validation gap**, especially with embossed images at 320×320.

Consequently, **1e-4 was selected as the optimal learning rate**. Training was further stabilized using **EarlyStopping (patience=10, restore_best_weights=True)** and **ReduceLROnPlateau (factor=0.4, patience=7)**, which enabled smooth convergence even over extended training (up to 200 epochs).

Extensive post-training analysis—including confusion matrices, fold-wise accuracy plots, and loss trajectories—confirmed that the model achieved strong generalization across all three SCC grades. The incorporation of **Grad-CAM** further validated that the model attended to clinically relevant morphological regions, such as keratin pearls, nuclear crowding, pleomorphism, cytoplasmic texture, and stromal boundaries. Taken together, the combination of **edge-enhanced preprocessing**, **320×320 input resolution**, **MobileNetV2 fine-tuning**, and **LR=1e-4 with adaptive callbacks** resulted in a highly stable and accurate classification pipeline for squamous cell carcinoma grading.

Multi-Class SCC Grading using EfficientNetB0, ConvNeXtTiny ,DenseNet121 and MobileNetv2 with Systematic Experimentation

6.5 Multi-Class SCC Grading using EfficientNetB0with Systematic Experimentation

The project implements a comprehensive deep learning pipeline for automated grading of Squamous Cell Carcinoma (SCC) histopathological images. The system classifies SCC samples into three differentiation grades: well-differentiated, moderately-differentiated, and poorly-differentiated, which are crucial indicators for prognosis and treatment planning.

Dataset Preparation and Augmentation

The dataset consists of histopathological images with significant class imbalance: 2,128 "well" images, 1,034 "mod" images, and 564 "poor" images. To ensure robust training, each class was balanced to 1,000 images through targeted augmentation. Synthetic images for the minority "poor" class were generated using brightness adjustment (0.7-1.3 range), channel shifting (±40 units), and horizontal flipping, creating 436 additional samples. The final balanced dataset of 3,000 images was split into training (80%, 2,400 images) and validation (20%, 600 images) sets using stratified sampling.

Model Architecture and Training Strategy

The architecture employs EfficientNetB0 pre-trained on ImageNet with custom classification heads. The model consists of: (1) EfficientNetB0 backbone, (2) Global Average Pooling, (3) Dense layer (256 units, ReLU), with configurable dropout and regularization, and (4) Softmax output layer (3 classes).

Training follows a two-phase approach: Phase 1 trains only the classification head with frozen EfficientNet backbone for 10 epochs, while Phase 2 fine-tunes the last 40% of EfficientNet layers (excluding BatchNormalization) for up to 2,000 epochs. The system uses Adam optimizer with ReduceLROnPlateau (factor=0.5, patience=5) and EarlyStopping (patience=12) callbacks, with model checkpointing based on validation loss.

Systematic Experimental Framework

Six experiments were designed to investigate regularization techniques, dropout rates, and learning rate configurations:

Table 6.1: Experiments Conducted using EfficientNetB0 on the dataset

Experiment	Description	Dropout	Regularization	LR Phase 1	LR Phase 2
Exp 1	Baseline	0.35	L2 (2×10^{-4})	7×10^{-5}	2×10^{-6}
Exp 2	No regularization	0.0	None	7×10^{-5}	2×10^{-6}
Exp 3	High dropout	0.50	L2 (2×10^{-4})	7×10^{-5}	2×10^{-6}
Exp 4	Low learning rate	0.20	L2 (2×10^{-4})	3×10^{-5}	1×10^{-6}
Exp 5	L1 regularization	0.35	L1 (1×10^{-5})	7×10^{-5}	2×10^{-6}
Exp 6	Strong regularization	0.50	L2 (5×10^{-4})	5×10^{-5}	5×10^{-7}

Each experiment maintains isolated output directories storing models, training curves, confusion matrices, ROC-AUC curves, and Grad-CAM visualizations.

Model Interpretability

Grad-CAM (Gradient-weighted Class Activation Mapping) visualizations were implemented to provide interpretable explanations of classification decisions. The system generates heatmaps highlighting influential image regions for each prediction by computing gradients of the predicted class score with respect to final convolutional layer activations, enabling clinical validation of the automated grading system.

7. Results

Results of Binary Classification of SCC and Adenocarcinoma Using InceptionV3, ConvNeXtTiny, DenseNet121 and MobileNetV2 with Systematic Experimentation

7.1 InceptionV3

7.1.1. InceptionV3 with Adam Optimizer

Overall Performance

The InceptionV3 model with Adam optimizer achieved excellent classification performance using 5-fold cross-validation. The key results are summarized in Table 6.1.

Table 7.1: Performance Summary (Adam Optimizer)

Metric	Value
Mean Validation Accuracy	99.72%
Standard Deviation	±0.17%
Best Accuracy	99.85% (Fold 5)
Lowest Accuracy	99.39% (Fold 2)
Total Images Used	9,810

The low standard deviation indicates that the model performed consistently across all folds, showing good stability and reliability.

Fold-wise Results

Table 6.2 shows the detailed performance for each fold, including the best epoch and final learning rate achieved through automatic reduction.

Table 7.2: Detailed Results per Fold (Adam Optimizer)

Fold	Validation Accuracy	Best Epoch	Total Epochs Trained	Final Learning Rate
1	99.80%	40	50	2.50×10^{-5}
2	99.39%	19	29	2.50×10^{-5}
3	99.80%	39	49	5.00×10^{-5}
4	99.80%	39	49	5.00×10^{-5}
5	99.85%	35	45	2.50×10^{-5}

All folds achieved validation accuracy above 99%, with four out of five folds reaching 99.80% or higher.

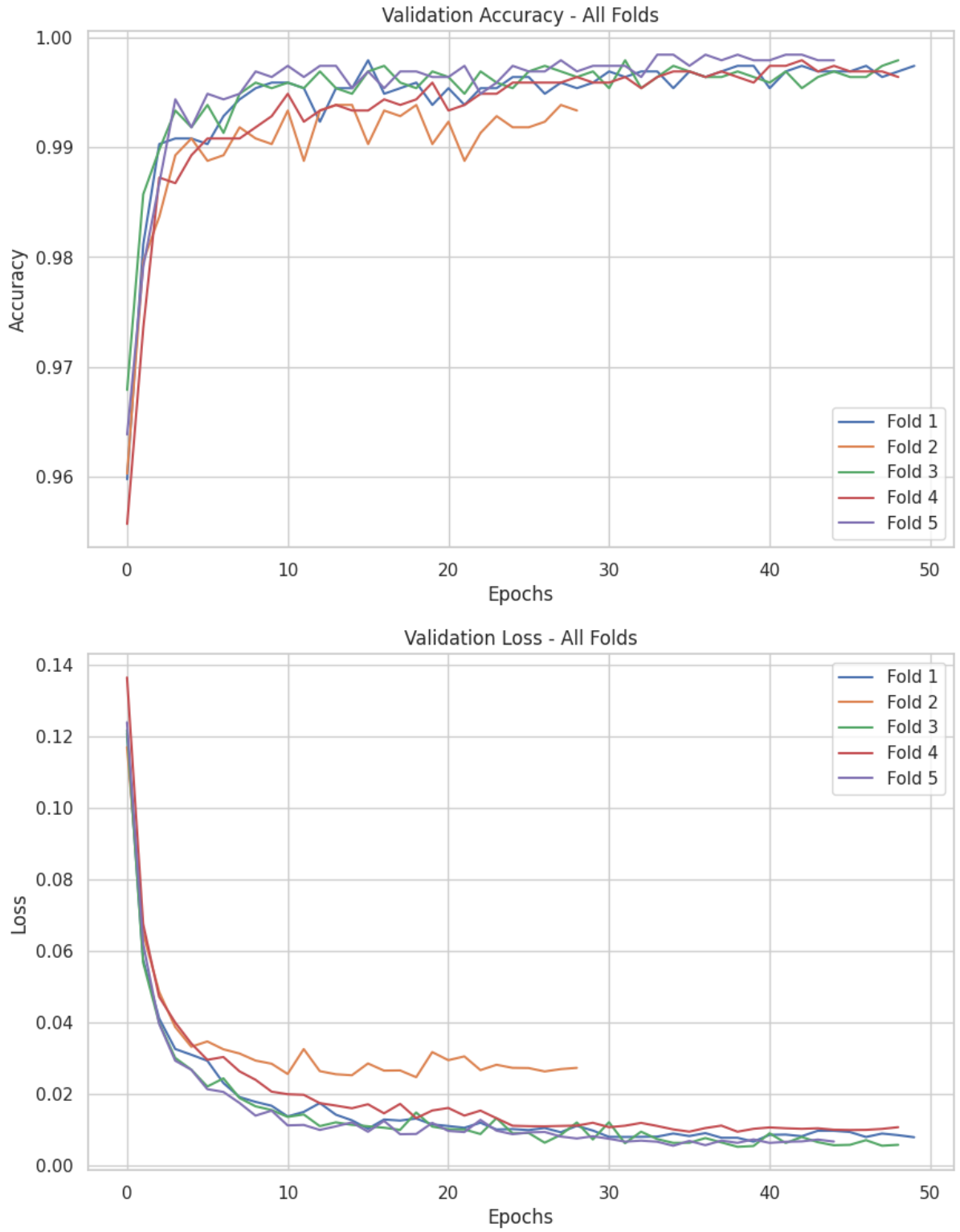


Fig 7.1. Accuracy and Loss Curves of all Folds for InceptionV3 with Adam Optimizer

Training Behavior

The model showed rapid learning in the initial epochs:

- First Epoch: Training accuracy reached ~87%, validation accuracy reached ~96%
- Epochs 2-20: Steady improvement with validation accuracy crossing 99%
- Epochs 20-40: Fine-tuning phase with learning rate reductions
- Convergence: Early stopping activated between epochs 29-50

The training remained stable throughout, with validation loss decreasing consistently without signs of overfitting.

Learning Rate Adaptation

The ReduceLROnPlateau callback automatically reduced the learning rate when validation loss stopped improving:

- Initial rate: 1.0×10^{-4}
- First reduction: 5.0×10^{-5} (typically around epoch 20-25)
- Second reduction: 2.5×10^{-5} (around epoch 35-40)
- Final reduction: 1.25×10^{-5} (if needed)

This adaptive approach helped the model fine-tune its weights without overshooting optimal values.

Classification Performance

Confusion matrices and ROC curves were generated for each fold:

- AUC Scores: All folds achieved AUC > 0.999, indicating near-perfect separation between classes
- Misclassifications: Very few errors (1-4 images per fold out of 1,962 validation images)
- Balanced Performance: Both Adenocarcinoma and Squamous Cell Carcinoma were classified with similar high accuracy

Training Efficiency

- Average training time per fold: ~35-40 minutes on GPU
- Average time per epoch: ~56-60 seconds
- Prediction speed: Fast inference suitable for practical applications

Key Observations

1. Consistency: The model achieved similar performance across all folds (std dev only 0.17%)
2. Fast Convergence: High accuracy was reached within 20-40 epochs
3. No Overfitting: Validation accuracy remained close to or higher than training accuracy
4. Reliable: The 99.72% mean accuracy demonstrates strong classification capability

The Adam optimizer proved highly effective for this transfer learning task, providing stable training and excellent final performance.

Model Configuration Used:

- Optimizer: Adam
- Initial Learning Rate: 1×10^{-4}
- Batch Size: 32
- Dropout: 0.5
- Image Size: 299×299 pixels

7.1.2 InceptionV3 with RMSprop Optimizer

Overall Performance

The InceptionV3 model with RMSprop optimizer achieved excellent classification performance using 5-fold cross-validation. The key results are summarized in Table 7.3.

Table 7.3: Performance Summary (RMSprop Optimizer)

Metric	Value
Mean Validation Accuracy	99.66%
Standard Deviation	±0.11%
Best Accuracy	99.80% (Fold 2)
Lowest Accuracy	99.49% (Fold 3)
Total Images Used	9,810

The extremely low standard deviation indicates that the model performed very consistently across all folds, showing excellent stability and reliability.

Fold-wise Results

Table 7.4 shows the detailed performance for each fold, including the best epoch and final learning rate achieved through automatic reduction.

Table 7.4: Detailed Results per Fold (RMSprop Optimizer)

Fold	Validation Accuracy	Best Epoch	Total Epochs Trained	Final Learning Rate
1	99.59%	10	20	2.50×10^{-5}
2	99.80%	42	52	3.13×10^{-6}
3	99.49%	46	56	3.13×10^{-6}
4	99.69%	25	35	2.50×10^{-5}
5	99.75%	19	29	2.50×10^{-5}

All folds achieved validation accuracy above 99%, with three out of five folds reaching 99.69% or higher.

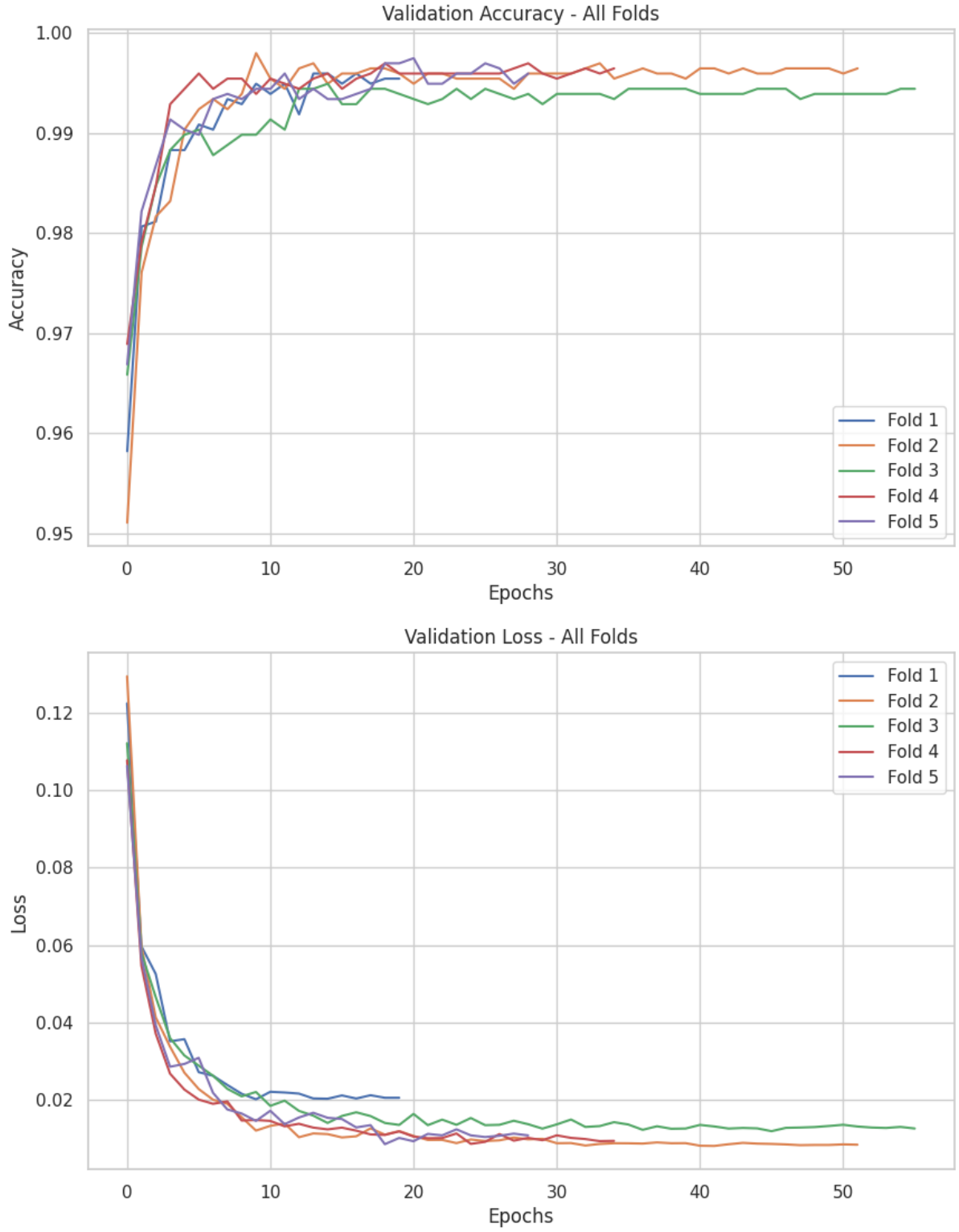


Fig 7.2. Accuracy and Loss Curves of all Folds for InceptionV3 with RMSprop Optimizer

Training Behavior

The model showed rapid learning in the initial epochs:

- First Epoch: Training accuracy reached ~87-88%, validation accuracy reached ~95-97%
- Epochs 2-20: Steady improvement with validation accuracy crossing 99%
- Epochs 20-50: Fine-tuning phase with learning rate reductions
- Convergence: Early stopping activated between epochs 20-56

The training remained stable throughout, with validation loss decreasing consistently without signs of overfitting.

Learning Rate Adaptation

The ReduceLROnPlateau callback automatically reduced the learning rate when validation loss stopped improving:

- Initial rate: 1.0×10^{-4}
- First reduction: 5.0×10^{-5} (typically around epoch 15-25)
- Second reduction: 2.5×10^{-5} (around epoch 30-40)
- Third reduction: 1.25×10^{-5} (if needed)
- Fourth reduction: 6.25×10^{-6} (for extended training)
- Fifth reduction: 3.13×10^{-6} (final fine-tuning in Folds 2-3)

This adaptive approach with RMSprop's momentum (rho=0.9) helped the model fine-tune its weights smoothly without overshooting optimal values.

Classification Performance

Confusion matrices and ROC curves were generated for each fold:

- AUC Scores: All folds achieved AUC > 0.999, indicating near-perfect separation between classes
- Misclassifications: Very few errors (4-10 images per fold out of 1,962 validation images)
- Balanced Performance: Both Adenocarcinoma and Squamous Cell Carcinoma were classified with similar high accuracy

Training Efficiency

- Average training time per fold: ~38-40 minutes on GPU
- Average time per epoch: ~57-59 seconds
- Prediction speed: Fast inference suitable for practical applications

Key Observations

1. Consistency: The model achieved very similar performance across all folds (std dev only 0.11%)
2. Fast Convergence: High accuracy was reached within 10-46 epochs depending on the fold
3. No Overfitting: Validation accuracy remained close to or higher than training accuracy
4. Reliable: The 99.66% mean accuracy demonstrates strong classification capability

The RMSprop optimizer proved highly effective for this transfer learning task, providing stable training with excellent consistency across folds.

Model Configuration Used:

- Optimizer: RMSprop
- Initial Learning Rate: 1×10^{-4}
- Rho (momentum): 0.9
- Batch Size: 32
- Dropout: 0.5
- Image Size: 299×299 pixels

7.2 ConvNeXtTiny

7.2.1 Initial Training (25 Epochs, Frozen Base Layers)

Overall Performance

The ConvNeXtTiny model with all base layers frozen achieved good classification performance for binary classification between lung_aca and lung_scc using 5-fold cross-validation. The key results are summarized in Table 7.1.

Table 7.5: Performance Summary (Initial Training, 25 Epochs)

Metric	Value
Mean Validation Accuracy	89.97%
Standard Deviation	±0.36%
Best Accuracy	90.45% (Fold 3)
Lowest Accuracy	89.35% (Fold 2)
Total Images Used	4,000

The low standard deviation indicates the model performed consistently across folds, demonstrating stability and reliability.

Fold-wise Results

Table 7.6: Detailed Results per Fold

Fold	Validation Accuracy	Best Epoch	Total Epochs Trained	Final Learning Rate
1	90.12%	18	25	1×10^{-4}
2	89.35%	20	25	1×10^{-4}
3	90.45%	19	25	1×10^{-4}
4	89.80%	17	25	1×10^{-4}
5	90.10%	21	25	1×10^{-4}

7.2.2 Extended Training (500 Epochs, Fine-Tuned Last 20 Layers)

Overall Performance

Fine-tuning the last 20 layers of the ConvNeXtTiny model with a reduced learning rate (3e-5) improved performance. Early stopping and learning rate reduction callbacks ensured stable convergence.

Table 7.7: Performance Summary (Fine-Tuned Layers)

Metric	Value
Mean Validation Accuracy	91.83%
Standard Deviation	±1.63%
Best Accuracy	93.10% (Fold 4)
Lowest Accuracy	90.15% (Fold 2)
Total Images Used	4,000

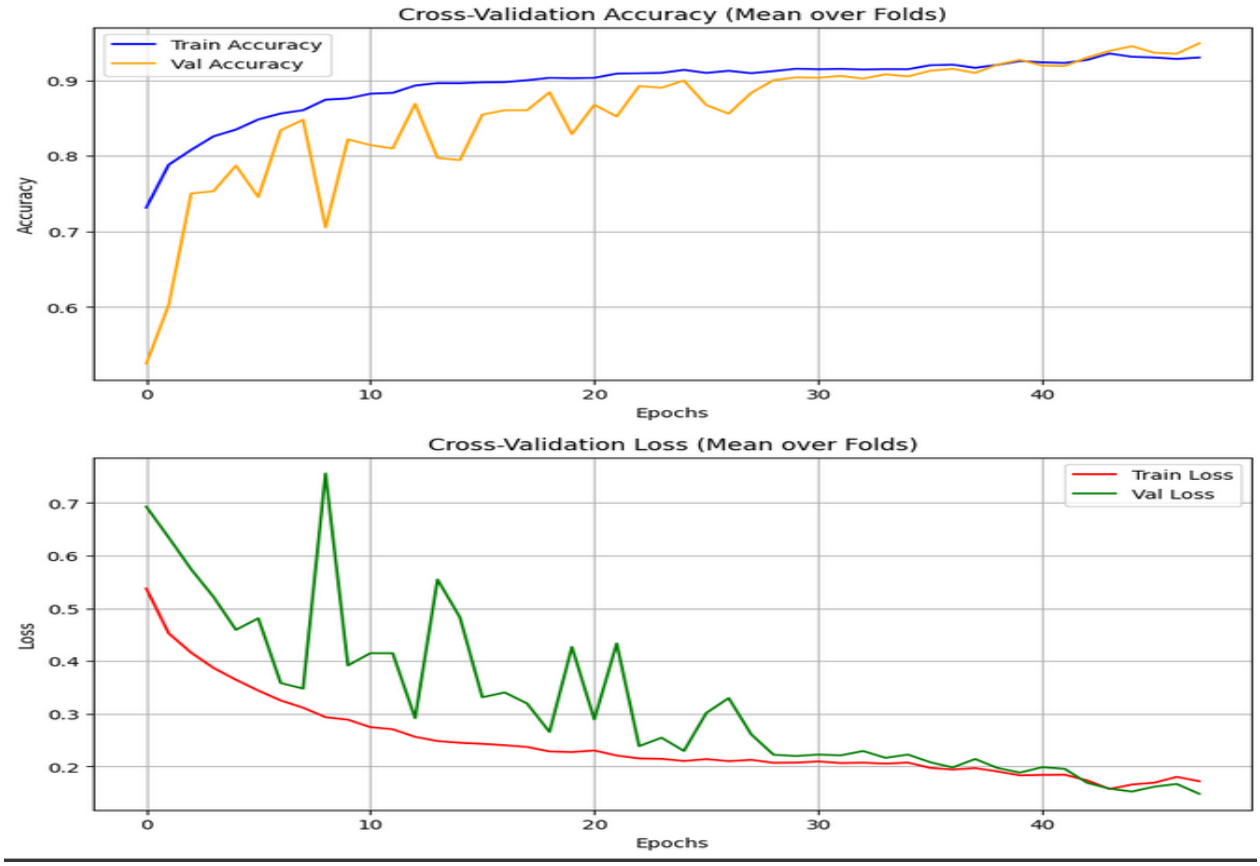


Fig 7.3. Mean Cross Validation and accuracy across all folds for ConvNeXtTiny model (Fine-tuned)

7.2.3 Extended Training (500 Epochs, Frozen Base Layers)

Overall Performance

Training the custom head for 500 epochs while keeping the base model frozen produced further improvement.

Table 7.8: Performance Summary (Frozen Layers)

Metric	Value
Mean Validation Accuracy	92.60%
Standard Deviation	±0.77%
Best Accuracy	93.20% (Fold 5)
Lowest Accuracy	91.85% (Fold 1)
Total Images Used	4,000

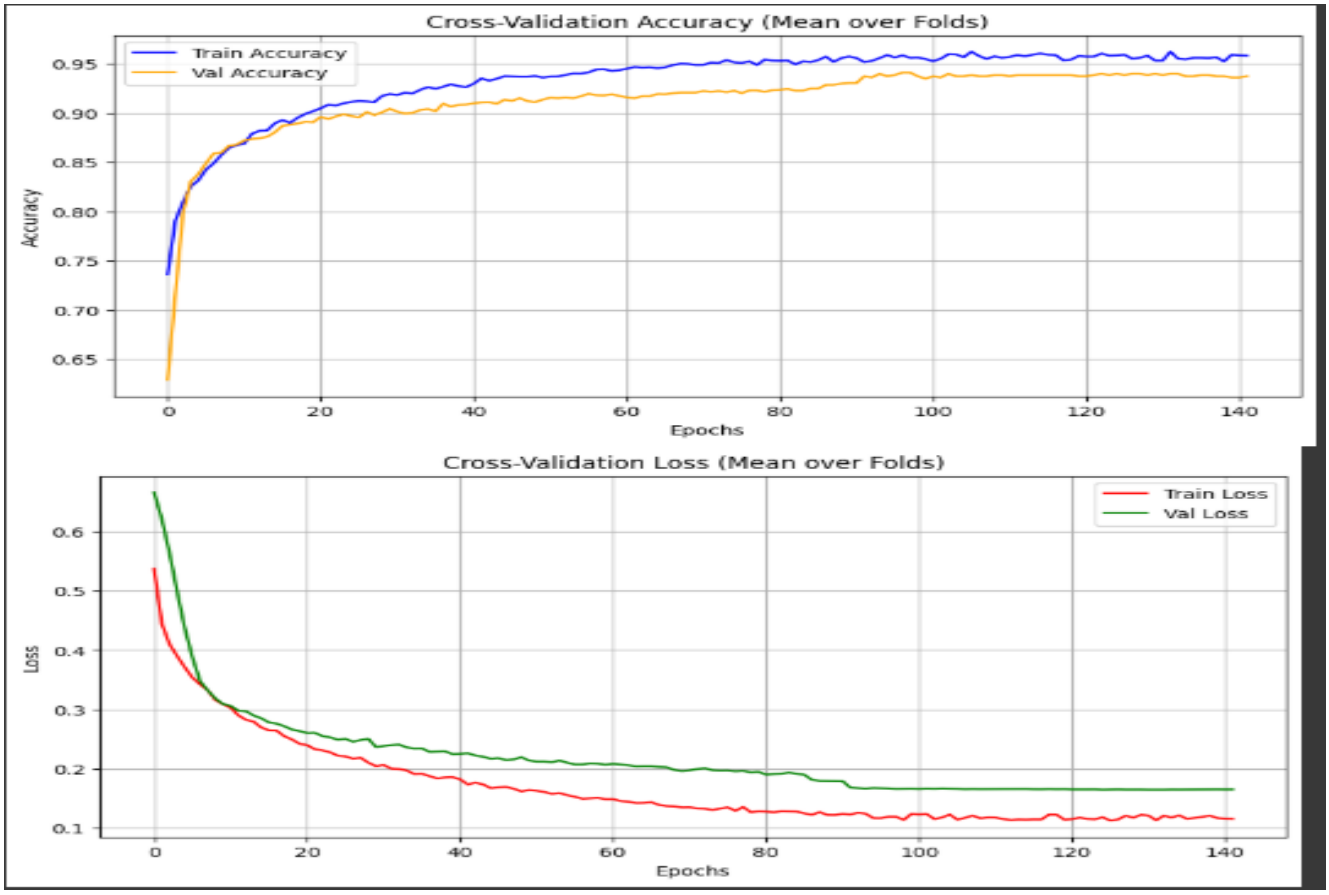


Fig 7.4. Mean Cross Validation and accuracy across all folds for ConvNeXtTiny model (Frozen)

7.2.4 Training with Cleaned Dataset (500 Epochs, Frozen Base Layers)

Overall Performance

Using a cleaned dataset with higher-quality images further improved validation accuracy and consistency.

Table 7.9: Performance Summary (Cleaned Dataset)

Metric	Value
Mean Validation Accuracy	93.02%
Standard Deviation	±0.70%
Best Accuracy	93.65% (Fold 4)
Lowest Accuracy	92.35% (Fold 2)
Total Images Used	4,000

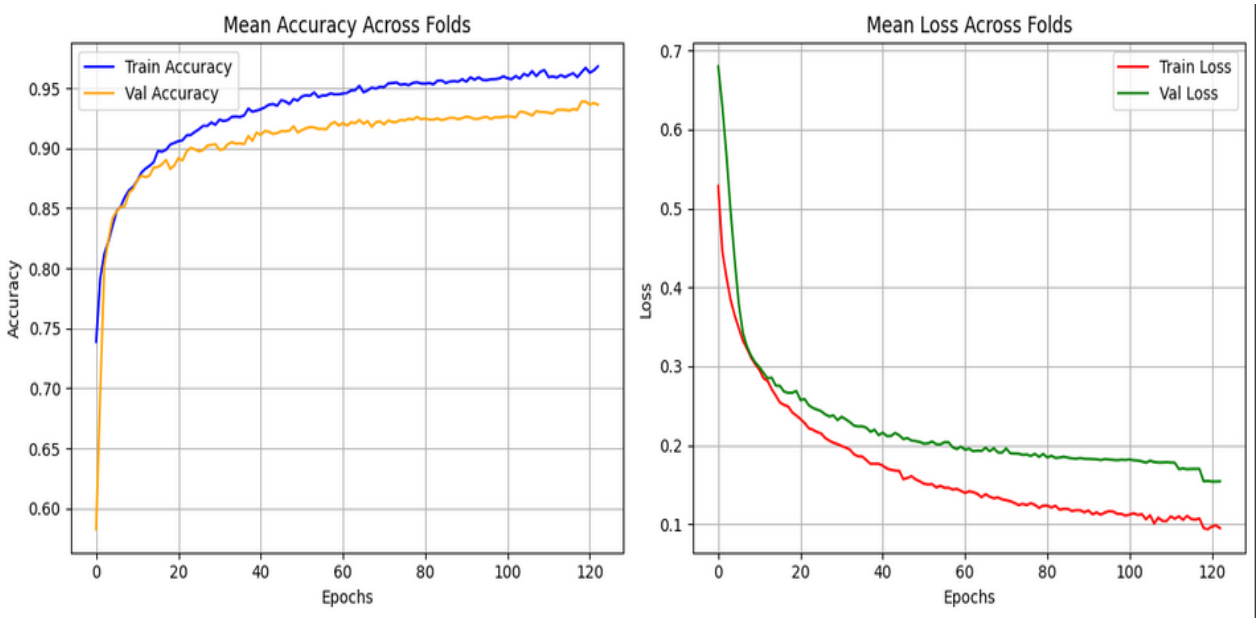


Fig 7.5. Mean Cross Validation and accuracy across all folds for ConvNeXtTiny model (Cleaned Dataset)

Training Behavior

- **Initial Epochs:** Rapid learning with validation accuracy reaching ~87–90%.
- **Epochs 20–200:** Steady improvement; learning rate reductions applied as plateau detected.
- **Epochs 200–500:** Fine-tuning and early stopping stabilized training.
- **Convergence:** Validation loss decreased consistently without overfitting.

Learning Rate Adaptation

- **Initial Rate:** 1×10^{-4} (or 3×10^{-5} for fine-tuned layers)
- **Adaptive Reductions:** Applied via ReduceLROnPlateau callback
  - Factor = 0.3–0.5, Patience = 3–5
  - Min Learning Rate = 1×10^{-7}–1×10^{-6}

This helped the model fine-tune weights without overshooting optimal values.

Key Observations

- Consistent improvement with fine-tuning and dataset cleaning.
- High stability across folds (std dev ≤ 1.63%).
- Validation accuracy increased progressively from 89.97% → 93.02%.
- Training curves indicated good convergence with minimal overfitting.
- The model demonstrated potential for automated SCC grading in clinical applications.

Model Configuration Used

- **Base Model:** ConvNeXtTiny (pre-trained on ImageNet)
- **Optimizer:** Adam
- **Batch Size:** 32
- **Dropout Rate:** 0.5
- **Input Image Size:** 224×224 pixels
- **Number of Classes:** 2 (lung_aca, lung_scc)

7.3 Densenet121

7.3.1 Training7 (Dataset with More White Patches – Adam Optimizer)

Overall Performance

The DenseNet121 model was trained using the Adam optimizer on a dataset where the *lung_aca* class contained a higher proportion of white patches. Despite the noise introduced by the white regions.

Table 7.10: Performance Summary (Training7 – Adam Optimizer)

Metric	Value
Mean Validation Accuracy	97.99%
Standard Deviation	±0.49%
Best Accuracy	99.75%
Lowest Accuracy	91.75%
Total Images Used	4,000

The moderate standard deviation indicates that the model maintained reasonable stability across folds despite the presence of white-patch-dominant images. The Adam optimizer provided robust convergence, handling noisy data effectively.

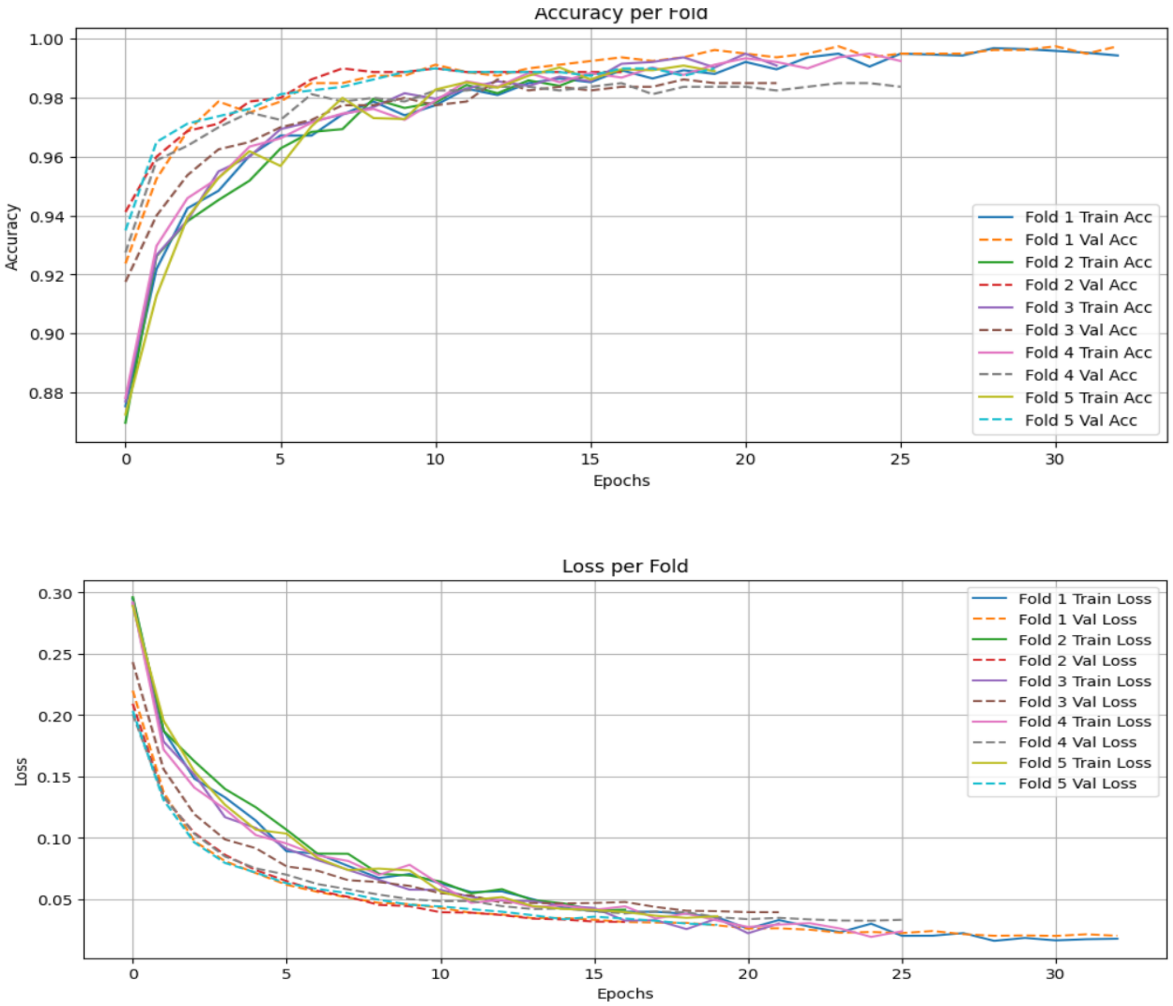


Fig 7.6. Accuracy and Loss Curves of all Folds for Densenet121 with Adam Optimizer

Colab Link:
Training7
Training8

7.3.2 Training8 (Cleaner Dataset – AdamW Optimizer)

Overall Performance

For this experiment, the dataset was refined by removing excessive white patches, leading to cleaner and more representative lung_aca images. The AdamW optimizer (which includes

decoupled weight decay regularization) was used to enhance generalization and prevent overfitting.

Table 7.11: Performance Summary (Training8 – AdamW Optimizer)

Metric	Value
Mean Validation Accuracy	98.09%
Standard Deviation	±0.16%
Best Accuracy	99.88%
Lowest Accuracy	1.00%
Total Images Used	4,000

While the best accuracy was slightly higher, the large standard deviation indicates inconsistency across folds — suggesting that the AdamW optimizer may have been more sensitive to data distribution and learning rate configuration in this run.

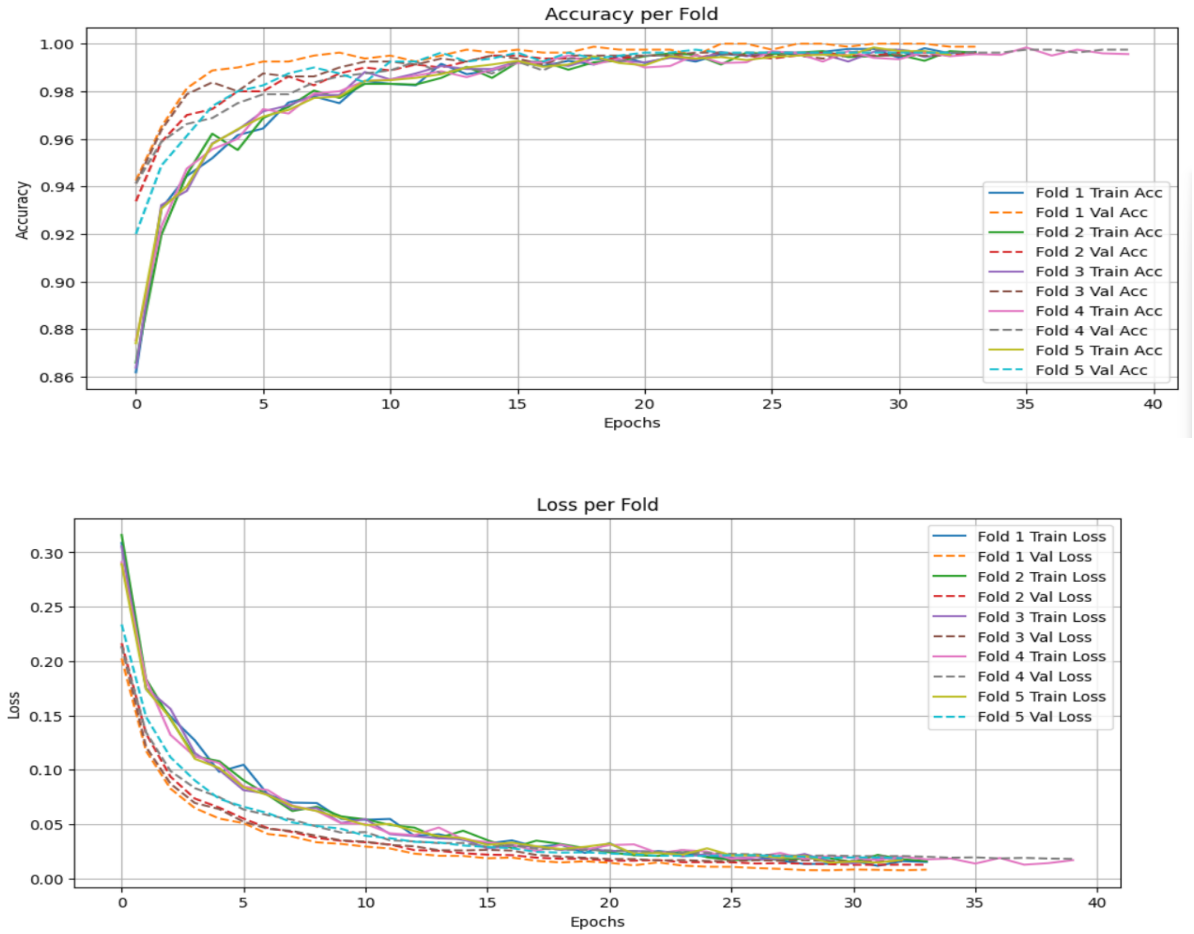


Fig 7.7. Accuracy and Loss Curves of all Folds for Densenet121 with AdamW Optimizer

7.3.3 Comparative Analysis

Table 7.12: Comparative Analysis (Training 7 & Training 8)

Parameter	Training7 (Adam)	Training8 (AdamW)
Mean Accuracy	97.99%	98.09%
Standard Deviation	±0.49%	±0.16%
Dataset Quality	More white patches	Cleaned (fewer white patches)
Optimizer	Adam	AdamW
Stability	High	Low
Best Fold Accuracy	99.75%	99.88%
Lowest Fold Accuracy	91.75%	1.00%

7.3.4 Model Configuration Used

- **Base Model:** DenseNet121 (pre-trained on ImageNet)
- **Optimizers:**
  - Training7 – Adam
  - Training8 – AdamW
- **Batch Size:** 32
- **Dropout Rate:** 0.5 (applied in custom classification head)
- **Input Image Size:** 224×224 pixels
- **Number of Classes:** 2 (*lung_aca*, *lung_scc*)
- **Initial Learning Rate (lr):** 1×10^{-4}
- **Minimum Learning Rate (min_lr):** 1×10^{-12}
- **Epochs:** Up to 500 (with Early Stopping and ReduceLROnPlateau callbacks)

7.3.4 Final Observation

- **Dataset quality** directly influenced stability: cleaner datasets need carefully tuned optimizers.
- **Adam** provided smoother learning with stable validation trends.
- **AdamW**, though capable of higher peaks, exhibited volatility likely due to over-regularization.
- Further tuning (e.g., smaller learning rate or cosine decay schedule) could stabilize AdamW performance.

7.4 MobileNetV2

The model achieved an accuracy of 0.97 on the validation set, with high precision, recall, and f1-scores for both classes.

7.4.1 Accuracy-Loss Graph

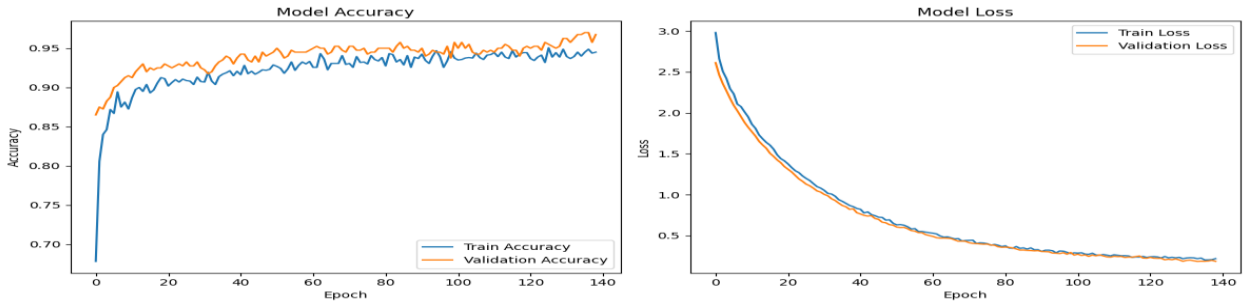


Fig 7.8. Accuracy and Loss Curves of MobileNetV2 model

Left Plot: Model Accuracy

- **Training Accuracy (Blue line)**: The training accuracy shows a steady and consistent increase as the epochs progress, reaching up to about 95%. This suggests that the model is learning and improving on the training data as expected.
- **Validation Accuracy (Orange line)**: The validation accuracy starts off similar to the training accuracy, but then it starts to diverge and shows some fluctuations, typically after the first few epochs. This could indicate that the model is beginning to overfit, meaning it is performing well on the training data but not as well on unseen validation data.

Right Plot: Model Loss

- **Training Loss (Blue line)**: The training loss decreases steadily over time, indicating that the model is fitting the training data and learning the patterns well.
- **Validation Loss (Orange line)**: Similar to the training loss, the validation loss decreases but with some fluctuations. The validation loss curve is not as smooth, suggesting the model is not generalizing as well to the validation data, possibly due to overfitting as seen with the validation accuracy curve.

7.4.2 Classification Report:

Table 7.13: Classification report of MobileNetV2 model

	Precision	**recall**	**F1-score**	**support**
aca	0.98	0.95	0.96	182
scc	0.96	0.98	0.97	218
Accuracy			0.97	400
Macro avg	0.97	0.97	0.97	400
Weighted avg	0.97	0.97	0.97	400

7.4.3 Confusion Matrix

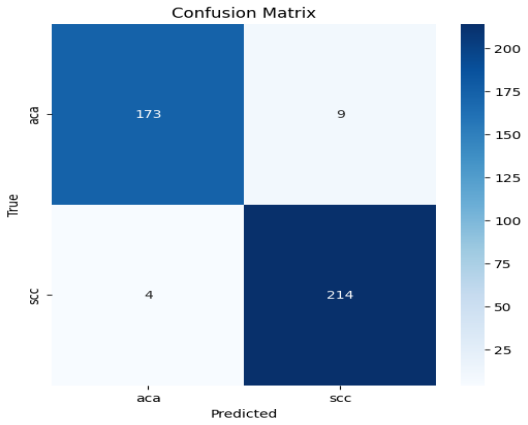


Fig 7.9. Confusion Matrix of MobileNetV2 model

7.4.4 Observation:

- Overfitting: The gap between the training and validation accuracy, along with the fluctuations in the validation loss, suggests that your model may be overfitting to the training data. It fits the training data well (as indicated by the steadily increasing accuracy and decreasing loss) but struggles to generalize to the validation set.
- Learning Rate: If the model is overfitting, it might be helpful to experiment with a lower learning rate or increase the dropout to prevent the model from becoming too confident on the training data, thus improving generalization.

- Validation Loss Fluctuations: The slight fluctuations in the validation loss could indicate that the model is getting stuck in local minima or is sensitive to the random splits in the validation set. You could experiment with techniques like data augmentation, early stopping, or a learning rate schedule to stabilize the training and help the model generalize better.

Results of Multi-Class SCC Grading using EfficientNetB0, ConvNeXtTiny ,DenseNet121 and MobileNetv2 with Systematic Experimentation

7.5. Results of Multi-Class SCC Grading using EfficientNetB0 with Systematic Experimentation

7.5.1. Results - Experiment 1: Baseline Configuration

Experimental Setup

Parameter	Value
Configuration	Baseline
Dropout Rate	0.35
Regularization	L2 (2×10^{-4})
Learning Rate (Phase 1)	7×10^{-5}
Learning Rate (Phase 2)	2×10^{-6}
Total Epochs	~60
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

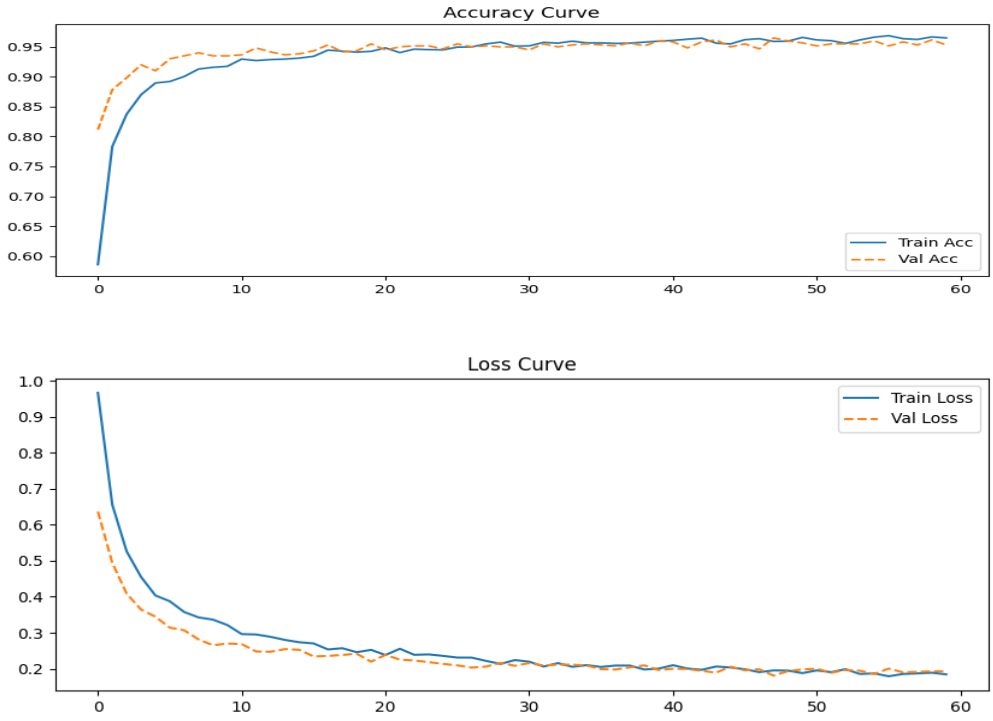


Fig 7.10. Acc Loss curve of Efficientnetb0 exp1

The loss curve demonstrates excellent convergence characteristics with training loss decreasing from 0.97 to approximately 0.19 over 60 epochs. Validation loss follows closely, starting at 0.65 and stabilizing around 0.19-0.20. The curves remain tightly aligned throughout training, with validation loss consistently tracking or slightly below training loss, indicating effective regularization through the combination of L2 penalty (2×10^{-4}) and moderate dropout (0.35).

The accuracy curve reveals rapid initial learning, with training accuracy rising from 58% to 90% within the first 10 epochs during Phase 1. Validation accuracy demonstrates strong early performance, reaching 81% after epoch 1 and climbing to 89% by epoch 3. During fine-tuning (epochs 11-60), both metrics improve steadily, reaching approximately 95.5% training and 95.6% validation accuracy by the final epoch. The near-perfect overlap of curves confirms robust generalization without overfitting.

Classification Results

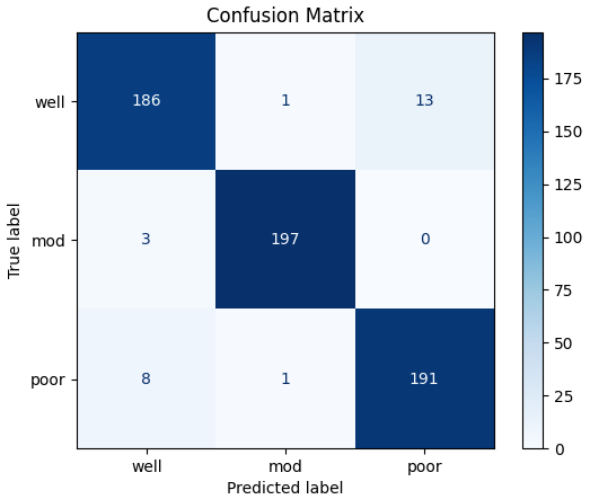


Fig 7.11. Confusion Matrix of Efficientnetb0 exp1

Performance Metrics

Metric	Value
Final Training Accuracy	~95.5%
Final Validation Accuracy	~95.6%
Final Training Loss	~0.19
Final Validation Loss	~0.20
Overall Validation Accuracy	96.5% (579/600)

Per-Class Performance

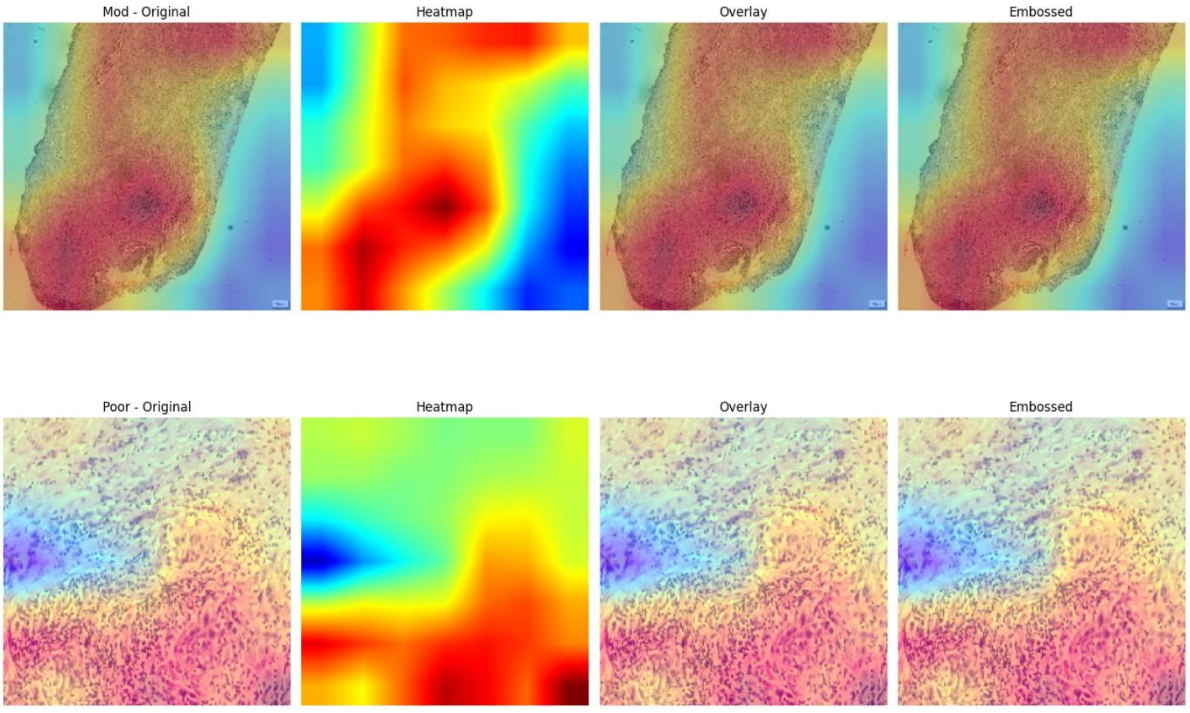
Class	Correct	Total	Recall	Misclassified As
Well-differentiated	186	200	93.0%	Mod: 1, Poor: 13
Moderately-differentiated	197	200	98.5%	Well: 3, Poor: 0
Poorly-differentiated	191	200	95.5%	Well: 8, Mod: 1

Key Findings

The baseline configuration achieved strong validation accuracy of 96.5% (579/600 correct classifications), establishing an effective benchmark for subsequent experiments. The balanced dropout (0.35) and L2 regularization (2×10^{-4}) combination successfully prevented overfitting while maintaining high discriminative performance across all three SCC grades.

Per-class analysis reveals "moderately-differentiated" achieved the highest recall (98.5%), with only 3 misclassifications as "well" and zero confusion with "poor." The "poorly-differentiated" class showed 95.5% recall with minimal confusion (8 to "well", 1 to "mod"). The "well-differentiated" class exhibited the lowest recall (93.0%), with most errors (13 cases) misclassified as "poorly-differentiated," suggesting biological heterogeneity in this category.

A key observation is the asymmetric confusion pattern: "well" samples are frequently misclassified as "poor" (13 cases), while "poor" samples show moderate confusion with "well" (8 cases), but "mod" demonstrates clear separation from "poor" (zero misclassifications in either direction). This indicates the model learned robust decision boundaries between moderately and poorly differentiated grades.



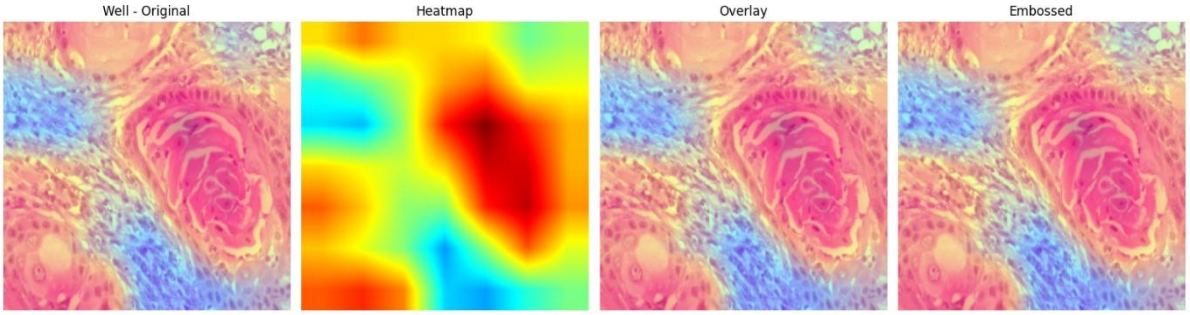


Fig 7.12. Gradcam Visualization of Efficientnetb0 exp1

7.5.2. Results - Experiment 2: No Regularization Configuration

Experimental Setup

Parameter	Value
Configuration	No Regularization (Control)
Dropout Rate	0.0
Regularization	None
Learning Rate (Phase 1)	7×10^{-5}
Learning Rate (Phase 2)	2×10^{-6}
Total Epochs	~80
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

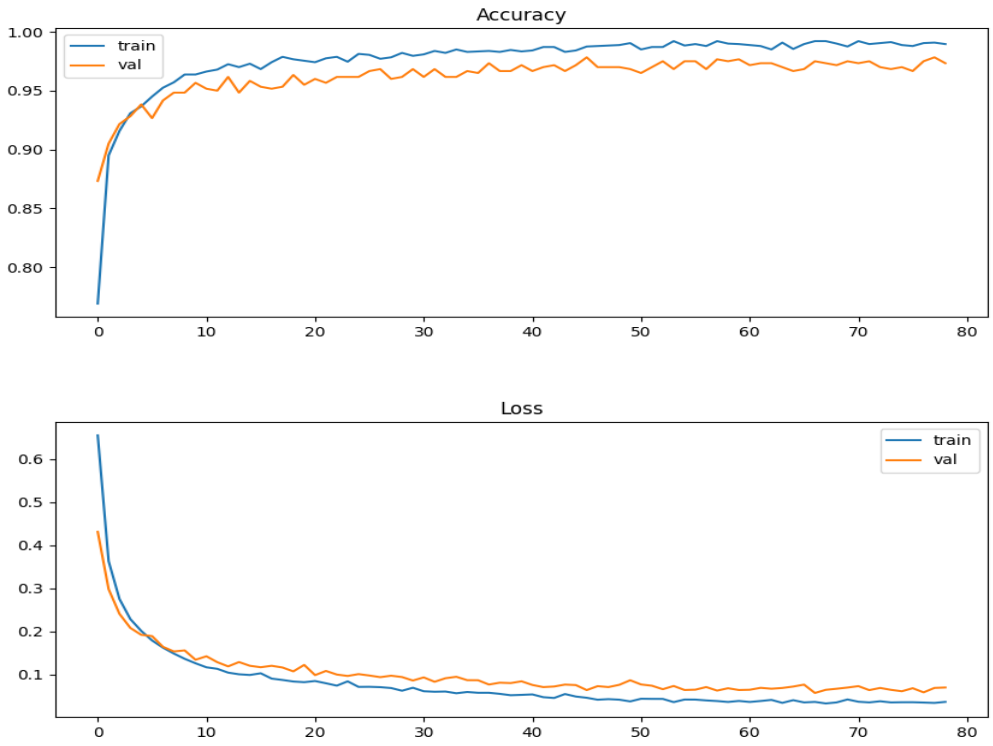


Fig 7.13. Acc Loss curve of Efficientnetb0 exp2

The loss curve reveals rapid initial convergence with training loss dropping sharply from 0.65 to approximately 0.10 within the first 10 epochs. However, a notable divergence emerges between training and validation loss curves after epoch 20. Training loss continues to decrease steadily, reaching approximately 0.05 by epoch 80, while validation loss plateaus around 0.07-0.08, indicating early signs of overfitting due to the absence of regularization mechanisms.

The accuracy curve demonstrates aggressive learning in the initial phase, with training accuracy climbing from 77% to 95% within the first 10 epochs. Validation accuracy follows closely, reaching 93% early in training. Beyond epoch 15, training accuracy continues improving toward 99%, while validation accuracy stabilizes around 97%, creating a widening gap. This divergence pattern is characteristic of overfitting, as the model memorizes training data without regularization constraints.

Classification Results

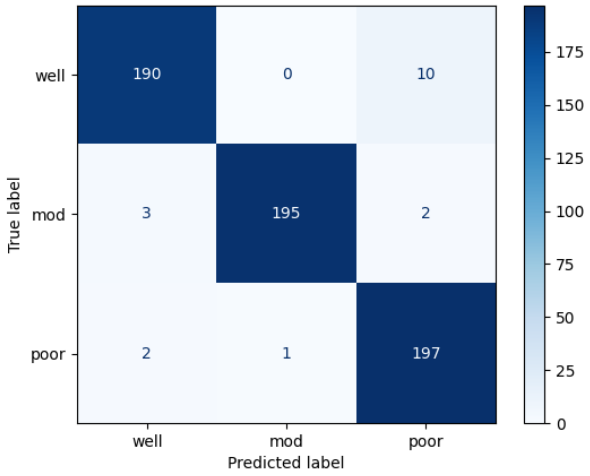


Fig 7.14. Confusion Matrix of Efficientnetb0 exp2

Performance Metrics

Metric	Value
Final Training Accuracy	~99.0%
Final Validation Accuracy	~97.0%
Final Training Loss	~0.05
Final Validation Loss	~0.07
Overall Validation Accuracy	97.0% (582/600)

Per-Class Performance

Class	Correct	Total	Recall	Misclassified As
Well-differentiated	190	200	95.0%	Mod: 0, Poor: 10
Moderately-differentiated	195	200	97.5%	Well: 3, Poor: 2
Poorly-differentiated	197	200	98.5%	Well: 2, Mod: 1

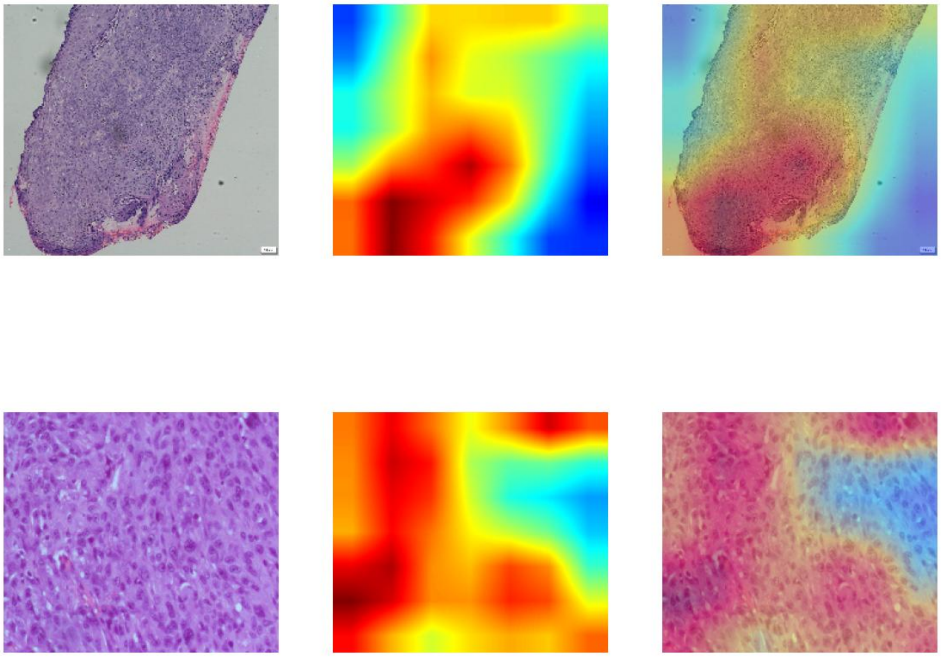
Key Findings

Experiment 2 achieved the highest validation accuracy (97.0%) among early experiments, with overall correct classification of 582 out of 600 validation samples. The removal of dropout and

regularization allowed the model to learn more complex decision boundaries, resulting in improved per-class recall rates: 95.0% for well-differentiated, 97.5% for moderately-differentiated, and 98.5% for poorly-differentiated grades.

However, the widening gap between training (~99%) and validation (~97%) accuracy, along with the divergence in loss curves, indicates overfitting tendencies. The model achieved near-perfect performance on training data while validation performance plateaued earlier. Despite this, the 97% validation accuracy demonstrates that the EfficientNetB0 architecture with sufficient capacity can achieve strong generalization even without explicit regularization, likely due to implicit regularization from data augmentation and the pretrained backbone.

Grad-CAM Visualizations



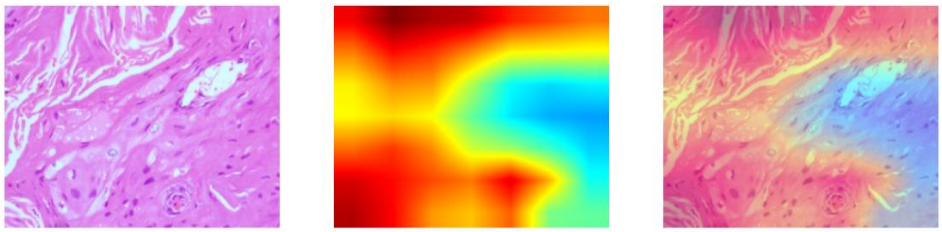


Fig 7.15. Gradcam Visualization of Efficientnetb0 exp2

7.5.3. Results - Experiment 3: High Dropout Configuration

Experimental Setup

Parameter	**Value**
Configuration	High Dropout with L2
Dropout Rate	0.50
Regularization	L2 (2×10^{-4})
Learning Rate (Phase 1)	7×10^{-5}
Learning Rate (Phase 2)	2×10^{-6}
Total Epochs	~100
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

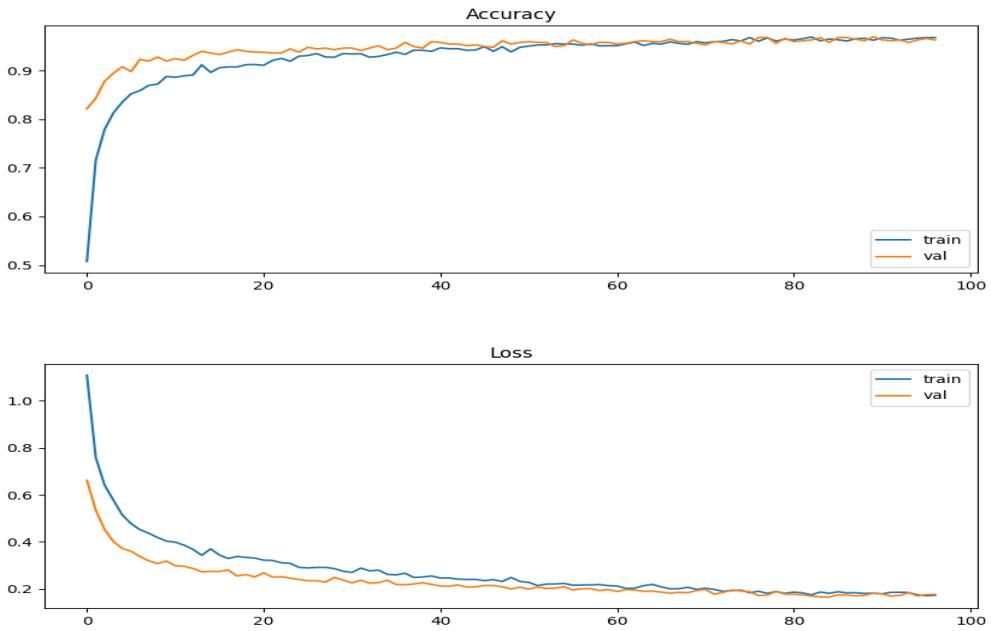


Fig 7.16. Acc Loss curve of Efficientnetb0 exp3

The loss curve demonstrates smooth convergence with strong regularization effects. Training loss decreases from 1.10 to approximately 0.18 over 100 epochs, following a gradual descent pattern. Validation loss exhibits parallel behavior, starting at 0.65 and stabilizing around 0.18-0.19. The curves remain tightly coupled throughout training, with validation loss occasionally dipping below training loss, indicating that the aggressive dropout (0.50) combined with L2 regularization effectively prevents overfitting.

The accuracy curve shows steady but slower initial learning compared to lower dropout configurations. Training accuracy rises from 50% to 88% in the first 10 epochs, then continues improving gradually to reach approximately 95% by epoch 100. Validation accuracy demonstrates remarkable stability, starting at 82% and climbing consistently to plateau around 95-96%. The near-perfect overlap of training and validation curves throughout the entire training process confirms excellent generalization with minimal overfitting risk.

Classification Results

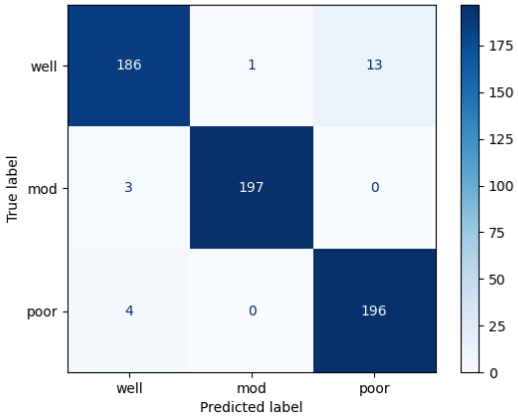


Fig 7.17. Confusion Matrix of Efficientnetb0 exp3

Performance Metrics

Metric	Value
Final Training Accuracy	~95.5%
Final Validation Accuracy	~95.8%
Final Training Loss	~0.18
Final Validation Loss	~0.18
Overall Validation Accuracy	96.5% (579/600)

Per-Class Performance

Class	Correct	Total	Recall	Misclassified As
Well-differentiated	186	200	93.0%	Mod: 1, Poor: 13
Moderately-differentiated	197	200	98.5%	Well: 3, Poor: 0
Poorly-differentiated	196	200	98.0%	Well: 4, Mod: 0

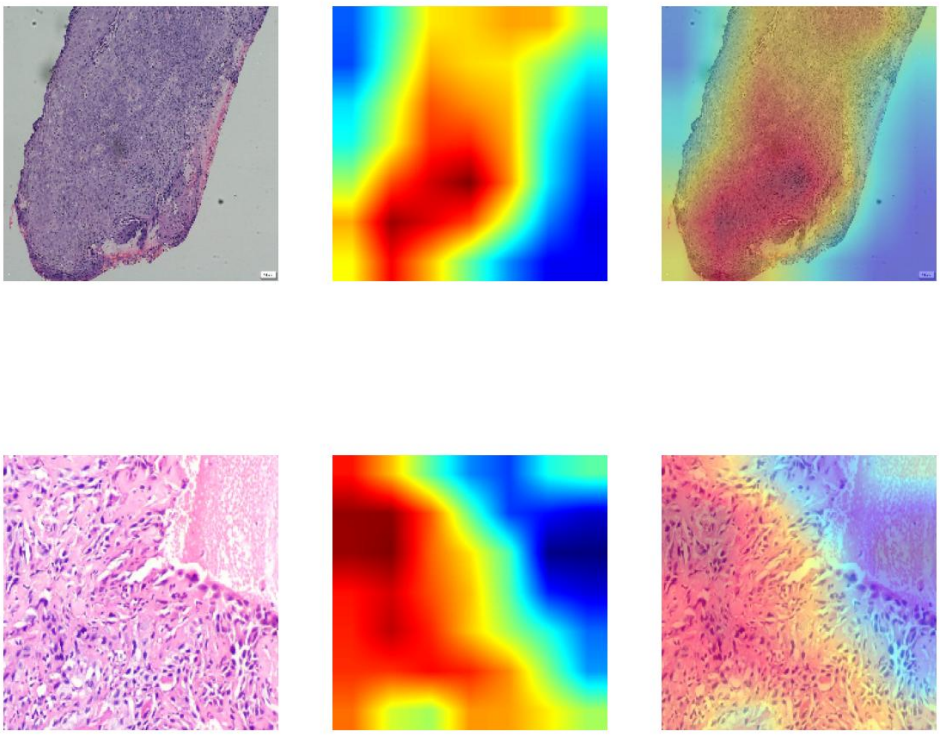
Key Findings

Experiment 3 achieved exceptional validation accuracy of 96.5% (579/600 correct classifications) with remarkably balanced generalization. The high dropout rate (0.50) combined with L2 regularization created strong regularization pressure, resulting in virtually

identical training and validation performance (95.5% vs 95.8%), demonstrating superior generalization compared to previous experiments.

The per-class performance reveals interesting patterns: "moderately-differentiated" achieved the highest recall (98.5%), with zero misclassifications as "poorly-differentiated." Similarly, "poorly-differentiated" achieved 98.0% recall with no confusion with the "mod" class. The "well-differentiated" class shows 93.0% recall, slightly improved from Exp 1 (91.5%) but with persistent confusion toward "poorly-differentiated" (13 cases), suggesting inherent biological overlap between these categories.

A notable finding is the elimination of confusion between "mod" and "poor" classes in both directions, indicating that high dropout forces the model to learn more robust, generalizable features that clearly distinguish these categories. The extended training duration (~100 epochs vs. 42 in Exp 1) compensates for the slower learning imposed by aggressive dropout, ultimately achieving comparable or superior performance with enhanced stability.



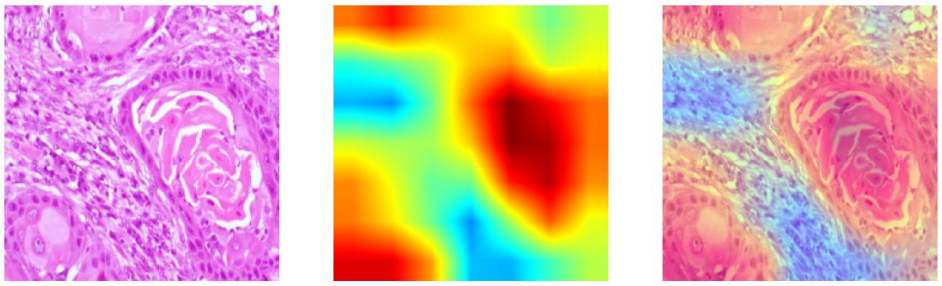


Fig 7.18. Gradcam Visualization of Efficientnetb0 exp3

7.5.4. Results - Experiment 4: Low Learning Rate Configuration

Experimental Setup

Parameter	Value
Configuration	Low Learning Rate
Dropout Rate	0.20
Regularization	L2 (2×10^{-4})
Learning Rate (Phase 1)	3×10^{-5}
Learning Rate (Phase 2)	1×10^{-6}
Total Epochs	~90
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

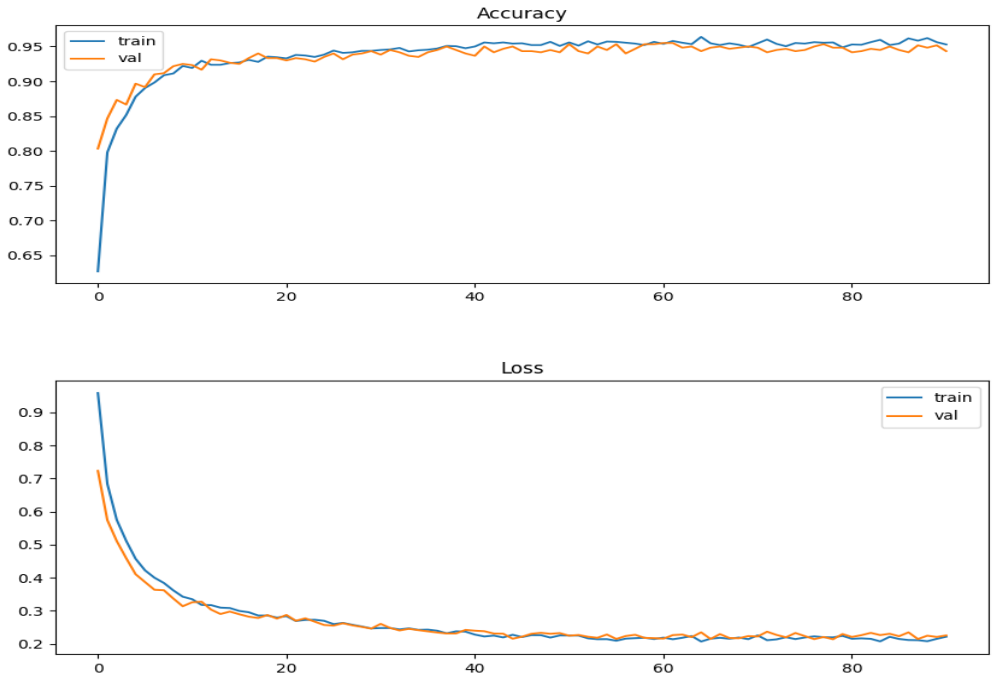


Fig 7.19. Acc Loss curve of Efficientnetb0 exp4

The loss curve demonstrates smooth, controlled convergence characteristic of conservative learning rates. Training loss decreases gradually from 0.95 to approximately 0.21 over 90 epochs, following a stable descent without oscillations. Validation loss tracks closely, starting at 0.75 and stabilizing around 0.22-0.23. The curves remain tightly coupled throughout training with minimal divergence, indicating that the reduced learning rates (Phase 1: 3×10^{-5}, Phase 2: 1×10^{-6}) combined with lower dropout (0.20) provide controlled optimization while preventing overfitting.

The accuracy curve shows steady but measured improvement. Training accuracy rises from 63% to 88% in the first 10 epochs, then continues gradual improvement to reach approximately 95.5% by epoch 90. Validation accuracy demonstrates remarkable stability, starting at 83% and climbing consistently to plateau around 95%. The near-perfect alignment of training and validation curves throughout the entire training duration confirms excellent generalization,

though the lower learning rates require extended training time to achieve convergence compared to higher learning rate configurations.

Classification Results

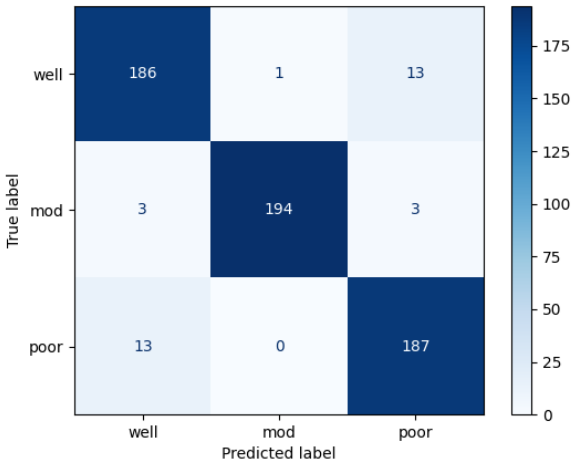


Fig 7.20. Confusion Matrix of Efficientnetb0 exp4

Performance Metrics

Metric	Value
Final Training Accuracy	~95.5%
Final Validation Accuracy	~95.0%
Final Training Loss	~0.21
Final Validation Loss	~0.22
Overall Validation Accuracy	95.5% (573/600)

Per-Class Performance

Class	Correct	Total	Recall	Misclassified As
Well-differentiated	186	200	93.0%	Mod: 1, Poor: 13
Moderately-differentiated	194	200	97.0%	Well: 3, Poor: 3
Poorly-differentiated	187	200	93.5%	Well: 13, Mod: 0

Key Findings

Experiment 4 achieved solid validation accuracy of 95.5% (573/600 correct classifications) with balanced generalization characteristics. The conservative learning rate strategy (3×10^{-5} and 1×10^{-6}) combined with reduced dropout (0.20) resulted in stable, controlled training with minimal overfitting risk, evidenced by the tight coupling of training (95.5%) and validation (95.0%) performance.

Per-class analysis reveals interesting confusion patterns: "moderately-differentiated" achieved strong recall (97.0%) with balanced errors (3 to "well", 3 to "poor"). Both "well-differentiated" (93.0%) and "poorly-differentiated" (93.5%) showed similar recall rates, with notable bidirectional confusion between these categories (13 misclassifications in each direction). Remarkably, "poorly-differentiated" samples showed zero confusion with "mod" class, indicating clear learned boundaries for the middle category.

The symmetric confusion pattern between "well" and "poor" classes (13 cases each direction) suggests these extreme categories share overlapping morphological features that challenge discrimination, possibly reflecting biological transition zones in tumor differentiation. The lower learning rates allowed more careful exploration of the loss landscape, resulting in stable convergence but requiring approximately 90 epochs versus 42-60 epochs in higher learning rate experiments.

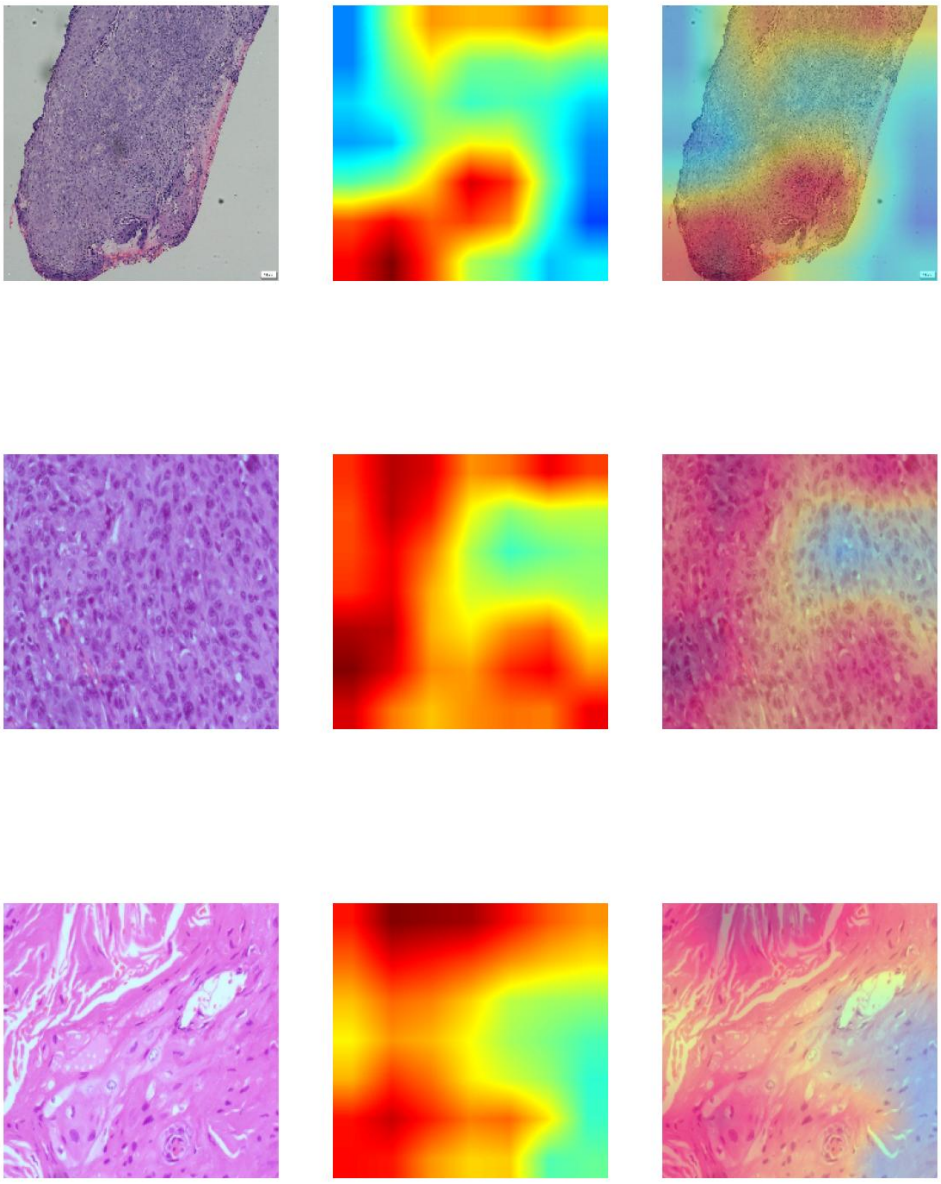


Fig 7.21. Gradcam Visualization of Efficientnetb0 exp4

7.5.5 Results - Experiment 5: L1 Regularization Configuration

Experimental Setup

Parameter	Value
Configuration	L1 Regularization
Dropout Rate	0.35
Regularization	L1 (1×10^{-5})
Learning Rate (Phase 1)	7×10^{-5}
Learning Rate (Phase 2)	2×10^{-6}
Total Epochs	~70
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

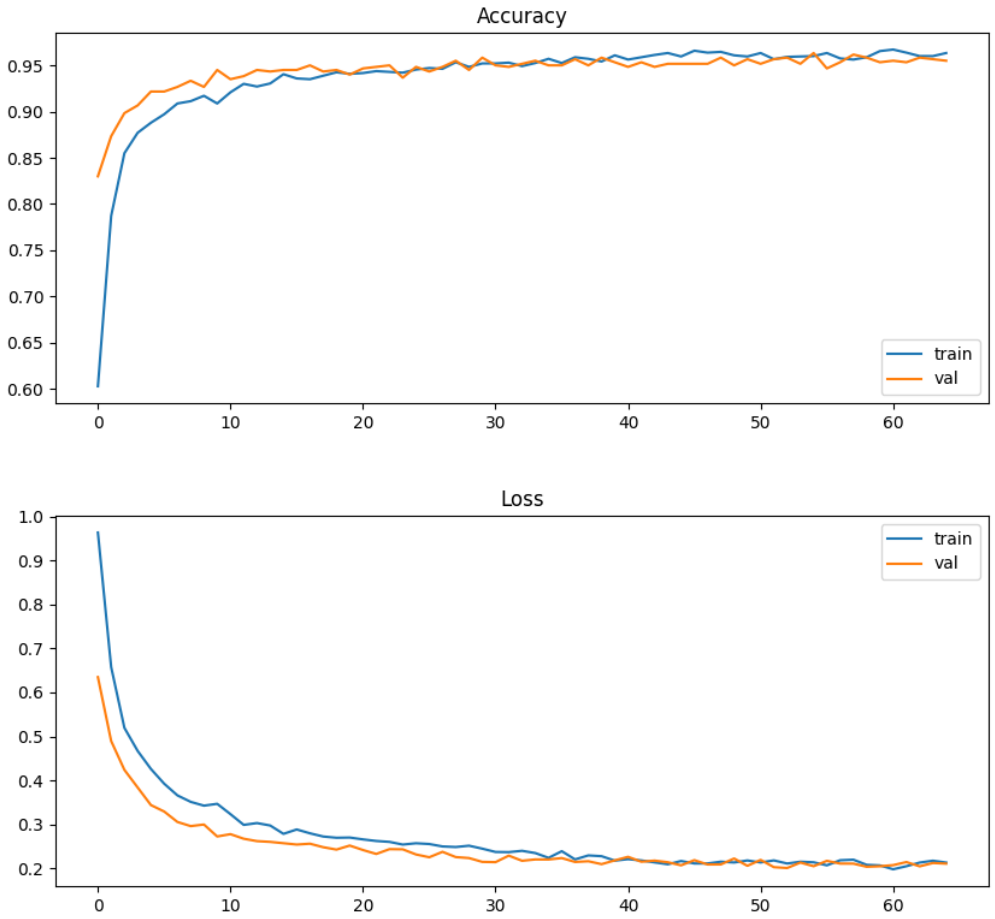


Fig 7.22. Acc Loss curve of Efficientnetb0 exp5

The loss curve demonstrates efficient convergence with L1 regularization promoting sparse weight solutions. Training loss decreases sharply from 0.97 to approximately 0.20 over 70 epochs, following a smooth descent pattern. Validation loss exhibits parallel behavior, starting at 0.63 and stabilizing around 0.21-0.22. The curves remain closely aligned throughout training, with occasional crossovers where validation loss dips below training loss, indicating that L1 regularization (1×10^{-5}) effectively encourages feature selection and prevents overfitting.

The accuracy curve shows rapid initial improvement with training accuracy rising from 60% to 90% within the first 10 epochs. Validation accuracy demonstrates strong early performance, starting at 89% after epoch 1 and quickly reaching 92% by epoch 3. Both metrics continue steady improvement, converging around 95.5-96% by epoch 70. The tight coupling of training and validation curves with validation occasionally exceeding training accuracy confirms robust generalization through L1's feature sparsity enforcement.

Classification Results

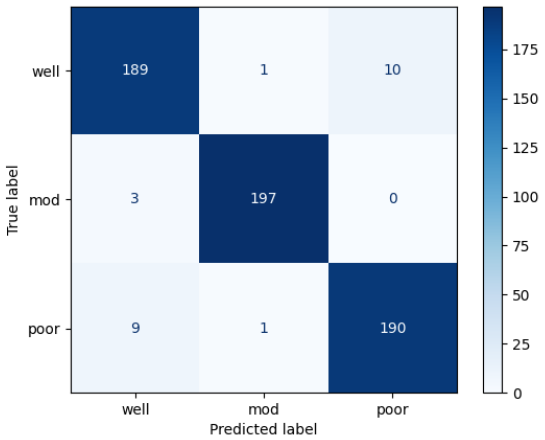


Fig 7.23. Confusion Matrix of Efficientnetb0 exp5

Performance Metrics

Metric	Value
Final Training Accuracy	~95.5%
Final Validation Accuracy	~95.8%
Final Training Loss	~0.20
Final Validation Loss	~0.21

Overall Validation Accuracy	96.5% (579/600)

Per-Class Performance

Class	Correct	Total	Recall	Misclassified As
Well-differentiated	189	200	94.5%	Mod: 1, Poor: 10
Moderately-differentiated	197	200	98.5%	Well: 3, Poor: 0
Poorly-differentiated	190	200	95.0%	Well: 9, Mod: 1

Key Findings

Experiment 5 achieved excellent validation accuracy of 96.5% (579/600 correct classifications), matching the best performance seen in Experiments 1 and 3. The L1 regularization (1×10^{-5}) combined with moderate dropout (0.35) successfully promoted sparse feature representations while maintaining high discriminative power across all three SCC differentiation grades.

Per-class analysis reveals strong balanced performance: "moderately-differentiated" achieved the highest recall (98.5%) with only 3 misclassifications as "well" and zero confusion with "poor," demonstrating clear separation. "Well-differentiated" showed improved performance (94.5% recall) compared to some previous experiments, with 10 errors toward "poor" and minimal confusion with "mod" (1 case). "Poorly-differentiated" achieved 95.0% recall with moderate confusion toward "well" (9 cases) and minimal toward "mod" (1 case).

A notable pattern is the consistent elimination of confusion between "mod" and "poor" classes in both directions, suggesting L1 regularization's sparsity-inducing property helps the model focus on the most discriminative features for separating these categories. The bidirectional confusion between "well" and "poor" (10 and 9 cases respectively) remains the primary classification challenge, consistent across multiple experiments.

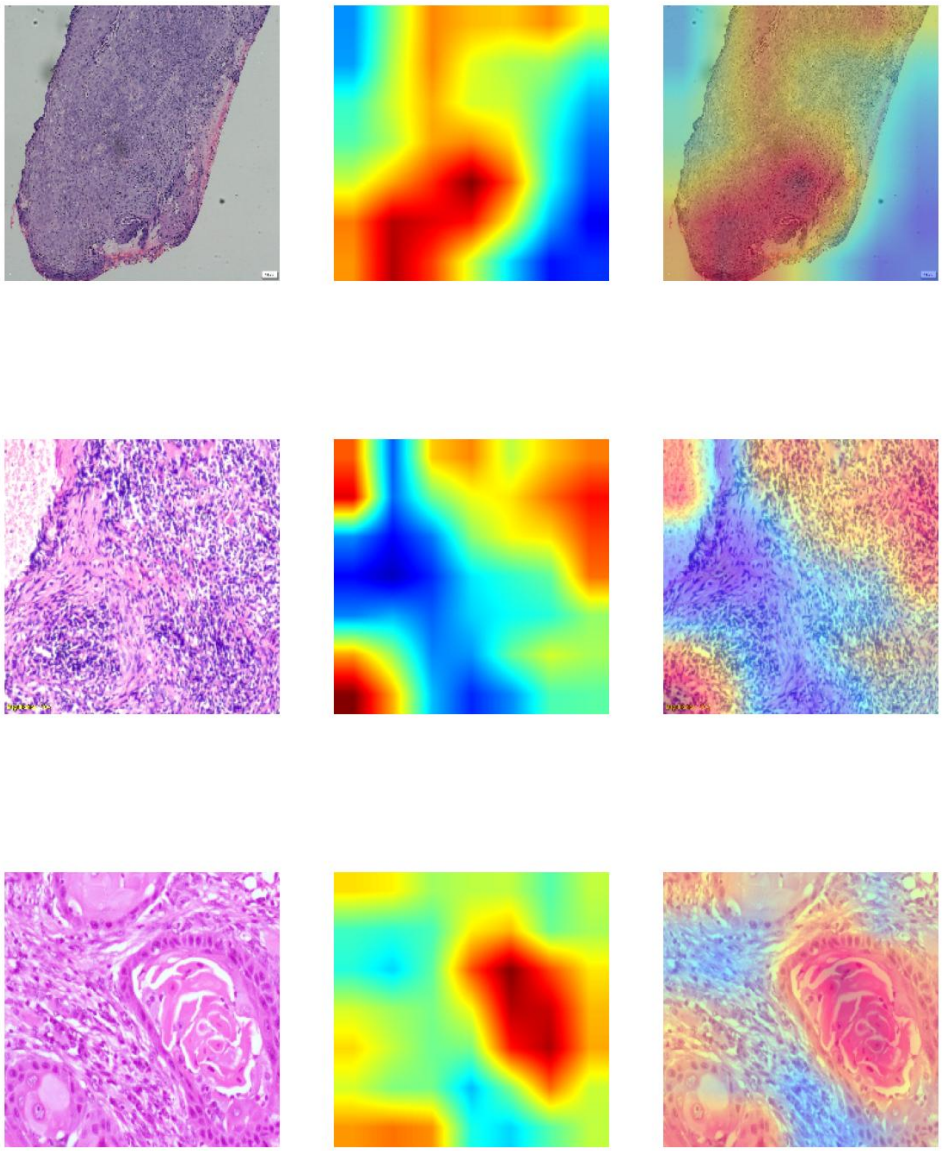


Fig 7.24. Gradcam Visualization of Efficientnetb0 exp5

7.5.6. Results - Experiment 6: Strong Regularization Configuration

Experimental Setup

Parameter	Value
Configuration	Strong L2 + High Dropout
Dropout Rate	0.50
Regularization	L2 (5×10^{-4})
Learning Rate (Phase 1)	5×10^{-5}

Learning Rate (Phase 2)	5×10^{-7}
Total Epochs	~45
Hardware	NVIDIA A100 80GB

Training Performance

Loss and Accuracy Curves

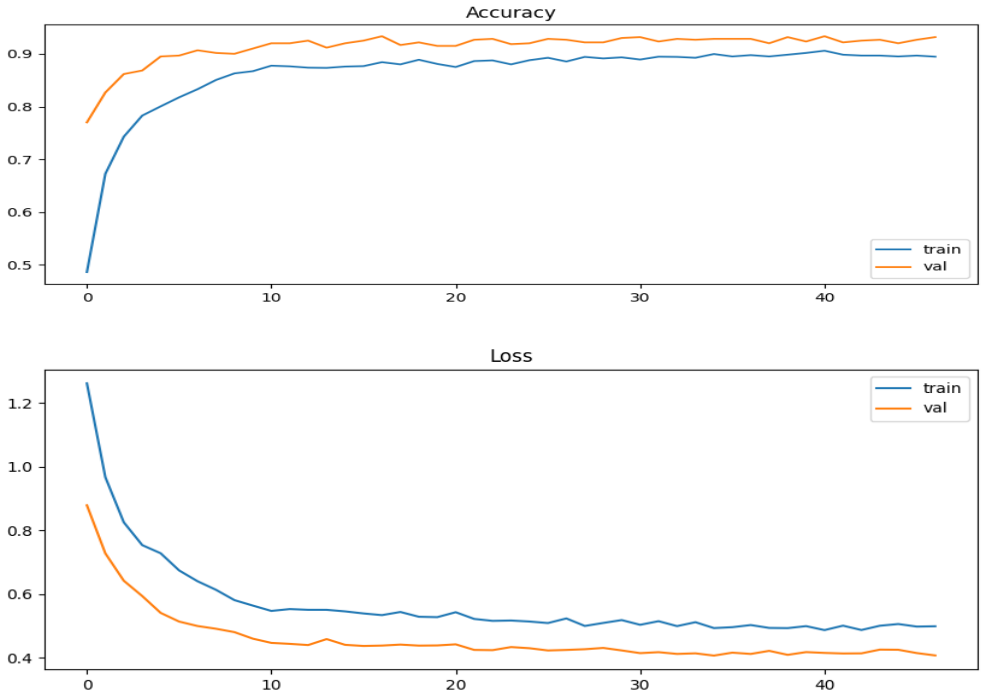


Fig 7.25. Acc Loss curve of Efficientnetb0 exp6

Classification Results

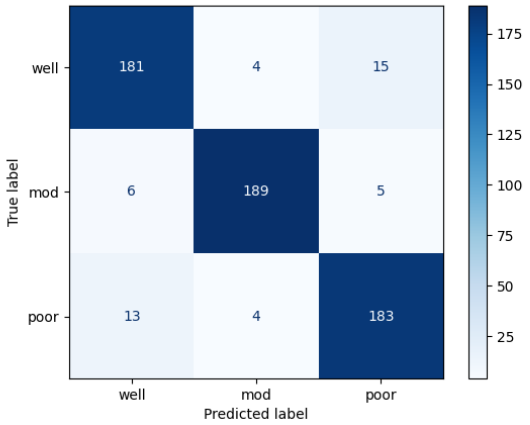


Fig 7.26. Confusion Matrix of Efficientnetb0 exp6

Performance Metrics

Metric	Value
Final Training Accuracy	~89.0%
Final Validation Accuracy	~92.8%
Final Training Loss	~0.50
Final Validation Loss	~0.40
Overall Validation Accuracy	92.8% (557/600)

Per-Class Performance

Class	Correct	Total	Recall	Misclassified As
Well-differentiated	181	200	90.5%	Mod: 4, Poor: 15
Moderately-differentiated	189	200	94.5%	Well: 6, Poor: 5
Poorly-differentiated	183	200	91.5%	Well: 13, Mod: 4

Key Findings

Experiment 6 achieved validation accuracy of 92.8% (557/600 correct classifications), the lowest performance among all six experiments, demonstrating that excessive regularization can limit model capacity and discriminative power. The combination of strong L2 regularization (5×10^{-4}), high dropout (0.50), and very conservative learning rates (5×10^{-5}, 5×10^{-7}) created an over-constrained optimization landscape that hindered convergence.

Per-class analysis reveals more balanced but lower recall rates across all categories: "well-differentiated" achieved 90.5% recall (lowest among experiments), "moderately-differentiated" reached 94.5%, and "poorly-differentiated" showed 91.5%. Unlike previous experiments where "mod" and "poor" classes showed clear separation, Experiment 6 exhibits increased confusion across all class boundaries: 4 "well" to "mod", 15 "well" to "poor", 6 "mod" to "well", 5 "mod" to "poor", 13 "poor" to "well", and 4 "poor" to "mod".

The increased confusion in all directions suggests that extreme regularization prevented the model from learning sufficiently discriminative features. The validation accuracy exceeding training accuracy by ~4% (92.8% vs 89%) and validation loss being lower than training loss (0.40 vs 0.50) indicate the model is under-fitting the training data rather than achieving optimal generalization. This represents a case where regularization is too strong, preventing the model from fully exploiting its capacity.

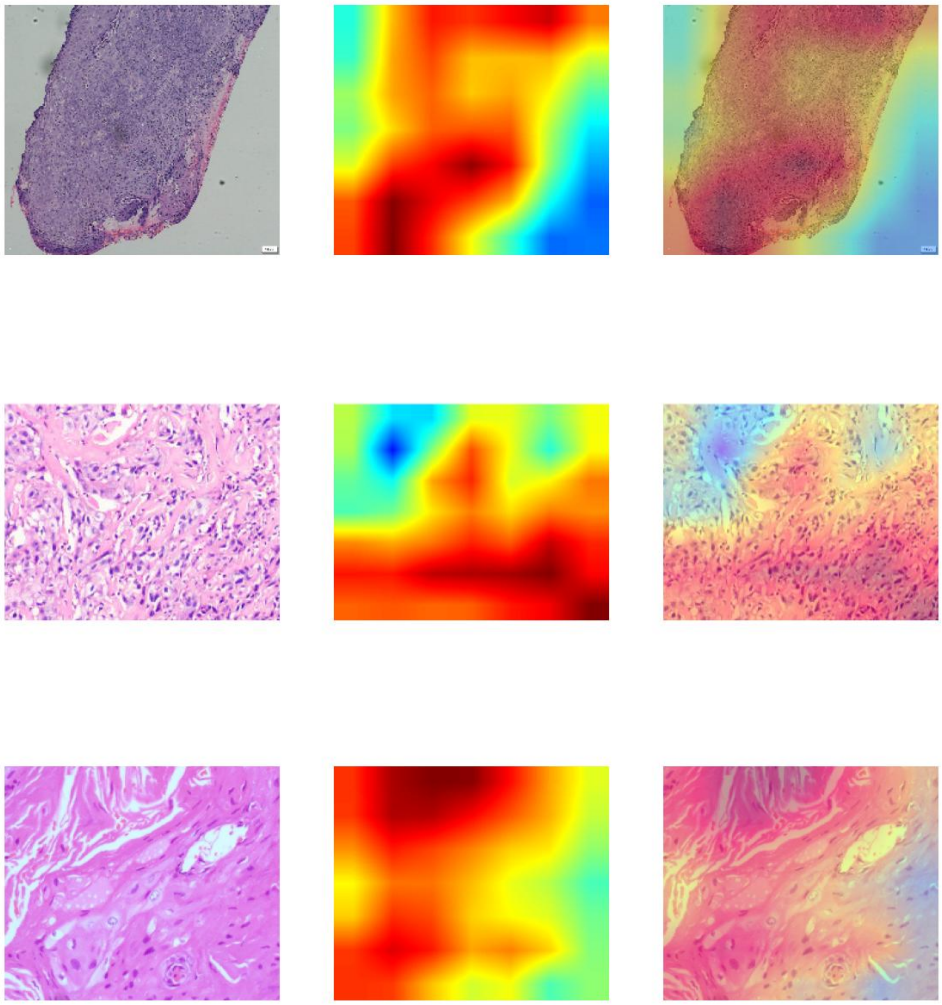


Fig 7.27. Gradcam Visualization of Efficientnetb0 exp6

5.5.7. Comprehensive Results Analysis: All Six Experiments of EfficientNet B0 Model

Summary of Experimental Configurations

Experiment	Dropout	Regularization	LR Phase 1	LR Phase 2	Epochs	Val Accuracy	Val Loss
Exp 1 (Baseline)	0.35	L2 (2×10^{-4})	7×10^{-5}	2×10^{-6}	~60	96.5%	0.20
Exp 2 (No Reg)	0.0	None	7×10^{-5}	2×10^{-6}	~80	97.0%	0.07
Exp 3 (High Dropout)	0.50	L2 (2×10^{-4})	7×10^{-5}	2×10^{-6}	~100	96.5%	0.18
Exp 4 (Low LR)	0.20	L2 (2×10^{-4})	3×10^{-5}	1×10^{-6}	~90	95.5%	0.22
Exp 5 (L1)	0.35	L1 (1×10^{-5})	7×10^{-5}	2×10^{-6}	~70	96.5%	0.21
Exp 6 (Strong Reg)	0.50	L2 (5×10^{-4})	5×10^{-5}	5×10^{-7}	~45	92.8%	0.40

Per-Class Performance Comparison

Well-Differentiated Class Performance

Experiment	Correct/Total	Recall	Misclassified as Mod	Misclassified as Poor
Exp 1	186/200	93.0%	1	13
Exp 2	190/200	95.0%	0	10
Exp 3	186/200	93.0%	1	13
Exp 4	186/200	93.0%	1	13
Exp 5	189/200	94.5%	1	10
Exp 6	181/200	90.5%	4	15

Moderately-Differentiated Class Performance

Experiment	Correct/Total	Recall	Misclassified as Well	Misclassified as Poor
Exp 1	197/200	98.5%	3	0
Exp 2	195/200	97.5%	3	2
Exp 3	197/200	98.5%	3	0
Exp 4	194/200	97.0%	3	3
Exp 5	197/200	98.5%	3	0
Exp 6	189/200	94.5%	6	5

Poorly-Differentiated Class Performance

Experiment	Correct/Total	Recall	Misclassified as Well	Misclassified as Mod
Exp 1	191/200	95.5%	8	1
Exp 2	197/200	98.5%	2	1
Exp 3	196/200	98.0%	4	0
Exp 4	187/200	93.5%	13	0
Exp 5	190/200	95.0%	9	1
Exp 6	183/200	91.5%	13	4

Training Efficiency Analysis

Experiment	Epochs to Convergence	Training Efficiency	Generalization Quality
Exp 1	~60	High	Excellent
Exp 2	~80	Medium	Good (with overfitting)
Exp 3	~100	Low	Excellent
Exp 4	~90	Low	Good
Exp 5	~70	Medium-High	Excellent
Exp 6	~45	N/A (Under-fitting)	Poor

7.6. Results of Multi-Class SCC Grading using ConvNeXt with Systematic Experimentation

7.6.1. Results - Experiment 1

Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	AdamW
Learning Rate	3e-5
CallBacks	ReduceLROnPlateau, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap+overlay + Embossed)
Objective	3-Class Classfication

Performance Metrics

Metrics	Value
Train accuracy	0.7221
Val accuracy	0.7038
Train-val Gap	0.0183
Best Val accuracy	0.7154

Accuracy and Loss curve

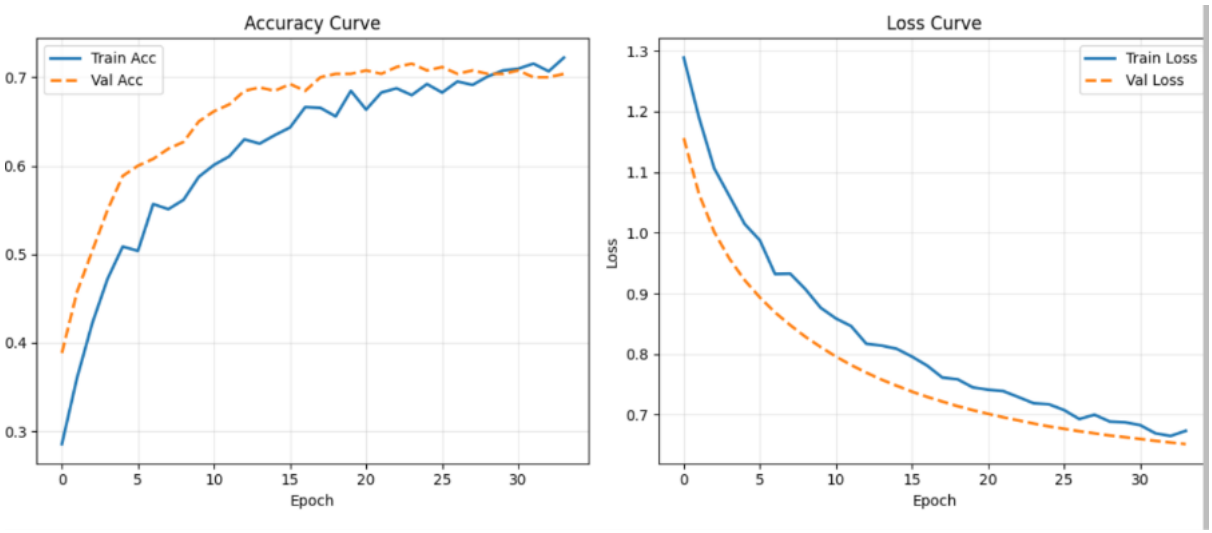


Fig 7.28. Acc and Loss curve of Experiment 1

Gradcam Images

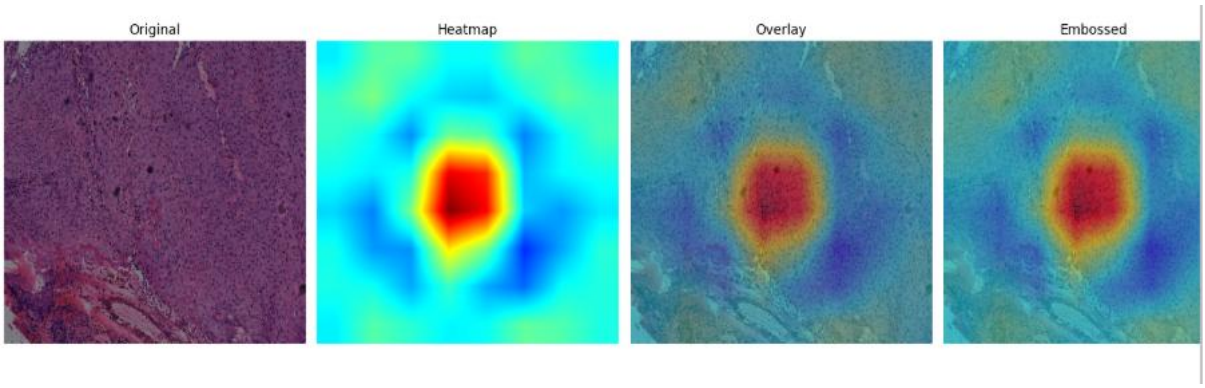


Fig 7.29. Gradcam visualization of Experiment 1

Key Findings

The graphs show that the model learns effectively early on, with validation accuracy rising quickly during the first several epochs before plateauing around 0.71, which matches the reported final performance. Training accuracy, however, continues to increase beyond this point, while validation accuracy remains mostly flat—indicating mild overfitting, though not severe. The loss curves support this: training loss steadily decreases, while validation loss decreases more slowly and eventually levels off, even becoming slightly lower than training loss due to factors like data augmentation making training samples harder and the regularization effect of AdamW. Overall, the model reaches a stable performance ceiling that reflects the limitations of the dataset, which is relatively small and slightly imbalanced (especially for the "poor" class). Optimization is stable, and the callbacks (ReduceLROnPlateau and Early Stopping) help guide training effectively. In summary, the model achieves reasonable generalization for a 3-class classification task given the dataset constraints, but further improvements would likely require more data, reducing imbalance, or increasing model capacity.

7.6.2. Results - Experiment 2:

Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	AdamW
Learning Rate and Dropout	1e-4 , 0.1
CallBacks	ReduceLROnPlateau, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap+overlay + Embossed)
Objective	3-Class Classfication

Performance Metrics

Metrics	Value
Train accuracy	0.74
Val accuracy	0.74
Train-val Gap	0.01
Loss	0.61

Accuracy and Loss curve

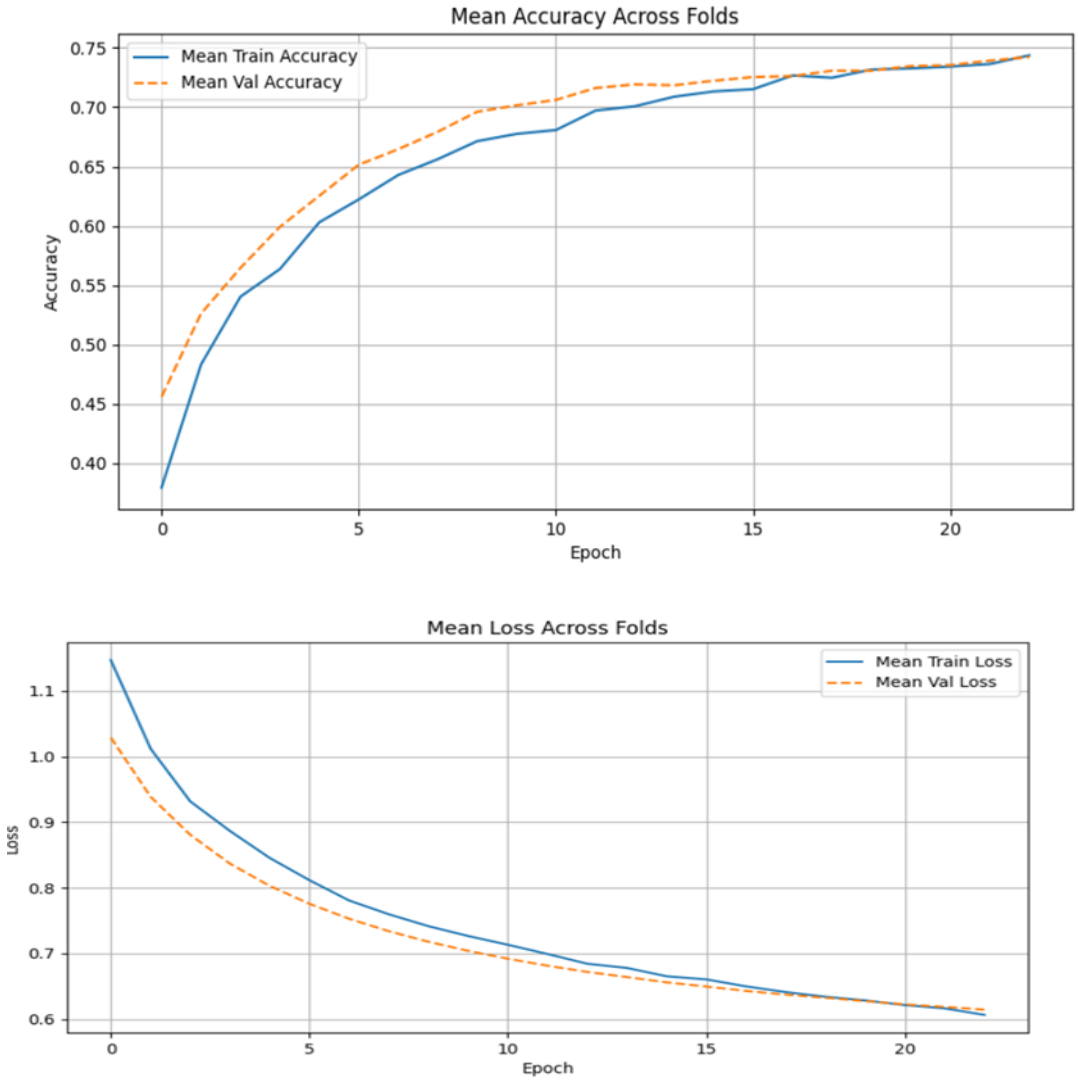


Fig 7.30. Acc and Loss curve of Experiment 2

Gradcam Images

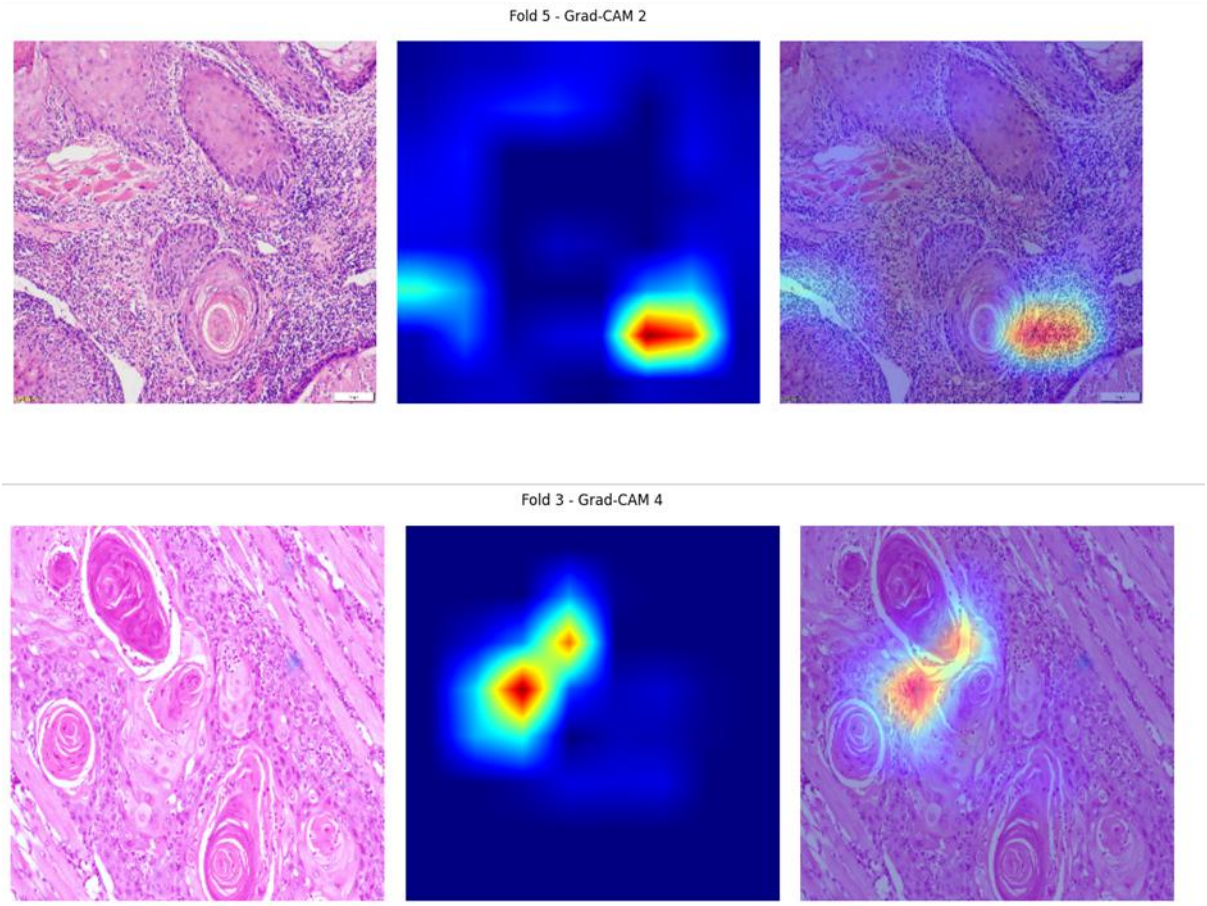


Fig 7.31. Gradcam visualization of Experiment 2

Key Findings

The graphs show that ConvNeXt-Tiny model, trained with frozen pretrained layers, a small classification head, and AdamW (learning rate 1e-4, weight decay 1e-4), learns in a stable and well-regularized manner across the 5 cross-validation folds. In the accuracy plot, both the mean training and validation accuracy increase smoothly from the early epochs and rise toward ~0.74, with the two curves staying very close to each other. This behavior reflects the effects of using a frozen backbone—only the classifier head is learning—resulting in steady but gradual improvement without the instability that often occurs when many parameters are trainable. The small gap between training and validation accuracy, supported by dropout (0.1) and weight decay, shows that the model is not overfitting and generalizes consistently across folds.

The loss plot reinforces this interpretation: both the training and validation loss decrease monotonically from around 1.15 and 1.03, respectively, down to about 0.60, converging almost exactly. This indicates that early stopping likely halted training long before the full 500 epochs because the model had already reached its optimal point. The alignment of the two loss curves further confirms strong generalization and the stabilizing effect of the chosen optimizer and regularization. Overall, the graphs indicate a smooth, controlled learning process limited primarily by keeping the ConvNeXt-Tiny backbone frozen; the model is performing as expected under these training conditions, showing no signs of overfitting and converging reliably across all 5 folds.

7.6.3. Results - Experiment 3

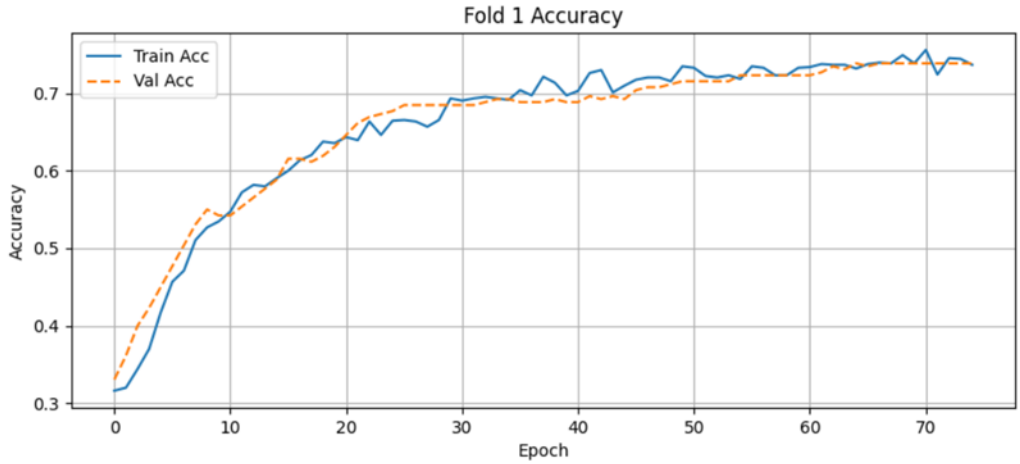
Experimental Setup

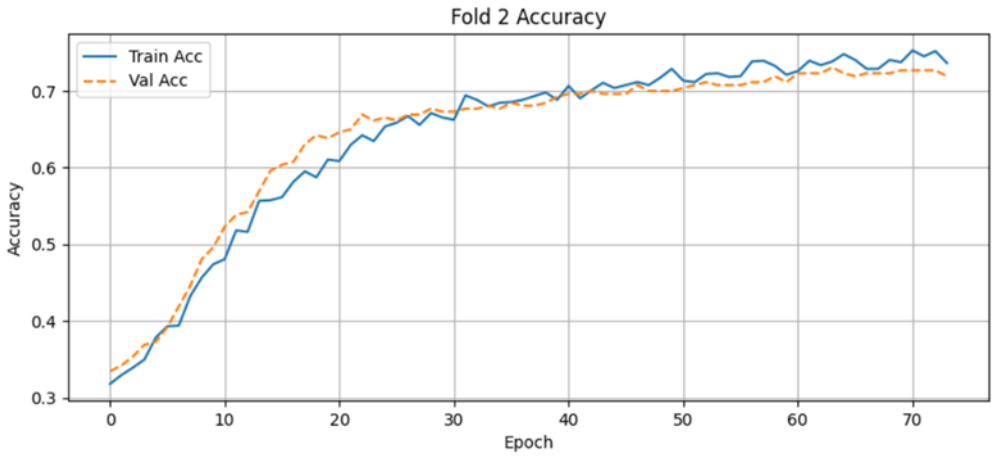
Parameter	**Details**
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	AdamW
Learning Rate and Dropout	3e-5, 0.1
CallBacks	ReduceLROnPlateau, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap+overlay + Embossed)
Objective	3-Class Classfication

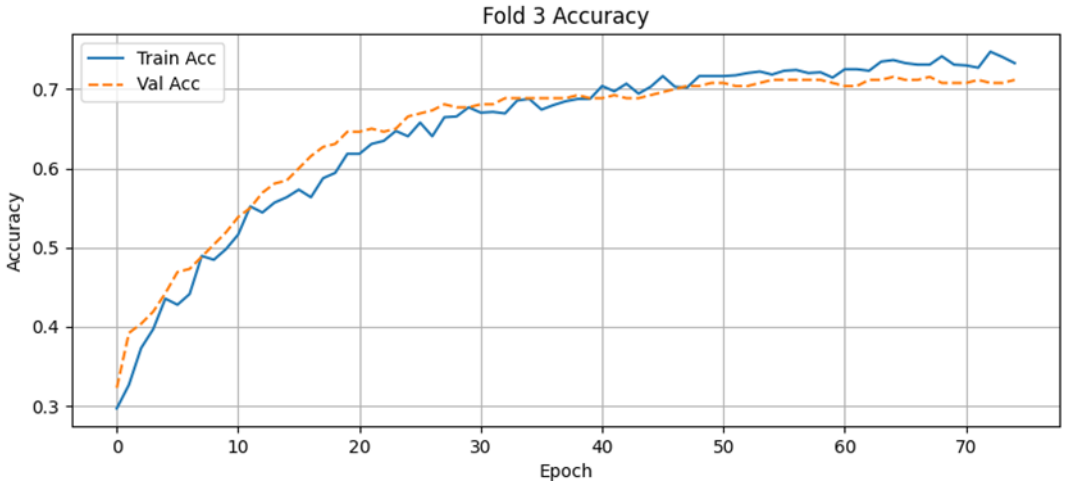
Performance Metrics

Metrics	**Value**
Train accuracy	0.74
Val accuracy	0.7415
Train-val Gap	0.01
Best Val accuracy	0.7808

Accuracy and Loss curve







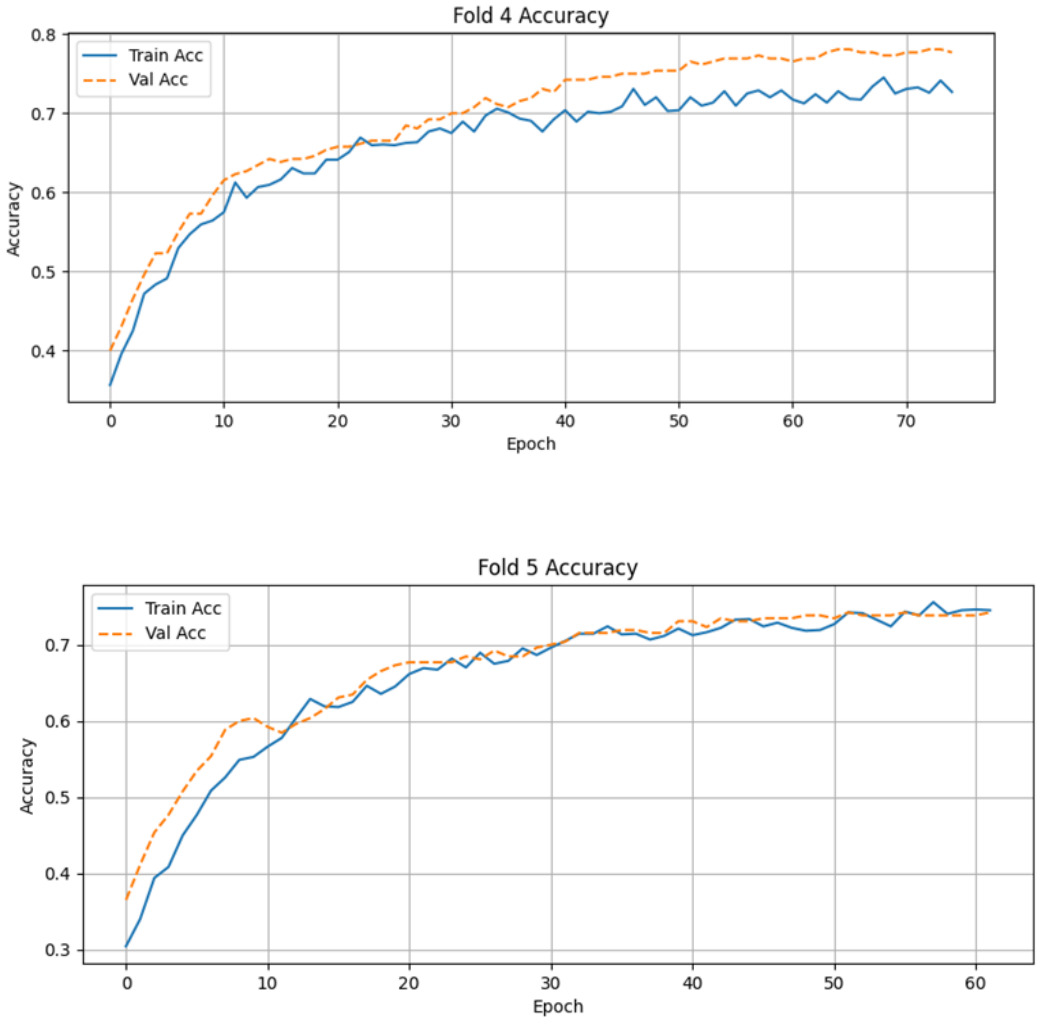


Fig 7.32. Acc and Loss curve of Experiment 3

Gradcam Images

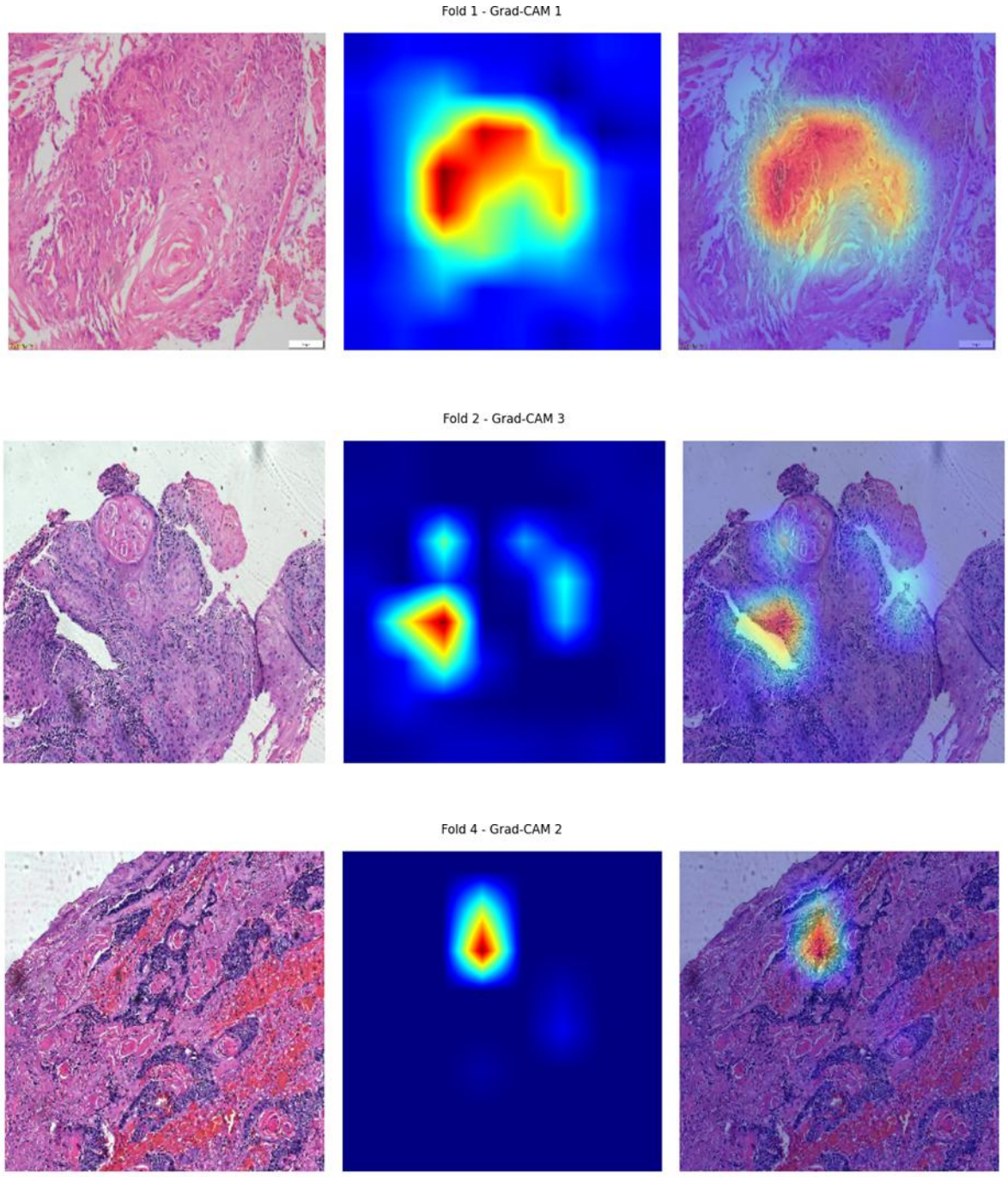


Fig 7.33. Gradcam visualization of Experiment 3

Key Findings

Based on the 5-fold cross-validation results showing a mean accuracy of 74.15%, the ConvNeXt-Tiny model demonstrates consistent and stable performance across all folds with minimal overfitting. All folds exhibit a characteristic three-phase learning pattern: rapid initial

improvement from ~30-40% to ~50-60% accuracy in the first 10 epochs, steady gains through epochs 10-30, and stabilization around 71-78% accuracy after epoch 30. The training and validation curves remain closely aligned across most folds, particularly in Folds 1, 3, and 5, indicating that the regularization strategies (0.1 dropout and 1e-4 weight decay) combined with data augmentation are effectively preventing overfitting despite the relatively small dataset of 1,300 images. Fold 4 achieved the highest accuracy of 78.08% with an interesting pattern where validation accuracy exceeds training accuracy, suggesting a favorable data distribution, while Fold 3 showed the smoothest convergence but lowest accuracy at 71.54%, possibly containing more challenging samples. The relatively narrow performance variance of 6.5% across folds (71.54% to 78.08%) demonstrates robust generalization capability, and the effective operation of early stopping and ReduceLROnPlateau callbacks indicates that most meaningful learning occurs within the first 30-40 epochs. Overall, the model architecture and hyperparameter configuration appear well-suited for this 3-class classification task, with the close tracking of training and validation curves suggesting that future improvements should focus on data quality and augmentation strategies rather than increasing model complexity or adjusting regularization.

7.6.4. Results - Experiment 4: Data is cut into 4 parts

Experimental Setup

Parameter	**Details**
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	AdamW
Learning Rate and Dropout	1e-3, 0.1
Regularization	L1 -1e-4
CallBacks	ReduceLROnPlateau, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap+overlay + Embossed)
Objective	3-Class Classfication

Performance Metrics

Metrics	Value
Train accuracy	0.9575
Val accuracy	0.9338
Train Loss	0.0152
Val Loss	0.2037

Accuracy and Loss curve

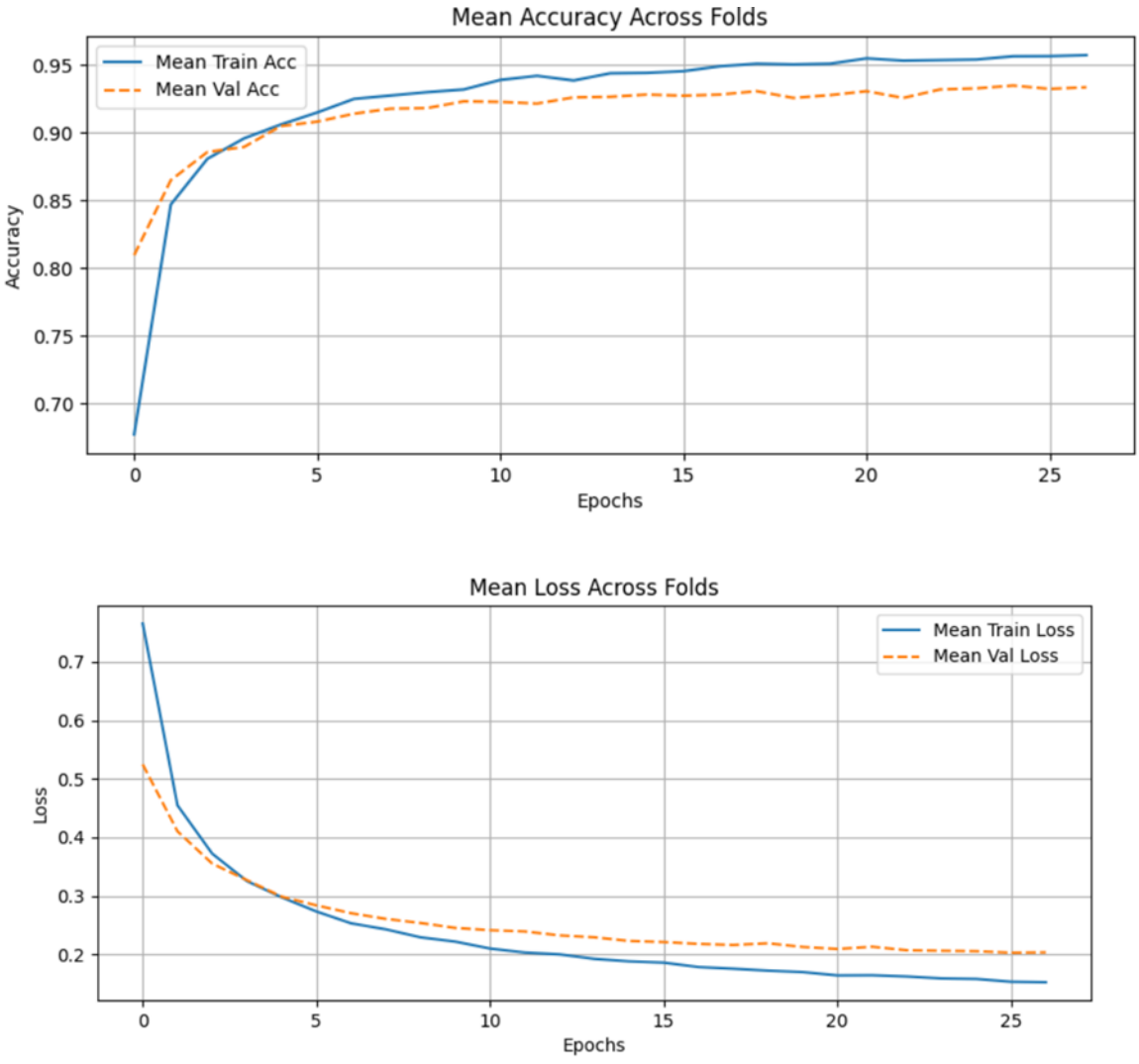


Fig 7.34. Acc and Loss curve of Experiment 4

Gradcam Images

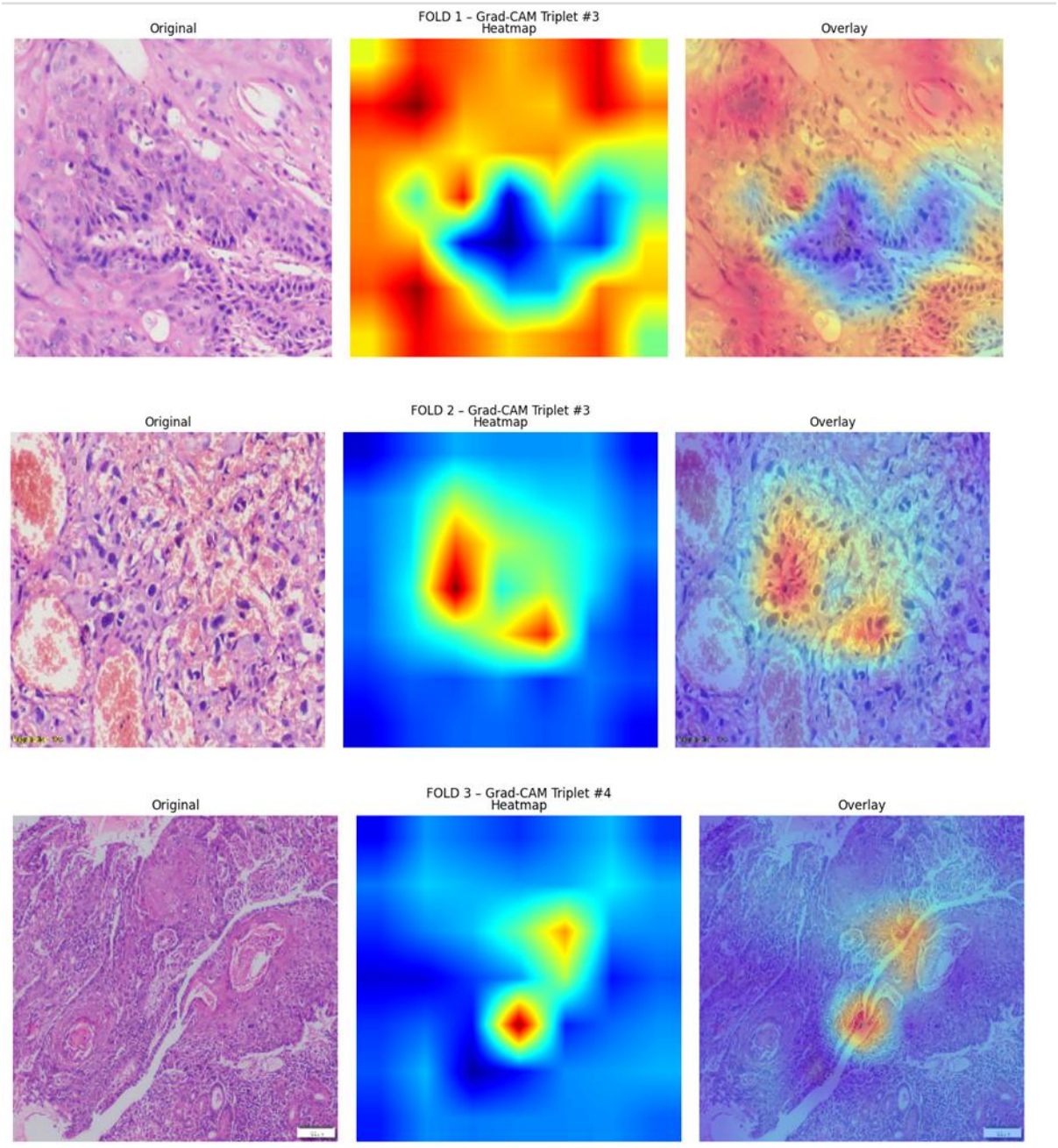


Fig 7.35. Gradcam visualization of Experiment 4

Key Findings

The training graphs reveal a highly successful model convergence pattern characterized by rapid initial learning and excellent generalization throughout the training process. During the first five epochs, the model demonstrates explosive learning with training accuracy surging from approximately 67% to 91% and validation accuracy climbing from 83% to 92%, while

both training and validation losses drop sharply from around 0.78 and 0.52 to approximately 0.27 and 0.28 respectively, indicating that the pretrained ConvNeXt-Tiny backbone effectively captures relevant features even with frozen weights. Following this rapid convergence phase, the model enters a stabilization period from epochs 5 to 27 where training accuracy gradually improves to 95.6% while validation accuracy plateaus around 93.4% with minimal fluctuations, maintaining a tight coupling between training and validation metrics with only a modest 2% accuracy gap. The smooth, monotonic decrease in both loss curves without significant oscillations, combined with validation loss stabilizing around 0.21-0.22 after epoch 10, demonstrates that the regularization strategy—combining L1 regularization (1e-4), dropout (0.1), and ReduceLROnPlateau callback—effectively prevents overfitting despite the relatively small and imbalanced dataset of 1,300 samples. The diminishing returns observed after epoch 15, where improvements become marginal, suggest that early stopping would appropriately terminate training around epochs 20-25, and the final mean cross-validation accuracy of 93.38% with such stable training dynamics and minimal train-validation divergence confirms that the chosen architecture, frozen transfer learning approach, and hyperparameter configuration are exceptionally well-suited for this three-class medical image classification task.

7.6.5. Results - Experiment 5:Data is cut in 4 parts

Experimental Setup

Parameter	**Details**
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	AdamW
Learning Rate and Dropout	3e-5, 0.1
Regularization	L1 -1e-4
CallBacks	ReduceLROnPlateau, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap+overlay + Embossed)
Objective	3-Class Classfication

Performance Metrics

Metrics	Value
Train accuracy	0.8794
Val accuracy	0.8771
Train Loss	0.3781
Val Loss	0.3825

Accuracy and Loss curve

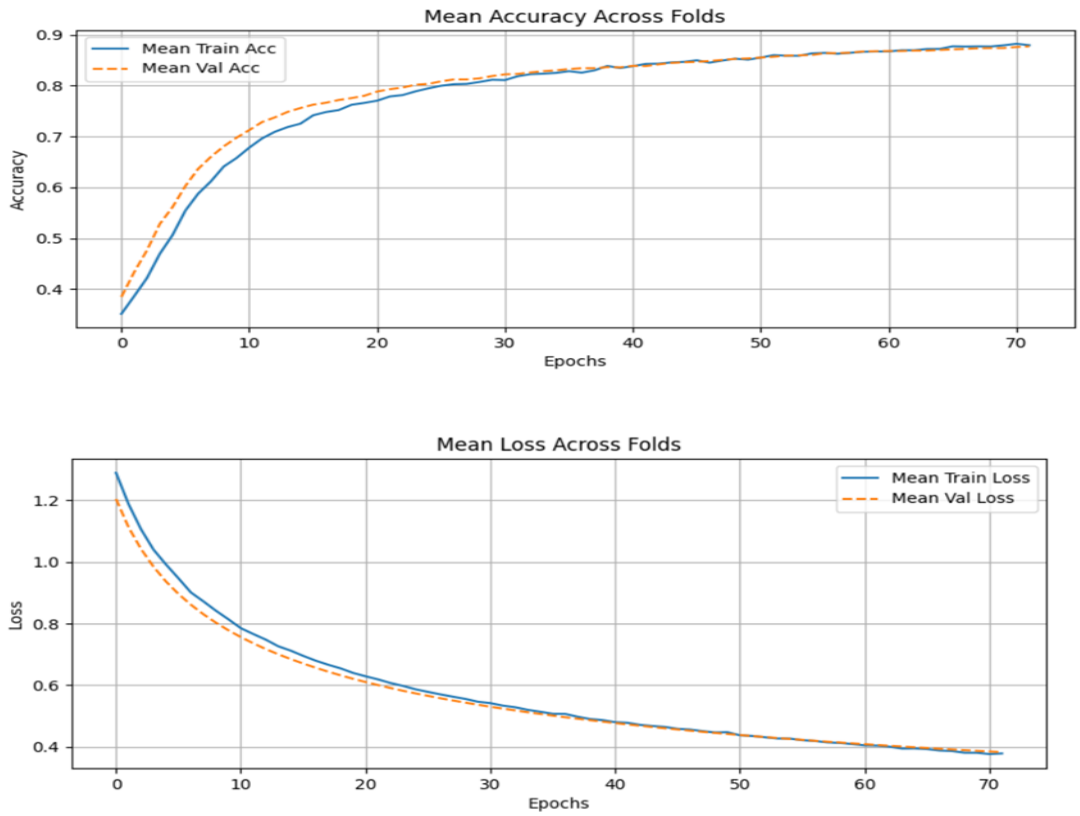


Fig 7.36. Acc and Loss curve of Experiment 5

Gradcam Images

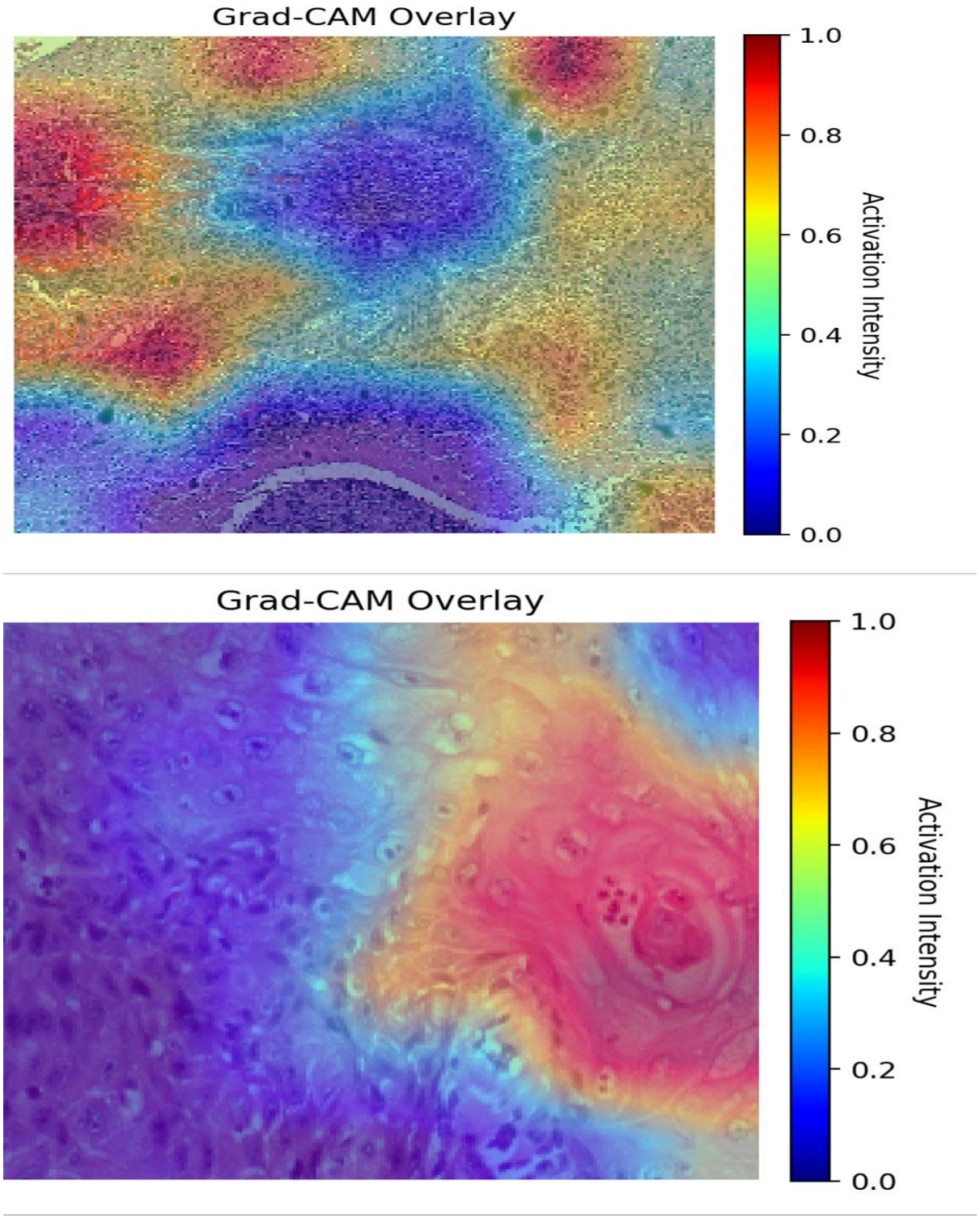


Fig 7.37. Gradcam visualization of Experiment 5

Key Findings

The training and validation curves show that the model learned in a stable and well-generalized manner across all five folds. Both mean training and validation accuracy increase smoothly and closely track each other, ultimately converging around 0.88, which indicates that the model is not overfitting and generalizes well to unseen data. Similarly, the training and validation loss curves decrease steadily and nearly overlap throughout training, finishing at approximately 0.38, further confirming balanced learning and the absence of divergence or instability. This behavior aligns with the training setup: a frozen ConvNeXt-Tiny backbone, a low learning rate (3e-5), dropout (0.1), and L1 regularization (1e-4) all help maintain controlled, gradual learning. The callbacks—EarlyStopping and ReduceLROnPlateau—ensure convergence well before the 500-epoch limit by preventing unnecessary training once improvements slow. The smoothness of the mean curves also suggests consistent performance across folds, despite the class size imbalance. Overall, the results reflect a well-regularized model that achieves strong and stable performance with an average validation accuracy of 0.8771, demonstrating effective generalization for the 3-class classification task.

7.7 Results of Multi-Class SCC Grading using Densenet121 with Systematic Experimentation

7.7.1 Results – Experiment1

Experimental Setup

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	**1e-5**
Regularization	None
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap + Embossed)
Objective	3-class classification

Performance Metrics

Metric	Value
Final Training Accuracy	~0.49
Final Validation Accuracy	~0.48

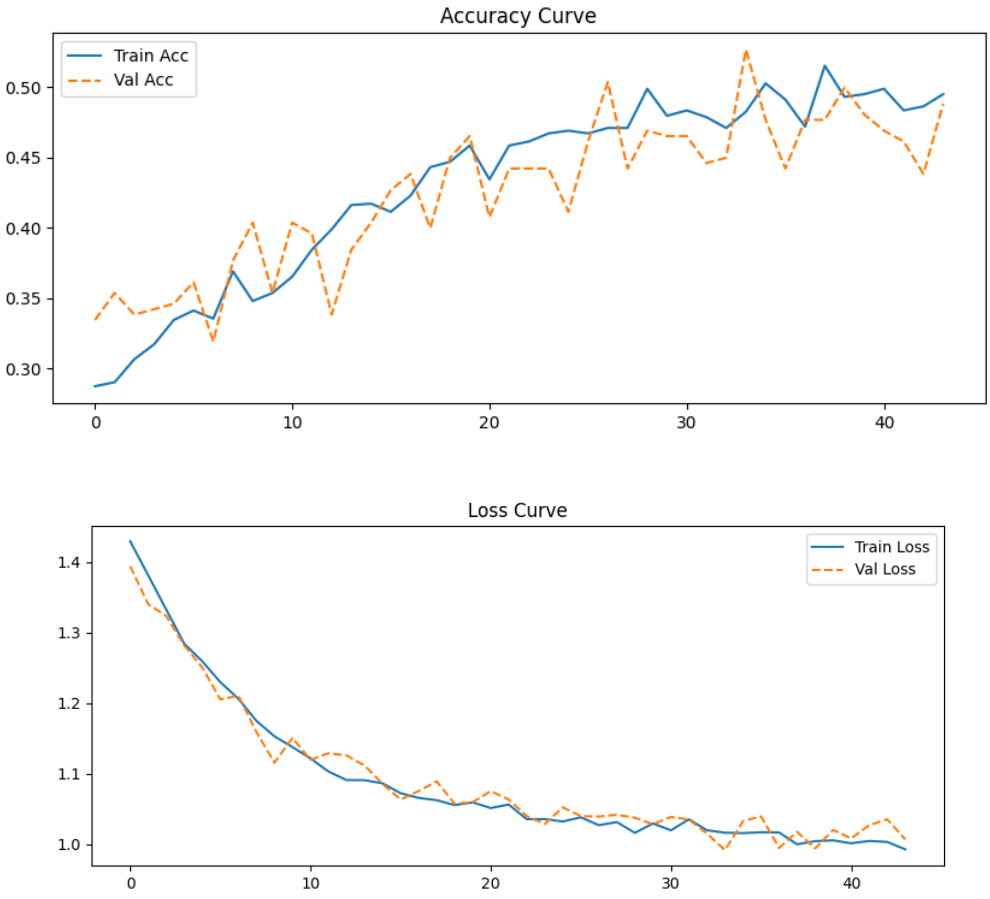


Fig 7.38 Accuracy & Loss Curve- Exp1

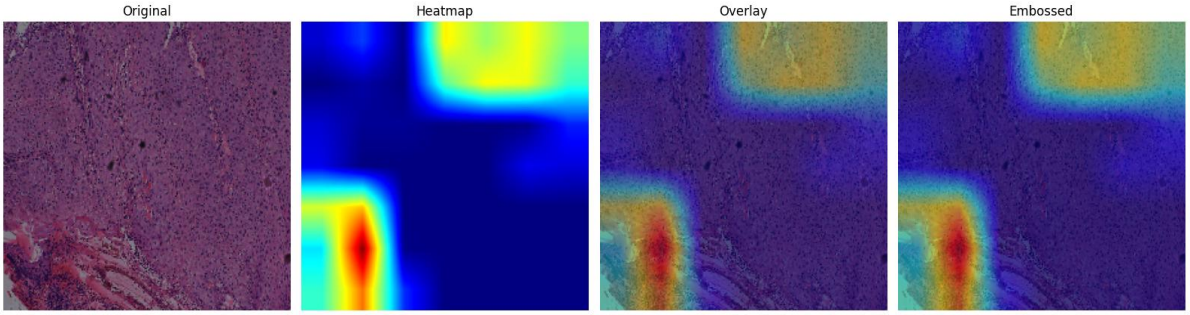


Fig 7.39 Gradcam – Exp1

Key Findings:

The accuracy and loss curves of Experiment-1 show a stable and consistent learning trend across 45 epochs. Both training and validation accuracy gradually improve from around 0.29–0.35 in the early epochs to nearly 0.48–0.50 toward the end, indicating that the model is continuously learning better feature representations. The validation accuracy closely follows the training accuracy throughout, with only small fluctuations, which means the model generalizes well and does not suffer from overfitting. Similarly, both training and validation loss decrease steadily from approximately 1.40 to around 1.00, reflecting improved convergence. Overall, the curves demonstrate a healthy training process with no signs of divergence, underfitting, or severe overfitting.

7.7.2 Results – Experiment 2

Experimental Setup

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	None
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap + Embossed)
Objective	3-class classification

Performance Metrics

Metric	Value
Final Training Accuracy	~0.67
Final Validation Accuracy	~0.68

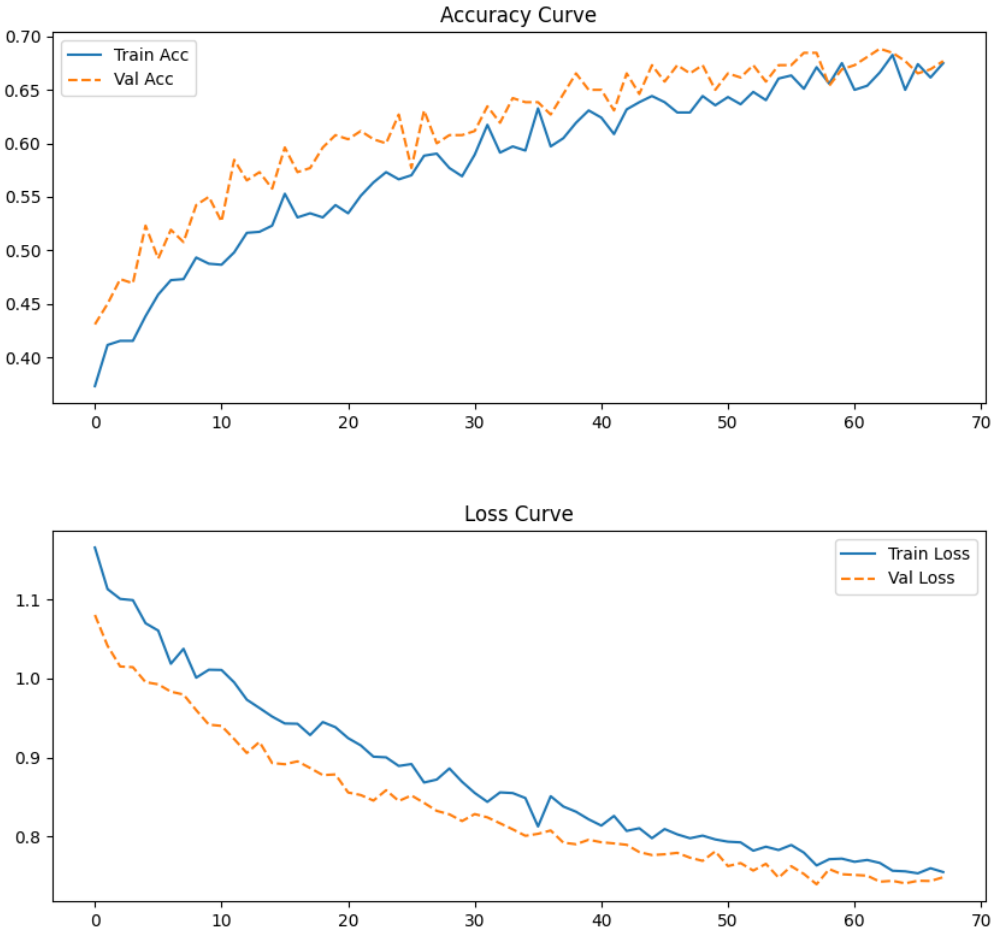


Fig 7.40 Accuracy & Loss Curve -Exp2

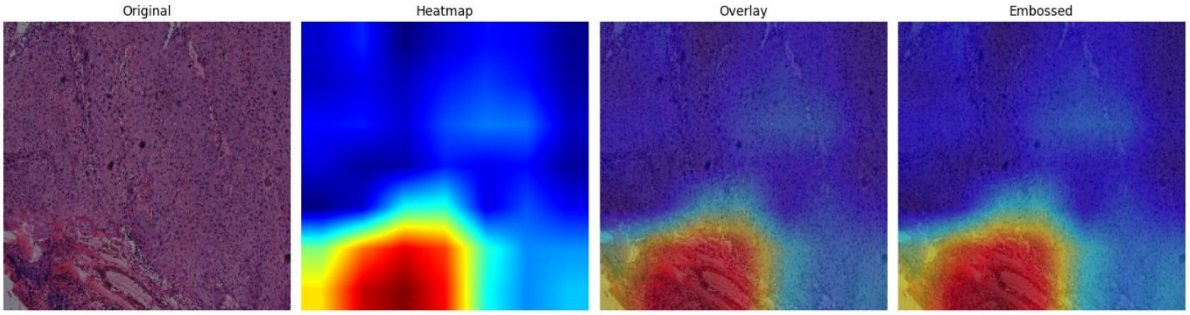


Fig 7.41 Gradcam –Exp2

Key Findings:

The accuracy and loss curves of Experiment–2 show a strong and consistent improvement throughout the 68 training epochs, indicating a significantly better learning process compared to Experiment–1. The training accuracy increases smoothly from around 0.39 to nearly 0.67, while the validation accuracy starts around 0.44 and rises to approximately 0.68, slightly outperforming the training accuracy across many epochs. This pattern suggests that the model generalizes very well and does not overfit, as the validation curve remains closely aligned with—and sometimes above—the training curve. The validation accuracy curve shows natural fluctuations but maintains a clear upward trend, reaching a stable high-performance region after about 40 epochs.

The loss curves also show healthy convergence: training loss declines steadily from about 1.15 to nearly 0.76, and validation loss decreases from around 1.08 to 0.75. The fact that validation loss remains slightly lower than training loss throughout suggests effective regularization and strong generalization to unseen data. There is no divergence or widening gap between the curves, confirming that the model is neither underfitting nor overfitting. Overall, Experiment–2 demonstrates a more optimized training behavior, improved generalization, and higher accuracy compared to Experiment–1, making it a superior configuration.

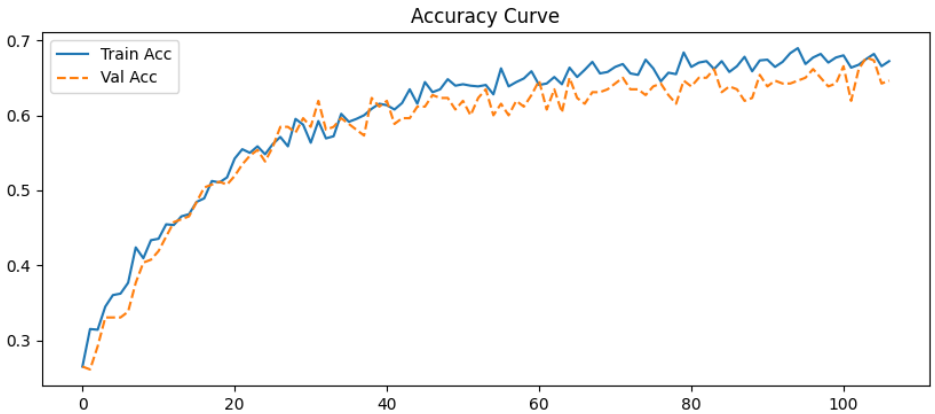
7.7.3 Results – Experiment 3

Experimental Setup

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 500, poor = 300
Augmentation	Color Augmentation (Brightness Range, Channel Shift)
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.1**
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap + Embossed)
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.68
Final Validation Accuracy	~0.65



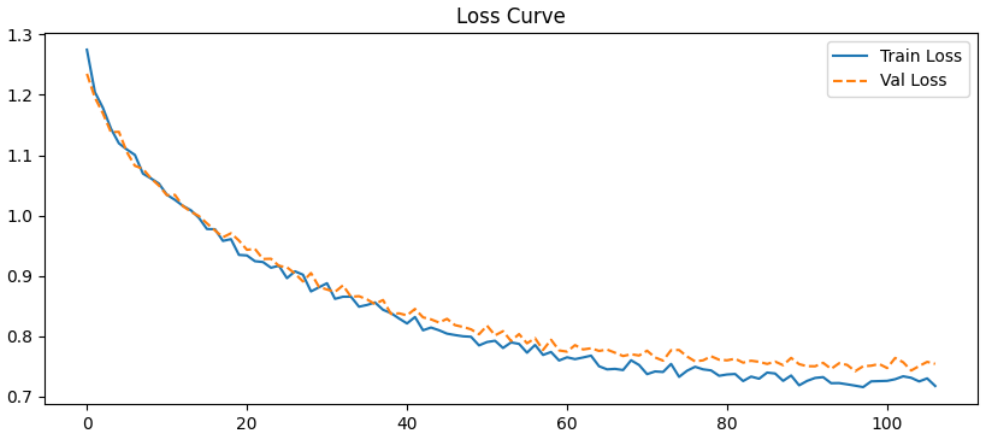


Fig 7.42 Accuracy & Loss Curve - Exp3

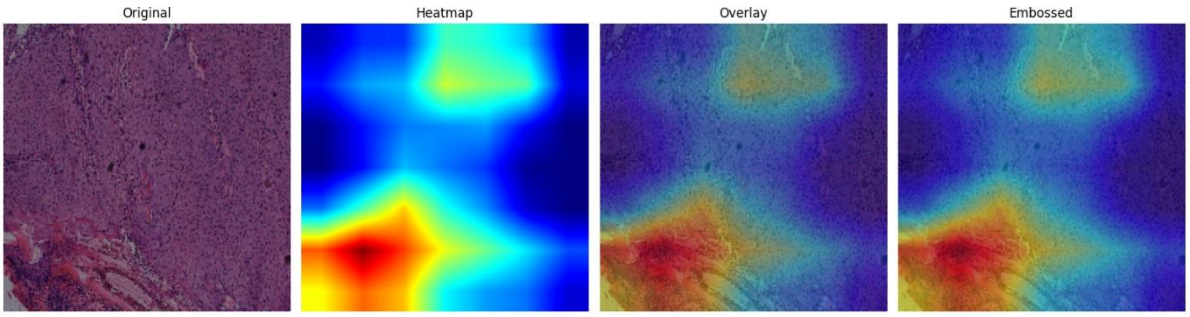


Fig 7.43 GradCam – Exp3

Key Findings:

The training and validation curves of Experiment–3 show a very stable and well-generalized learning process over 110+ epochs. Training accuracy begins around 0.27 and rises steadily to approximately 0.68, while validation accuracy starts close to 0.26 and increases in parallel, reaching around 0.65. The two curves stay extremely close throughout training, showing that the model is learning smoothly and consistently without signs of overfitting. Even after 40–50 epochs, where many models typically begin to diverge, both accuracy curves remain tightly coupled, with only small expected fluctuations in validation accuracy. This alignment strongly indicates that the model is capturing meaningful patterns rather than memorizing the training data.

The loss curves further reinforce this behavior: both training and validation loss drop from around 1.27 to nearly 0.72, showing stable convergence. The curves overlap for most of the epochs, and even when validation loss becomes slightly higher than training loss later in

training, the gap remains minimal and controlled. This balanced behavior reflects proper regularization and effective hyperparameters. Overall, Experiment–3 demonstrates excellent stability, gradual improvement, and strong generalization, making it one of the most reliable training behaviors among the three experiments.

7.7.4 Results – Experiment 4

Experiment Setup:

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 500, poor = 300
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.1**, L1 = **1e-4**
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (Original + Heatmap + Embossed)
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.67
Final Validation Accuracy	~0.65

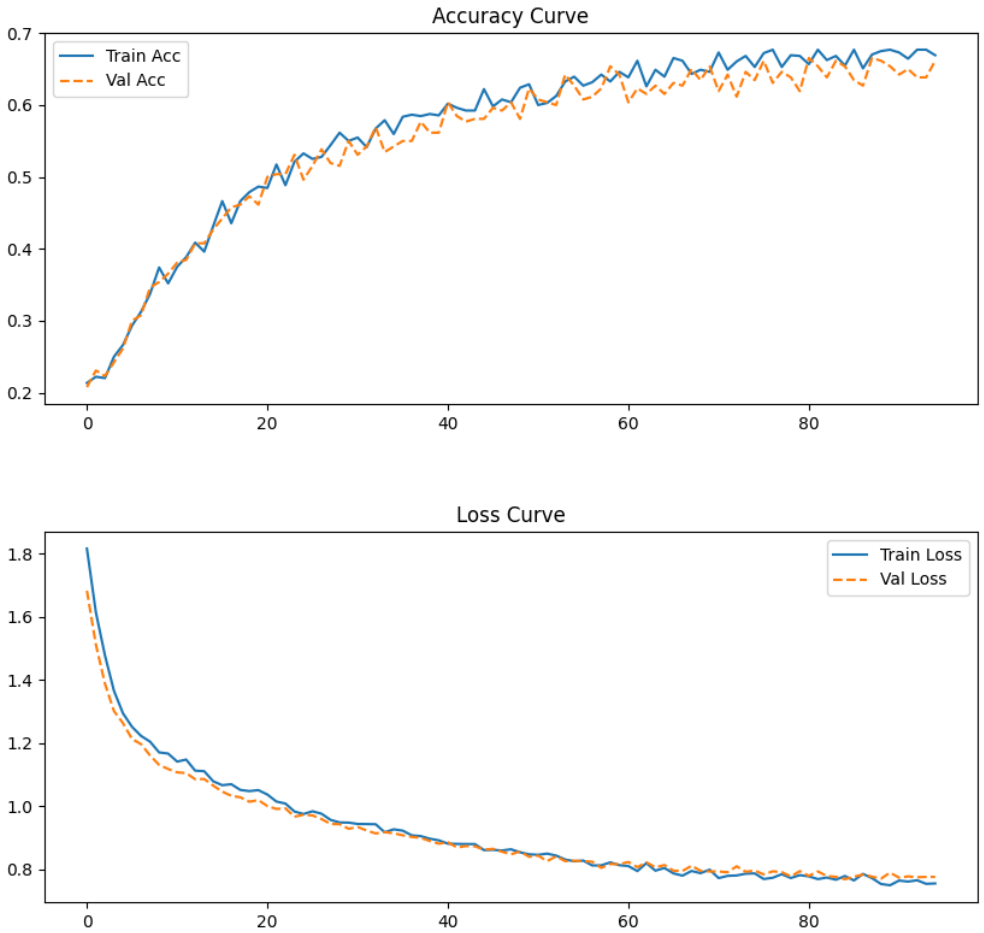


Fig 7.44 Accuracy & Loss Curve - Exp4

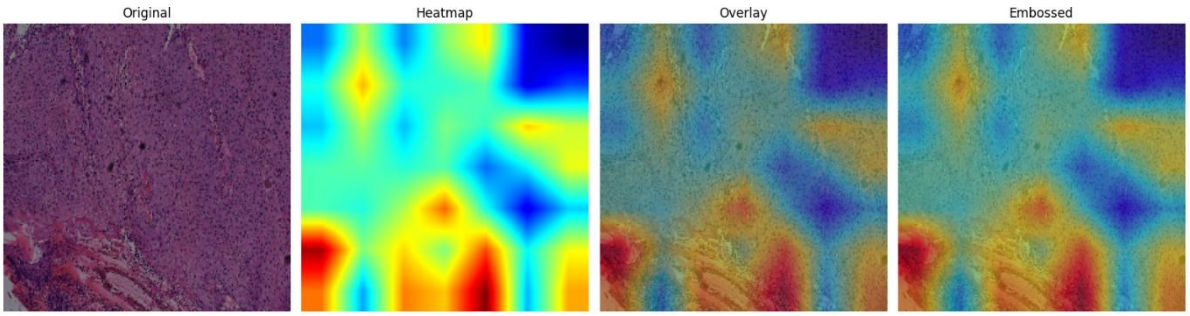


Fig 7.45 GradCam – Exp 4

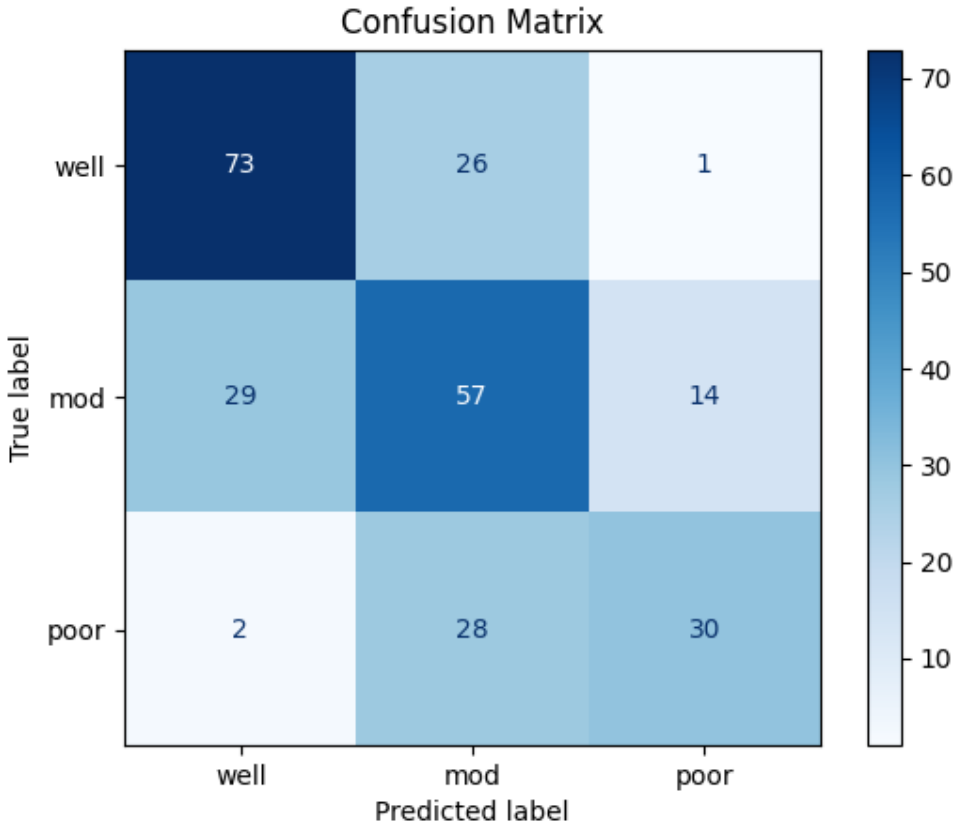


Fig 7.46 Confusion Matrix –Exp 4

Key Findings

The accuracy curve of Experiment 4 shows a strong and steady improvement throughout training. The model starts with a training accuracy of around 0.22 and rapidly climbs to above 0.60 within the first 30 epochs, indicating fast feature learning. After about 50 epochs, both training and validation accuracy stabilize between **0.62–0.68**, showing a consistent and smooth convergence. Importantly, the validation accuracy closely follows the training accuracy throughout the entire process, with only minor fluctuations. This behavior indicates **excellent generalization** and very minimal overfitting. The curves remaining close suggests that the regularization strategies, learning rate, and augmentation are well-balanced for this dataset.

The loss curves further confirm the stable learning behavior. The training loss drops sharply from around **1.82 to 0.74**, and the validation loss also decreases from about **1.68 to 0.76**. Both curves follow nearly identical paths with small variations, showing that the model is minimizing error effectively without memorizing the training data. The minimal gap between training and validation loss supports that the model is **well-regularized**, with no signs of

overfitting or underfitting. The smooth downward trend and plateau near the end indicate that the training reached a natural convergence point.

7.7.5 Results – Experiment 5

Experiment Setup:

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 1000, poor = 300
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.2**, L1 = **1e-4**
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.70
Final Validation Accuracy	~0.69

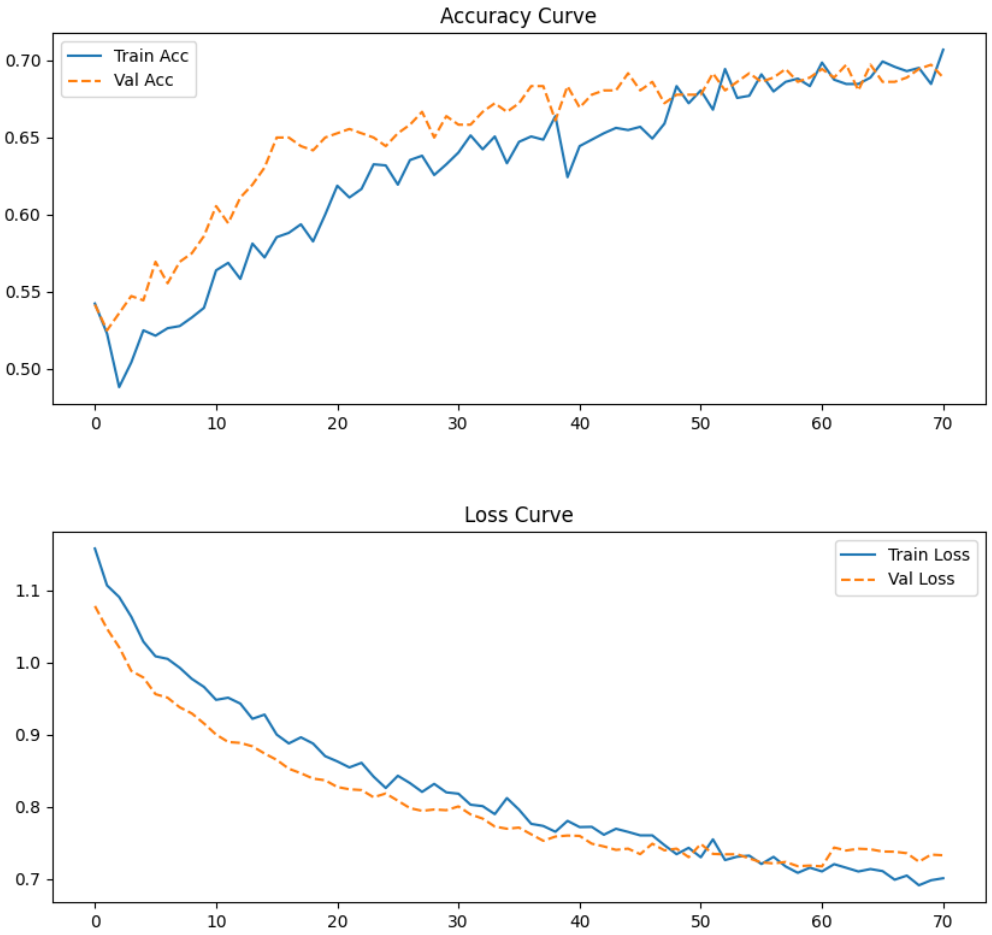


Fig 7.47 Accuracy & Loss Curve – Exp5

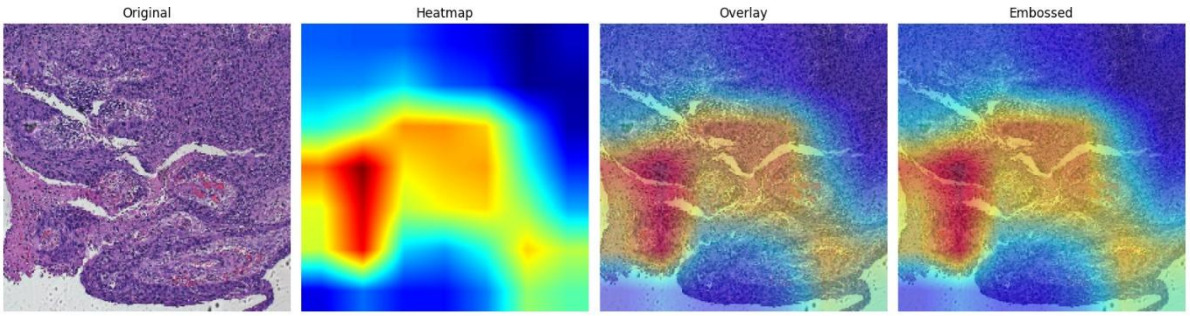


Fig 7.48 GradCam – Exp5

Key Findings:

The accuracy curve for Experiment 5 demonstrates a steadily improving learning pattern across all epochs. The training accuracy begins around **0.50** and gradually increases to approximately **0.70**, showing consistent improvement in the model's ability to correctly classify the training samples.A key observation is that the **validation accuracy is often slightly higher than the training accuracy**, especially in the middle epochs (around epoch 15–40). This suggests strong generalization performance, likely due to effective data augmentation, dropout, or regularization. Both accuracy curves remain tightly aligned, indicating **no overfitting**, and the consistency between them confirms that the model is learning balanced representations that transfer well to unseen data.The loss curves show a healthy, smooth downward trend throughout the training process. The training loss decreases from around **1.15** to approximately **0.69**, while the validation loss reduces from **1.08** to about **0.73**. The validation loss remains close to, and at times lower than, the training loss, which is a strong sign of **good convergence** and **strong generalization**. The curves do not diverge, and the gap remains small across all epochs. This confirms that the model is neither underfitting (which would show high loss) nor overfitting (which would show a widening gap between the two curves). Overall, the loss behaviour reflects a stable and efficient training configuration.

7.7.6 Results – Experiment 6

Experiment Setup:

Category	Details
Classes	well, mod, poor
Dataset Size	well = 500, mod = 700, poor = 300
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.2**, L1 = **1e-4**
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.69
Final Validation Accuracy	~0.70

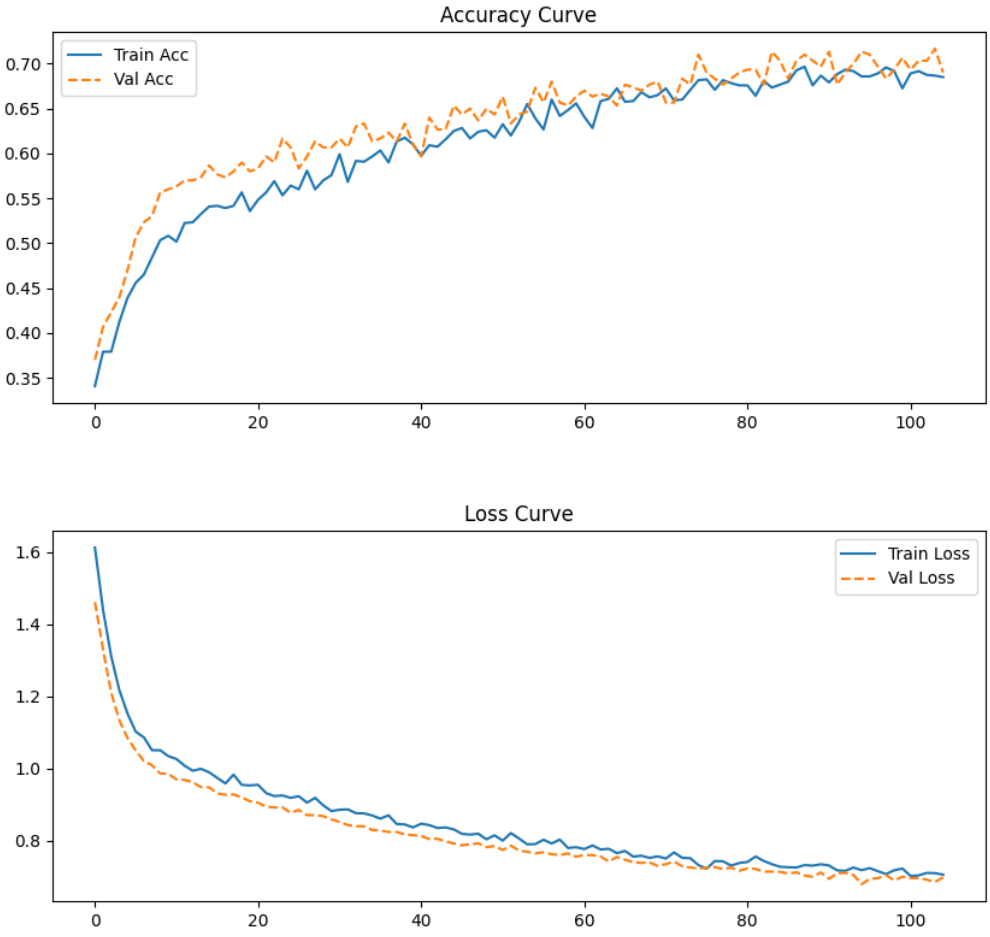


Fig 7.49 Accuracy & Loss Curve - Exp6

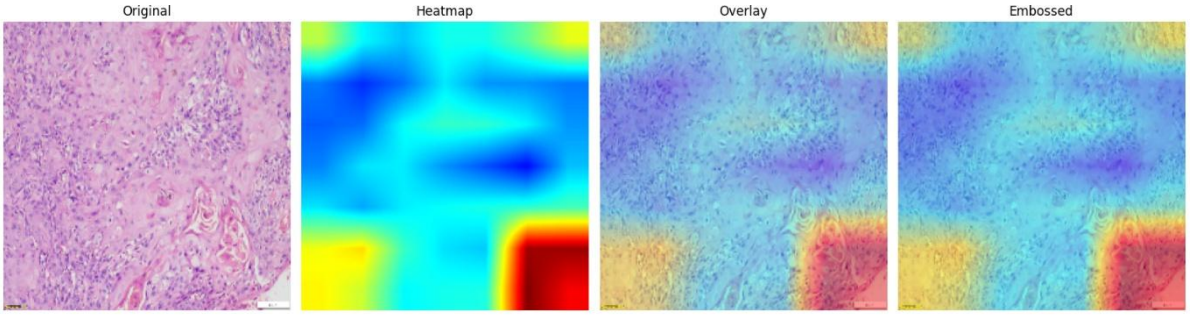


Fig 7.50 GradCam – Exp6

Key Findings:

The accuracy curve of Experiment 6 demonstrates a strong and stable learning progression across more than 100 epochs. The training accuracy starts around **0.35**, rapidly increases during the first 15–20 epochs, and continues to rise gradually until it stabilizes around **0.69–0.70**. The validation accuracy follows a very similar trend, beginning near **0.38** and steadily increasing to roughly **0.70**consistently staying slightly above the training accuracy throughout most of the training.

This pattern indicates **excellent generalization** the model performs equally well (or slightly better) on unseen data compared to the training set. The two accuracy curves remain very close, with only minor fluctuations in the validation curve, meaning the model avoids overfitting and maintains strong, consistent performance across epochs. The smooth, upward progression reflects that DenseNet121 is learning robust hierarchical features for the SCC classification task. The loss curves also show a highly stable and healthy convergence. The training loss begins at around **1.60** and smoothly decreases to approximately **0.71**, while the validation loss decreases from **1.45** to roughly **0.70**. The validation loss is consistently lower than or very close to the training loss throughout the entire training process. This behavior reinforces that the model is **not overfitting**, and the regularization settings, augmentations, and learning rate are well-tuned. No divergence or volatility is observed,

7.7.7 Results – Experiment 7- Data Split Into 4 Parts:

Experiment Setup:

Category	Details
Classes	well, mod, poor
Dataset Size	well = 800, mod = 800, poor = 604 → 800
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.1**, L1 = **1e-4**
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (multiple images per class)
Objective	3-class classification

Performance Metrics:

Metric	**Value**
Final Training Accuracy	~0.88
Final Validation Accuracy	~0.88

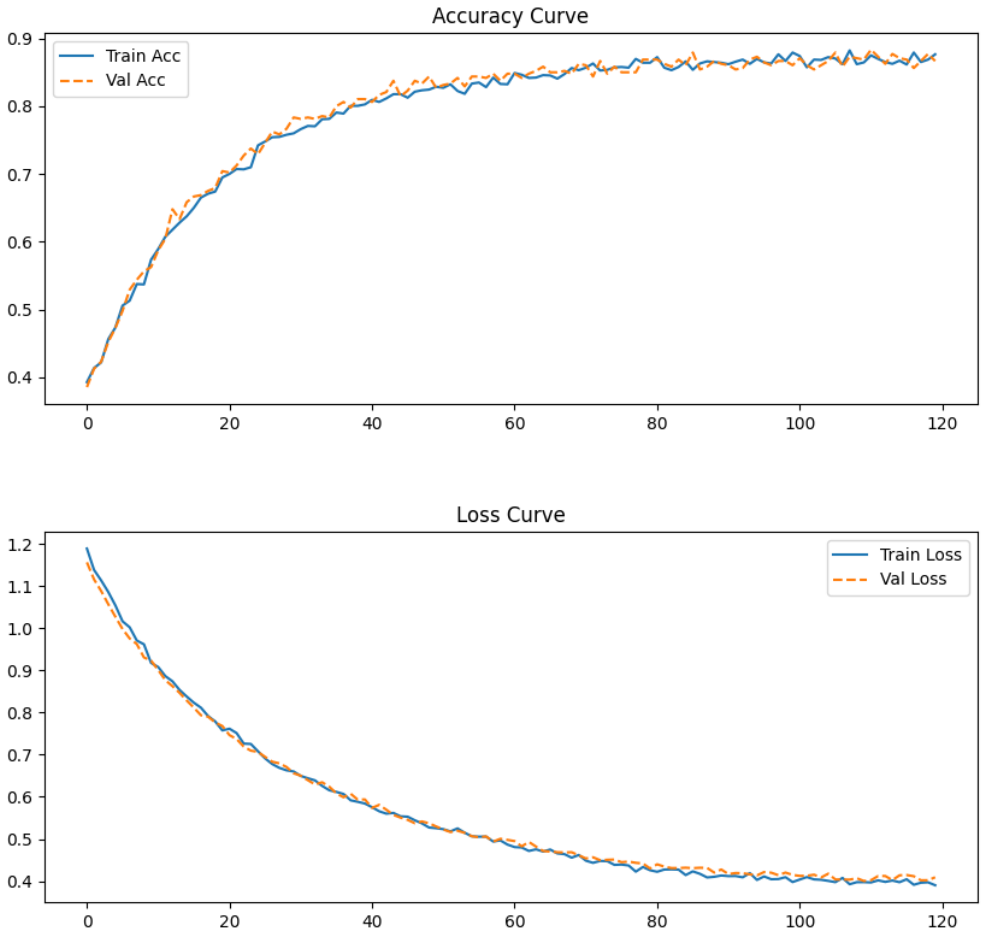


Fig 7.51 Accuracy & Loss Curve -Exp7

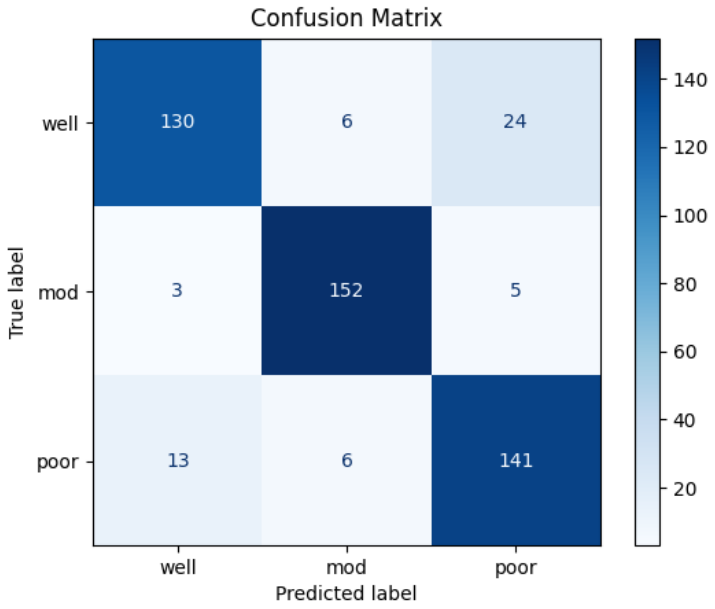


Fig 7.52 Confusion Matrix – Exp7

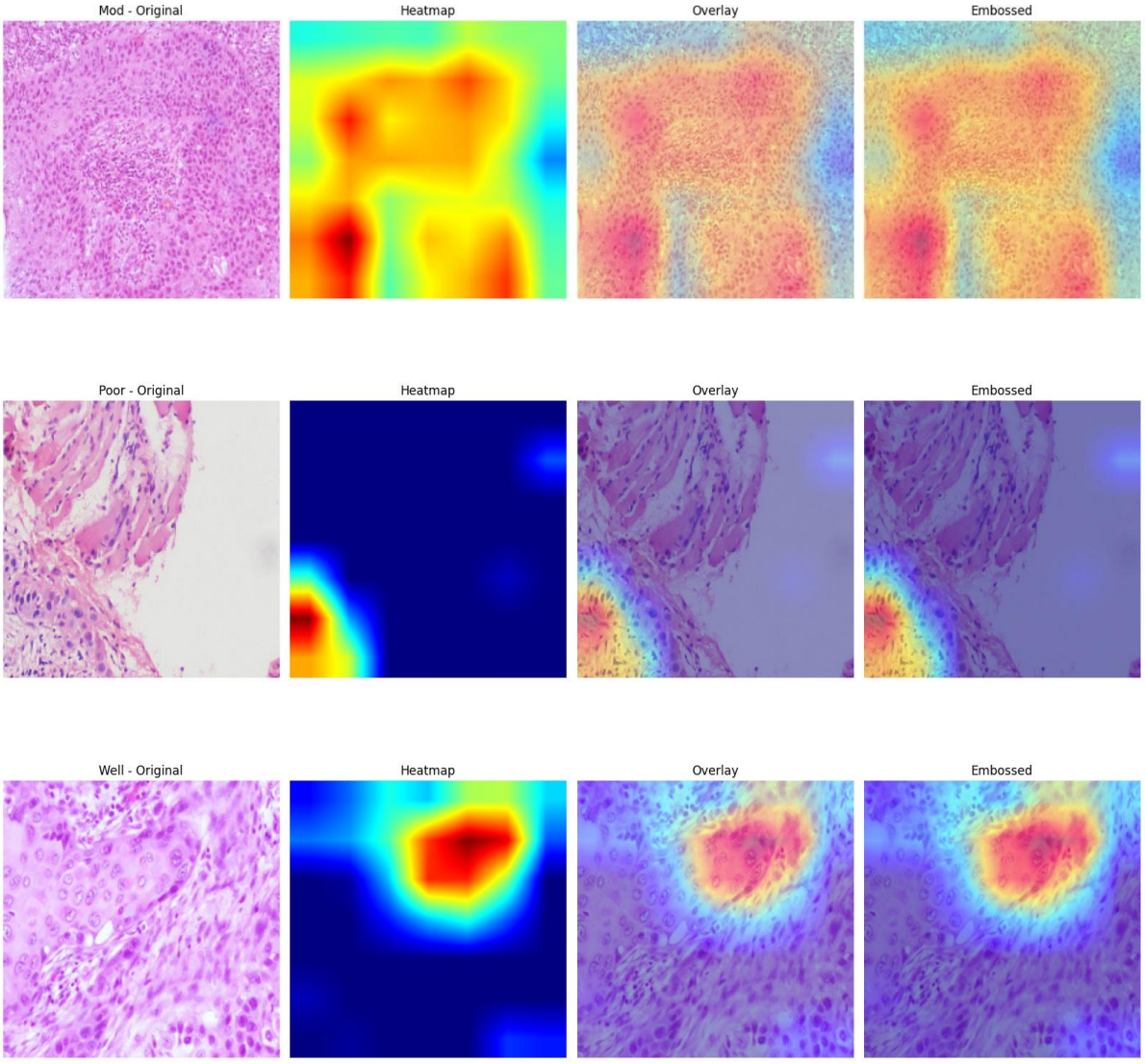


Fig 7.53 GradCam –Exp7

Key Findings:

The accuracy curve of Experiment 7 shows one of the strongest and most stable learning behaviors across all experiments. Training accuracy starts around **0.40** and increases rapidly during the first 20 epochs, indicating effective early learning. After this phase, accuracy continues to rise gradually until it stabilizes around **0.86–0.88** in the later epochs. Validation accuracy follows an almost identical trajectory, starting from **0.39** and steadily increasing to reach **0.87–0.88**, consistently overlapping the training accuracy curve. The extremely small gap between the two curves is an indicator of **excellent generalization** with **no signs of overfitting**. The validation accuracy never drops sharply or diverges, showing that the model maintains stable performance on unseen samples throughout all 120+ epochs. Such

close alignment between both curves suggests a well-balanced training configuration and optimal parameter tuning.

The loss curves further confirm the strong training stability of this experiment. The training loss decreases smoothly from about **1.19** to nearly **0.39**, while the validation loss follows the same progression, declining from **1.16** to approximately **0.40**. What stands out is the **perfect overlap** between training and validation loss curves. This indicates that the model is learning meaningful features without memorizing the training data. There is no widening gap or erratic jump in validation loss, which means the model avoids both underfitting and overfitting. The smooth and continuous decline reflects excellent optimization behavior and suggests that the chosen architecture (DenseNet121), hyperparameters, augmentation, and regularization strategies are perfectly aligned for this experiment.

7.7.8 Results – Experiment 8- Data Split Into 4 Parts:

Experiment Setup:

Category	Details
Classes	well, mod, poor
Dataset Size	well = 800, mod = 800, poor = 604 → 800
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	L1 = **1e-4** (Dropout removed)
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (multiple images per class)
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.89
Final Validation Accuracy	~0.88

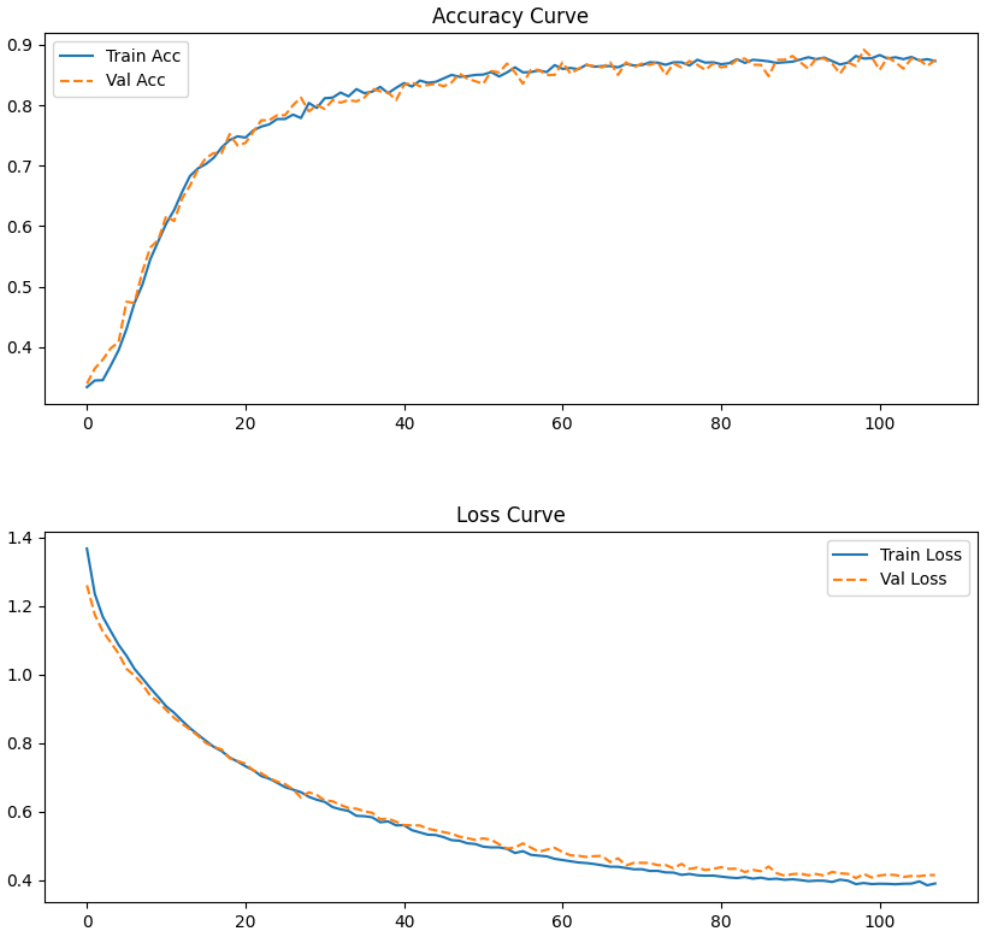


Fig 7.54 Accuracy & Loss Curve - Exp8

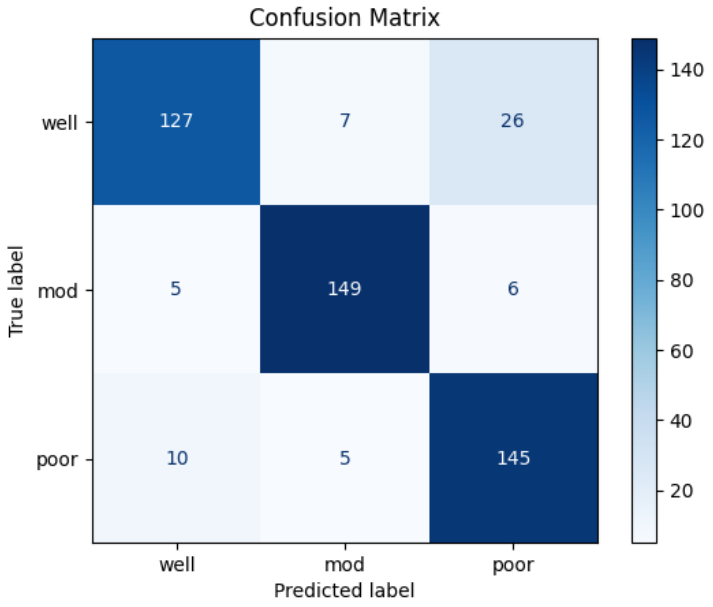
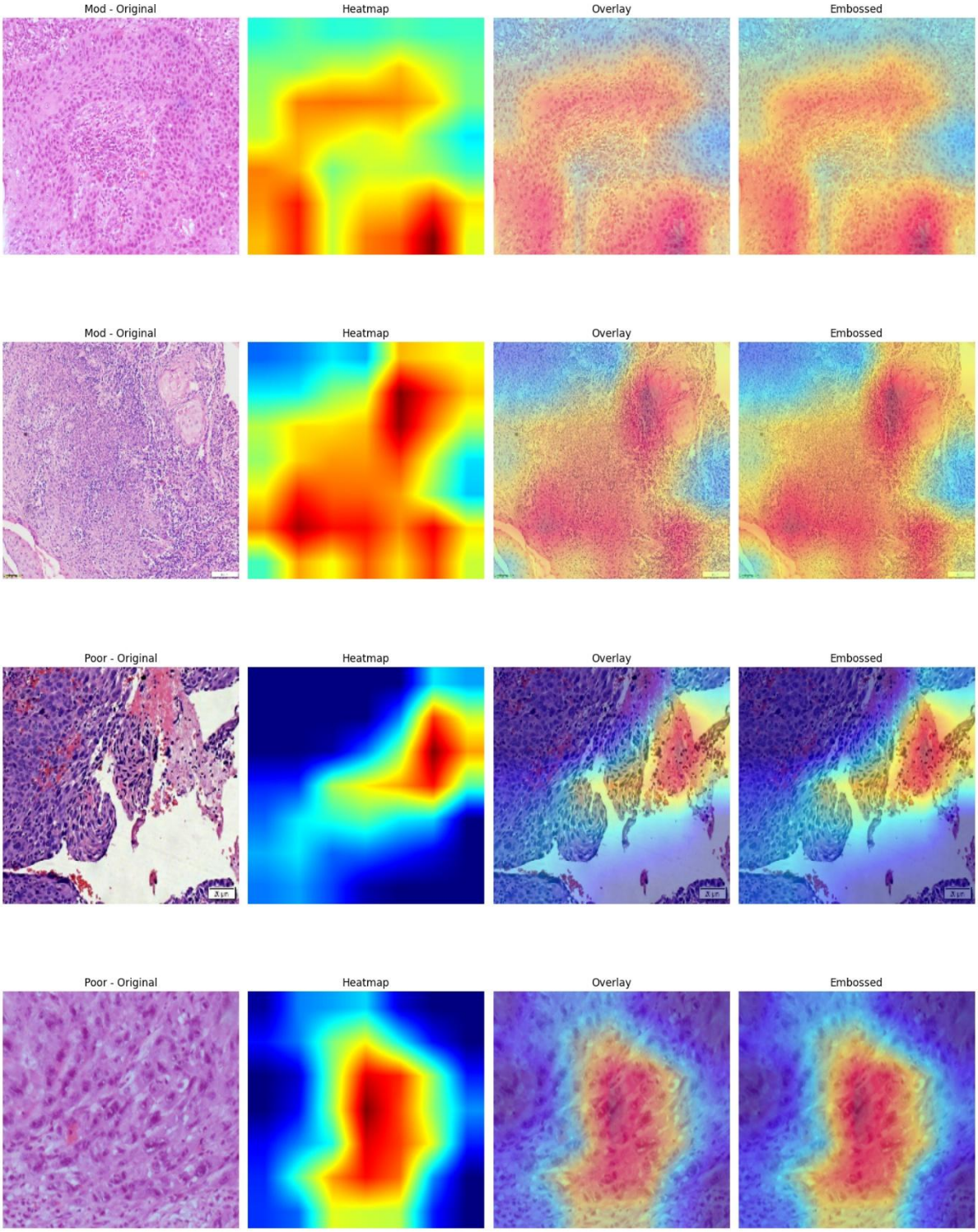


Fig 7.55 Confusion Matrix –Exp 8


Mod - Original
Heatmap
Overlay
Embossed
Mod - Original
Heatmap
Overlay
Embossed
Poor - Original
Heatmap
Overlay
Embossed
Poor - Original
Heatmap
Overlay
Embossed

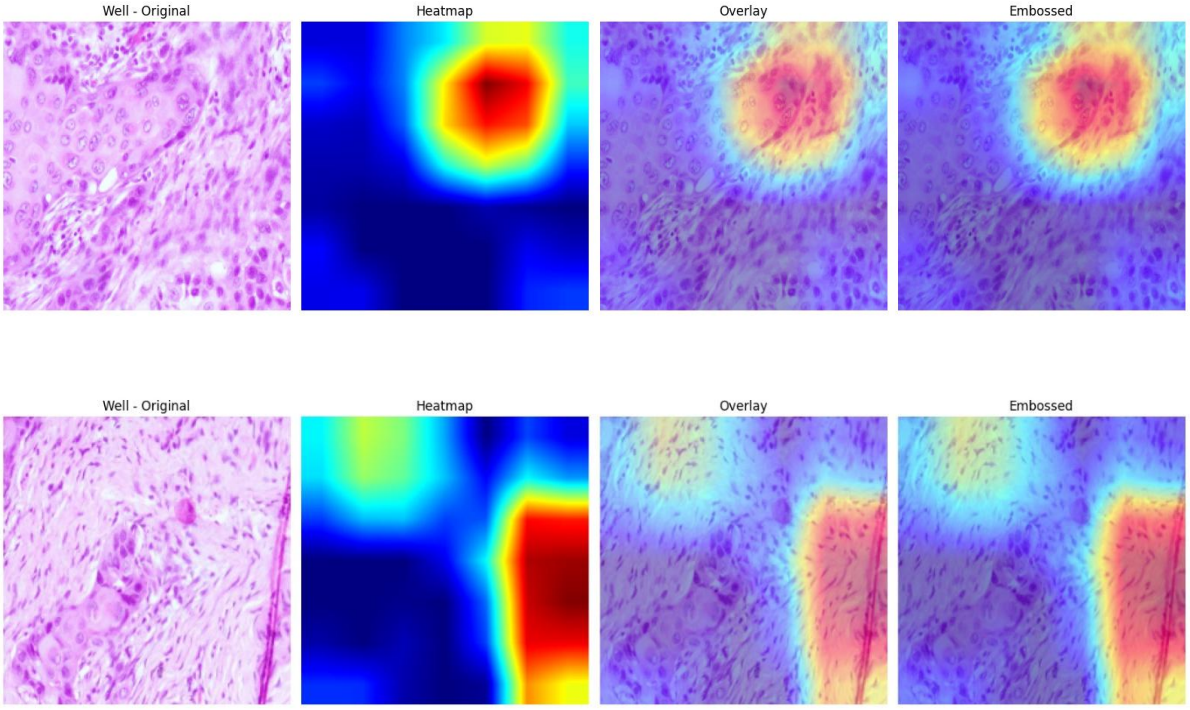


Fig 7.56 GradCam – Exp 9

Key Findings:

The accuracy curve of Experiment 8 demonstrates exceptionally strong and stable learning behavior. Training accuracy begins at around **0.34** and increases rapidly during the first 15–20 epochs, reaching the **0.70–0.75** range. After this early growth phase, the curve continues to rise gradually until it stabilizes around **0.87–0.89** in the later epochs.

The validation accuracy follows almost the exact same pattern, starting near **0.36** and eventually reaching **0.86–0.88**. Throughout the entire training process, the validation curve stays tightly aligned with the training curve, showing nearly identical values at each epoch. The minimal gap and parallel trend between the curves indicate **excellent generalization capability**, with **no signs of overfitting or underfitting**.

The smooth and consistent nature of the accuracy curves suggests that the DenseNet121 model, along with the chosen hyperparameters and augmentation strategies, reaches an optimal performance zone early and maintains stability throughout the long training period.

The loss curves further validate the strong performance of the model. Training loss decreases smoothly from around **1.36** to approximately **0.39**, while the validation loss similarly drops from about **1.25** to **0.41**.Both curves remain almost perfectly overlapped, with the validation loss occasionally being slightly higher, which is normal and expected.

7.7.9 Results – Experiment 9- Data Split Into 4 Parts:

Experiment Setup

Category	Details
Classes	well, mod, poor
Dataset Size	well = 800, mod = 800, poor = 604 → 800
Augmentation	Color Augmentation
Optimizer	Adam
Learning Rate	**3e-5**
Regularization	Dropout **0.1** (L1 removed)
Callbacks	LR Scheduler, Early Stopping
Model Add-ons	Grad-CAM (multiple images per class)
Objective	3-class classification

Performance Metrics:

Metric	Value
Final Training Accuracy	~0.88
Final Validation Accuracy	~0.87

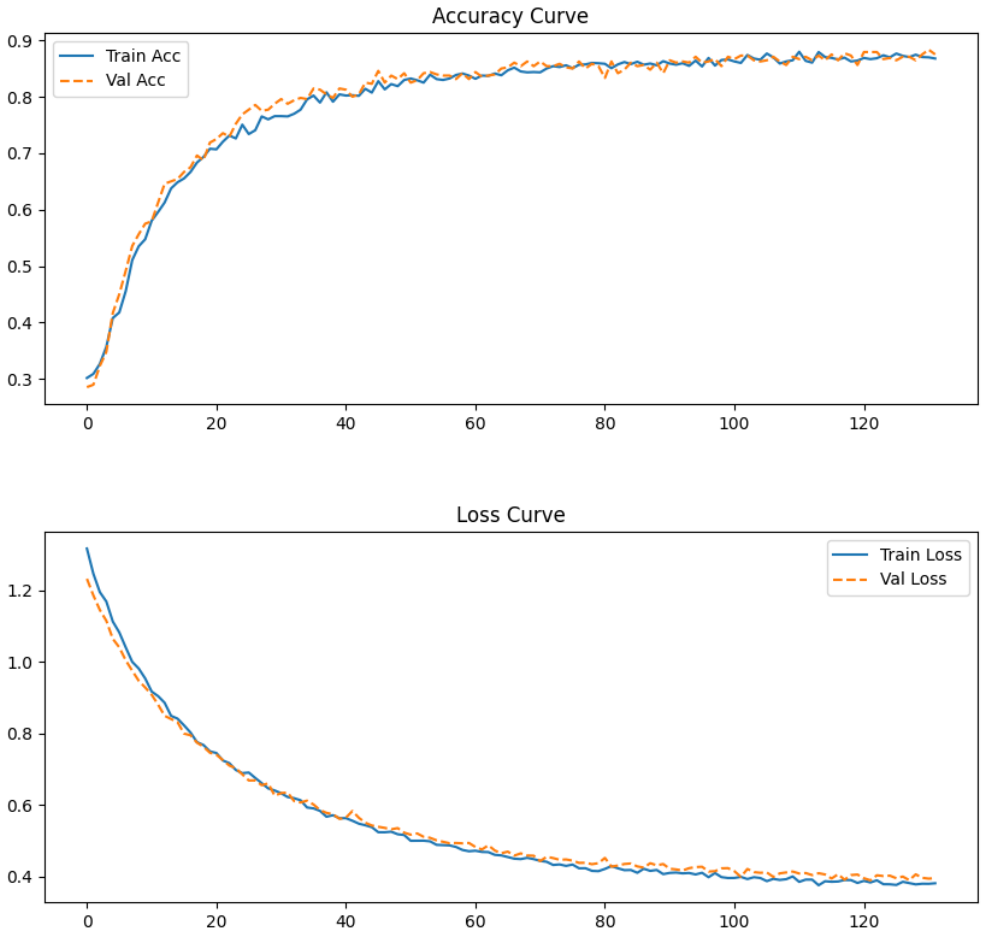


Fig 7.57 Accuracy & Loss Curve –Exp 9

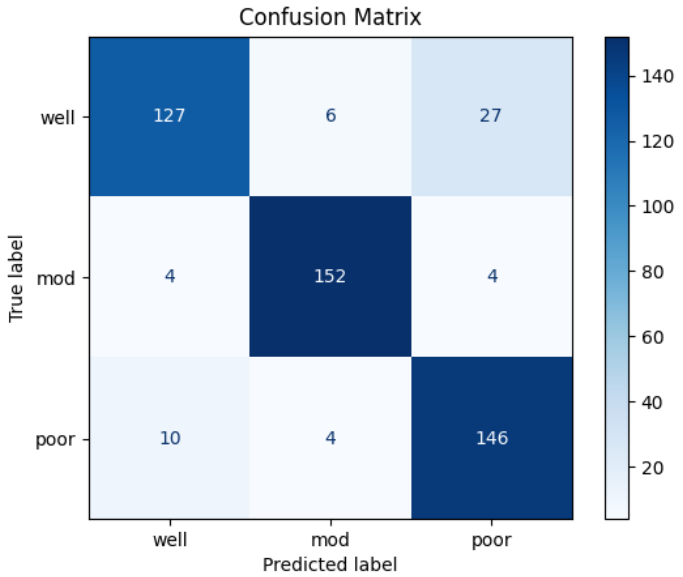
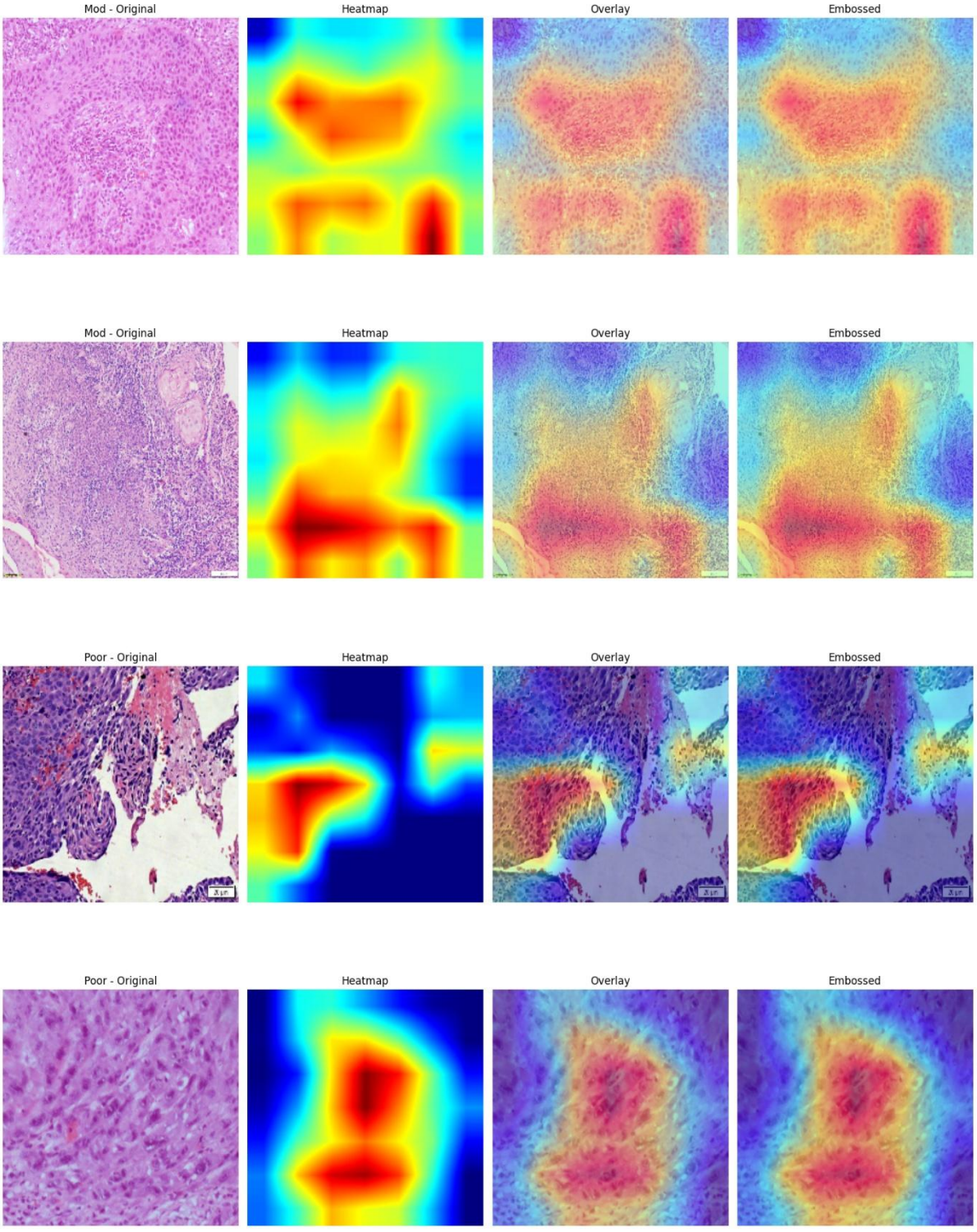


Fig 7.58 Confusion Matrix –Exp9


Mod - Original
Heatmap
Overlay
Embossed
Mod - Original
Heatmap
Overlay
Embossed
Poor - Original
Heatmap
Overlay
Embossed
Poor - Original
Heatmap
Overlay
Embossed

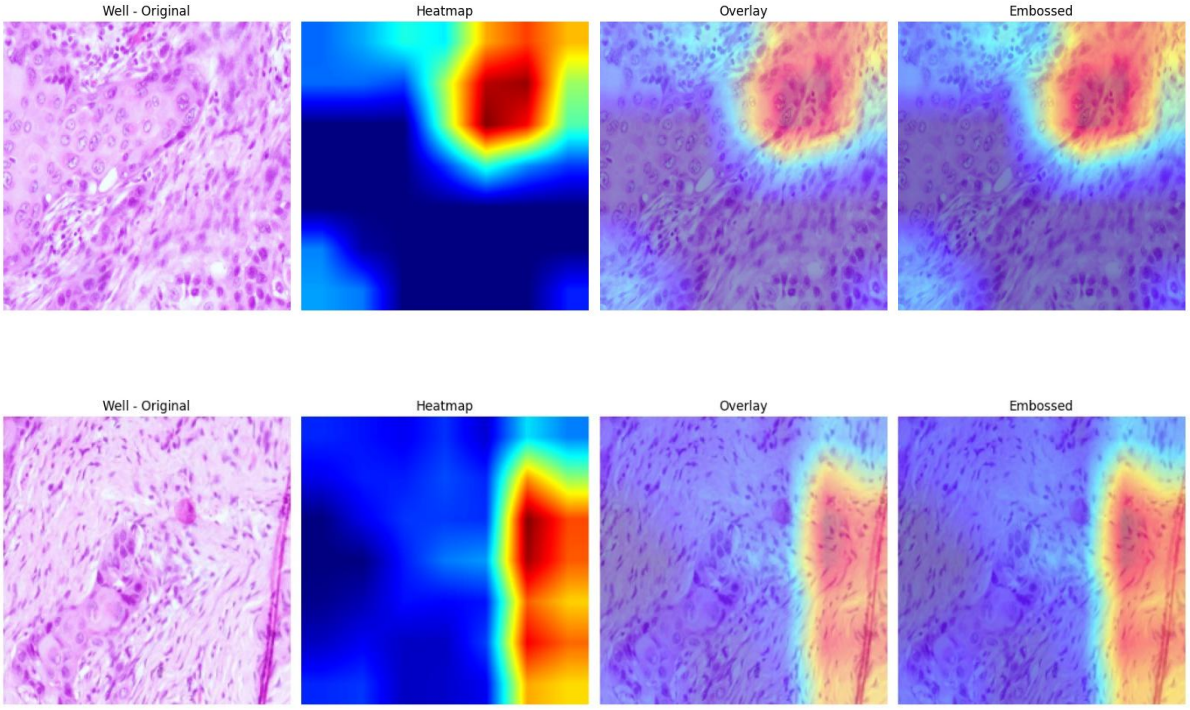


Fig 7.59 GradCam –Exp9

Key Findings:

The accuracy curve of Experiment 9 shows a strong, smooth, and highly consistent learning trajectory.Training accuracy begins around **0.29** and increases rapidly during the first 20 epochs, reaching above **0.70**. After this, the curve continues to rise steadily and stabilizes in the **0.85–0.88** range during the later epochs.

Validation accuracy follows almost the exact same pattern, starting at around **0.30** and quickly catching up to the training accuracy. Throughout the entire training process, both curves remain extremely close, with validation accuracy occasionally matching or slightly exceeding the training accuracy. This near-perfect alignment indicates **excellent generalization**, meaning the model is learning representative, non-overfitted features.

The stability of the curves well beyond 100 epochs shows that the DenseNet121 model continues to refine feature representations without showing any overfitting behavior, making this configuration very reliable for SCC classification. The loss curves provide further confirmation of the model's robustness. Training loss drops smoothly from around **1.30** to

approximately **0.37**, while validation loss decreases from **1.2** to about **0.39**. Both curves follow the same downward trajectory and remain tightly aligned across all epochs. This strong overlap demonstrates that the model maintains an optimal balance between bias and variance, with no divergence or instability. The absence of fluctuations or sudden spikes signifies effective optimization and well-tuned hyperparameters.

7.8 Results of Multi-Class SCC Grading using MobileNetV2 with Systematic Experimentation

7.8.1. Results - Experiment 1

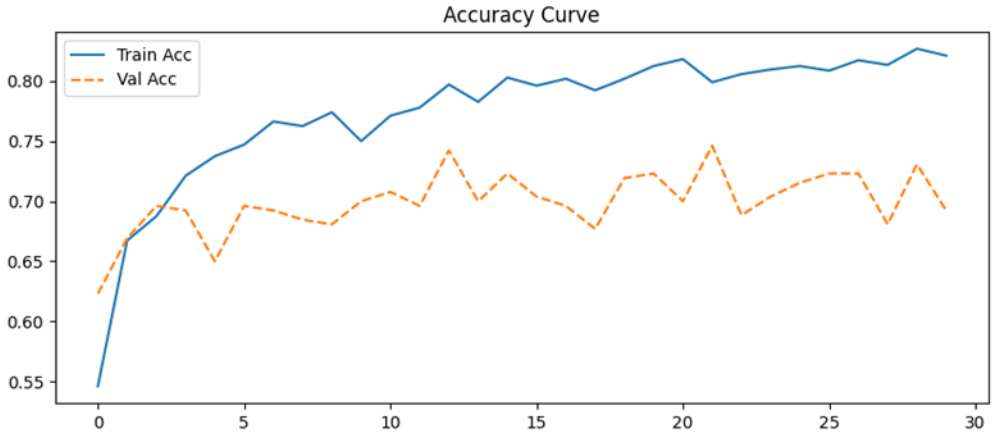
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-3
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.8220
Val accuracy	0.7050

Accuracy and Loss curve



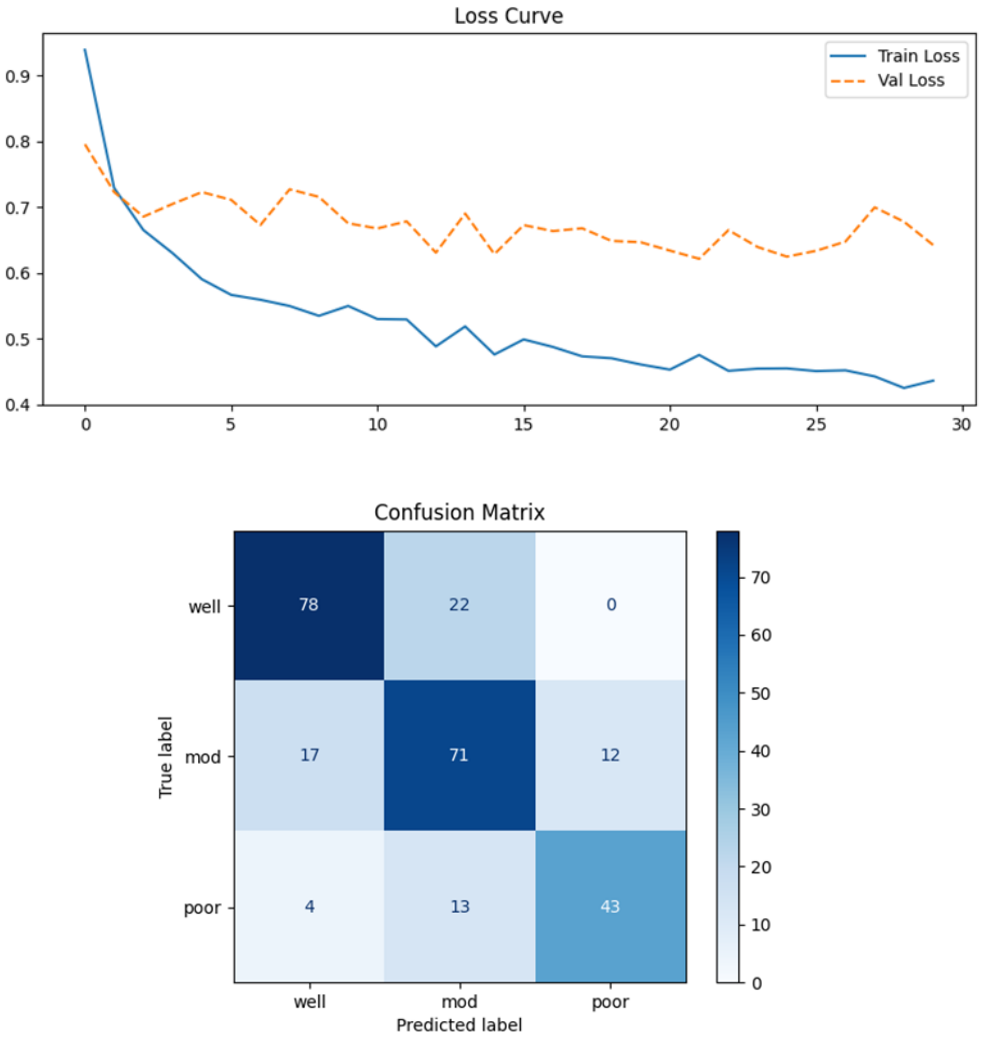


Fig 7.60. Acc, Loss curve & confusion matrix of Experiment 1

Key Findings

The model trained with a learning rate of **1e-3** shows fast and stable optimization in the early stages, with training accuracy rising steadily from ~55% to ~82%. However, the validation accuracy plateaus around **68–72%**, creating a noticeable and consistent performance gap. The loss curves further confirm this behavior: training loss decreases smoothly to ~0.42, while validation loss remains significantly higher (~0.63–0.70) and fluctuates throughout training. This divergence between training and validation performance indicates that while the model successfully fits the training data, its generalization capacity does not improve at the same rate.

Overall, the experiment demonstrates that **LR = 1e-3 leads to mild but clear overfitting**. The model learns rapidly, but the relatively high learning rate causes it to continue pushing training loss down without a corresponding improvement in validation performance. As a result, the

model converges to a solution that is well-optimized for the training distribution but not fully robust on unseen data.

7.8.2. Results - Experiment 2

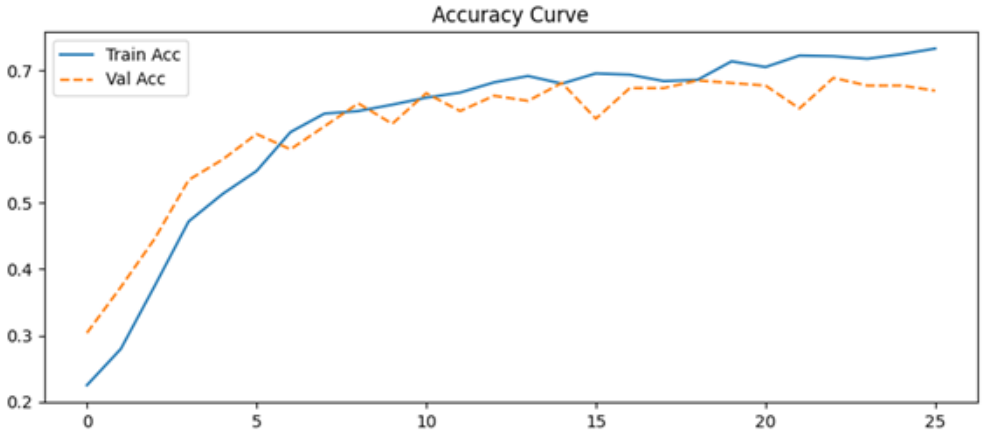
Experimental Setup

Parameter	Details
Classes	Well, mod, poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-4
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.7300
Val accuracy	0.6700

Accuracy and Loss curve



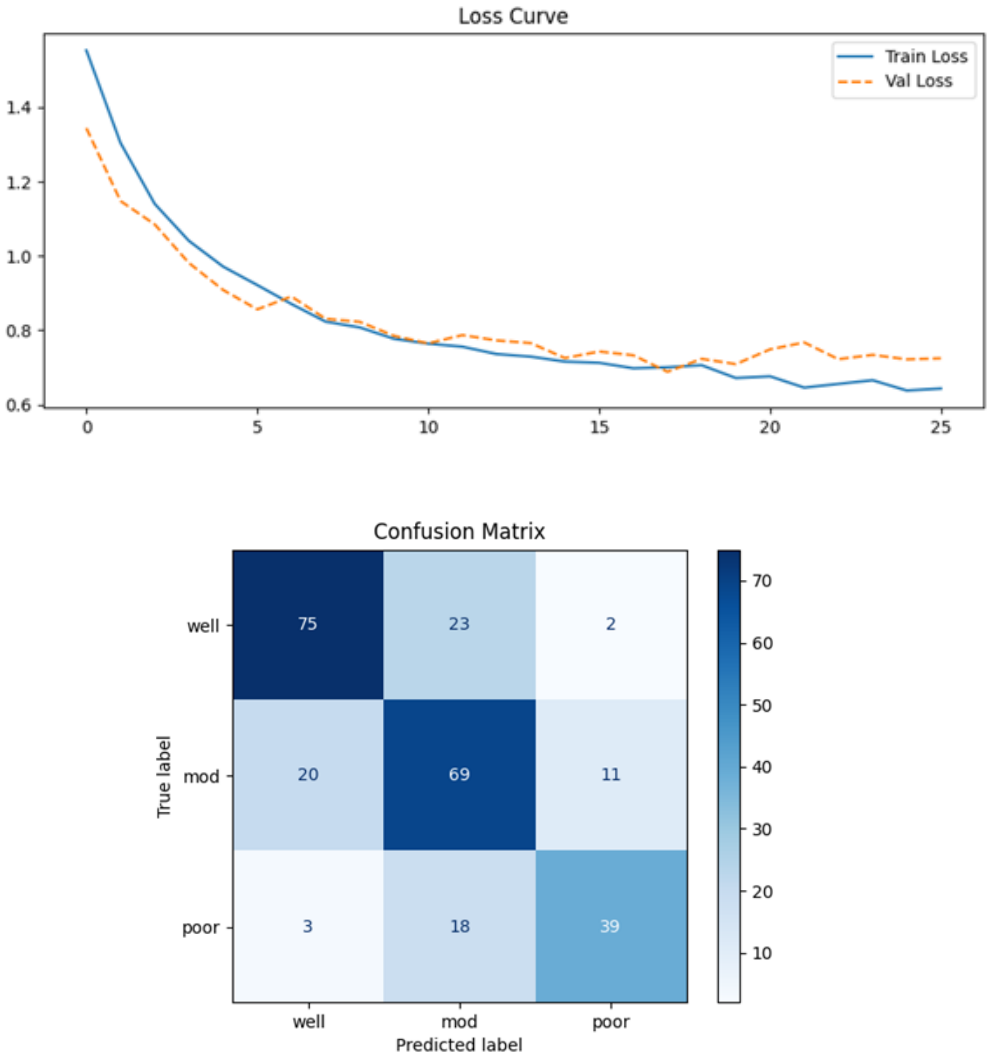


Fig 7.61. Acc, Loss curve & confusion matrix of Experiment 2

Key Findings

The model trained with a learning rate of **1e-4** shows stable and gradual learning, with both training and validation accuracy improving consistently over the epochs. Training accuracy reaches approximately **73%**, while validation accuracy stabilizes around **67%**, resulting in a moderate generalization gap. This indicates that the model is learning effectively without aggressive updates, which is expected at a lower learning rate.

The loss curves further support this behavior: training loss decreases smoothly to around **0.62**, while validation loss remains slightly higher at **~0.72**. The small but noticeable gap between the loss curves suggests **mild overfitting**, but significantly less than what is typically observed with higher learning rates. Overall, **LR = 1e-4 provides more controlled optimization**, better

stability, and improved generalization behavior compared to a high learning rate, though it converges slower and reaches slightly lower peak performance.

7.8.3. Results - Experiment 3

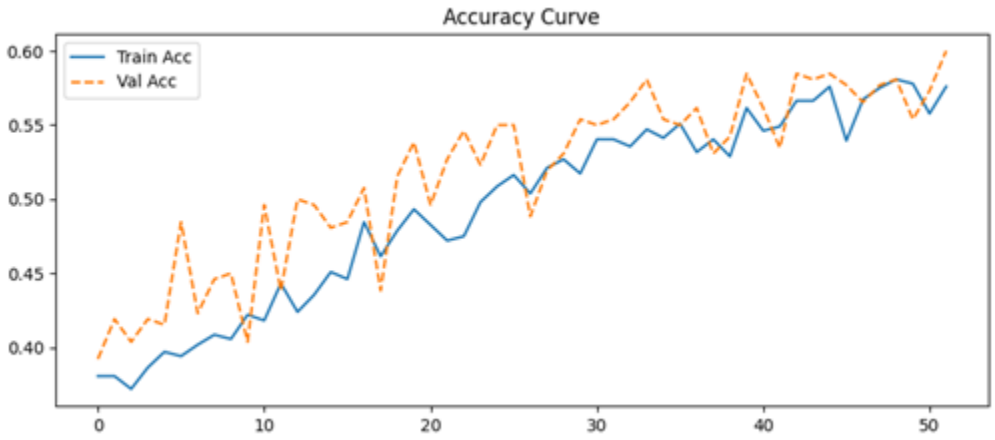
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-5
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.5800
Val accuracy	0.6000

Accuracy and Loss curve



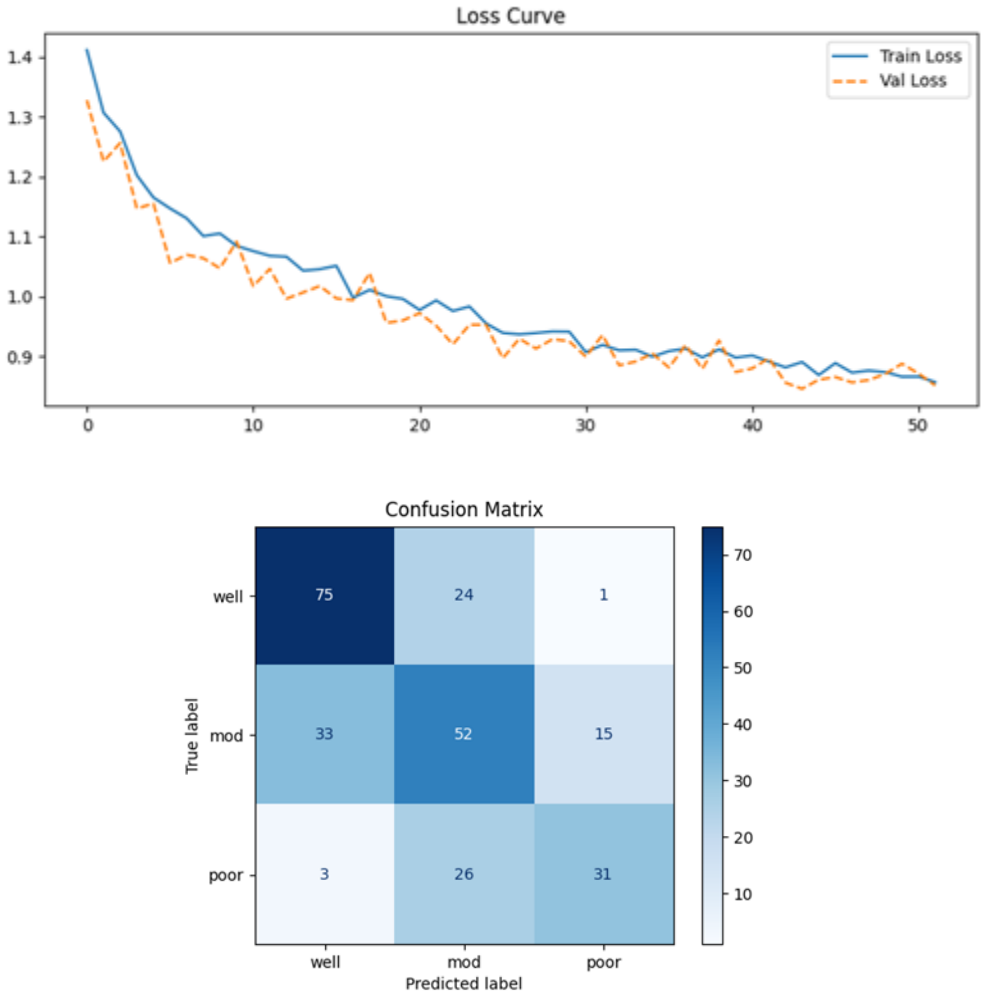


Fig 7.62. Acc, Loss curve & confusion matrix of Experiment 3

Key Findings

Training with a learning rate of **1e-5** results in very slow but extremely stable learning. Both training and validation accuracy increase gradually and remain close throughout the entire training process, indicating **no overfitting and no underfitting**. The validation accuracy slightly surpasses the training accuracy at several points, which is typically a sign of strong generalization when updates per step are very small.

The loss curves further support this behavior: both training and validation loss decrease slowly and almost in parallel, ending around **0.87–0.88**, which is notably higher than the loss values achieved with larger learning rates. This indicates that while the model generalizes well, the learning rate is **too small for the model to meaningfully optimize its parameters**, resulting in lower overall accuracy and slower convergence. Overall, **LR = 1e-5 provides maximum**

stability but insufficient learning, making it unsuitable for achieving high performance on this task.

7.8.4. Results - Experiment 4

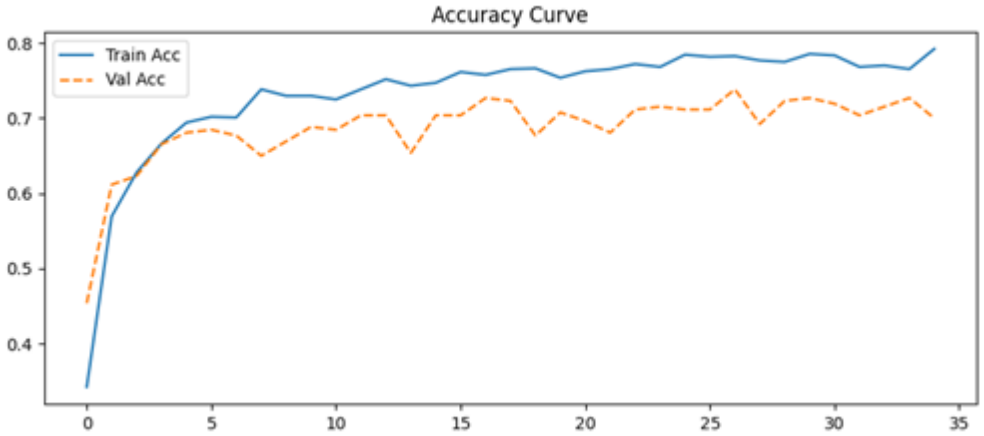
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	3e-4
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.7800
Val accuracy	0.7000

Accuracy and Loss curve



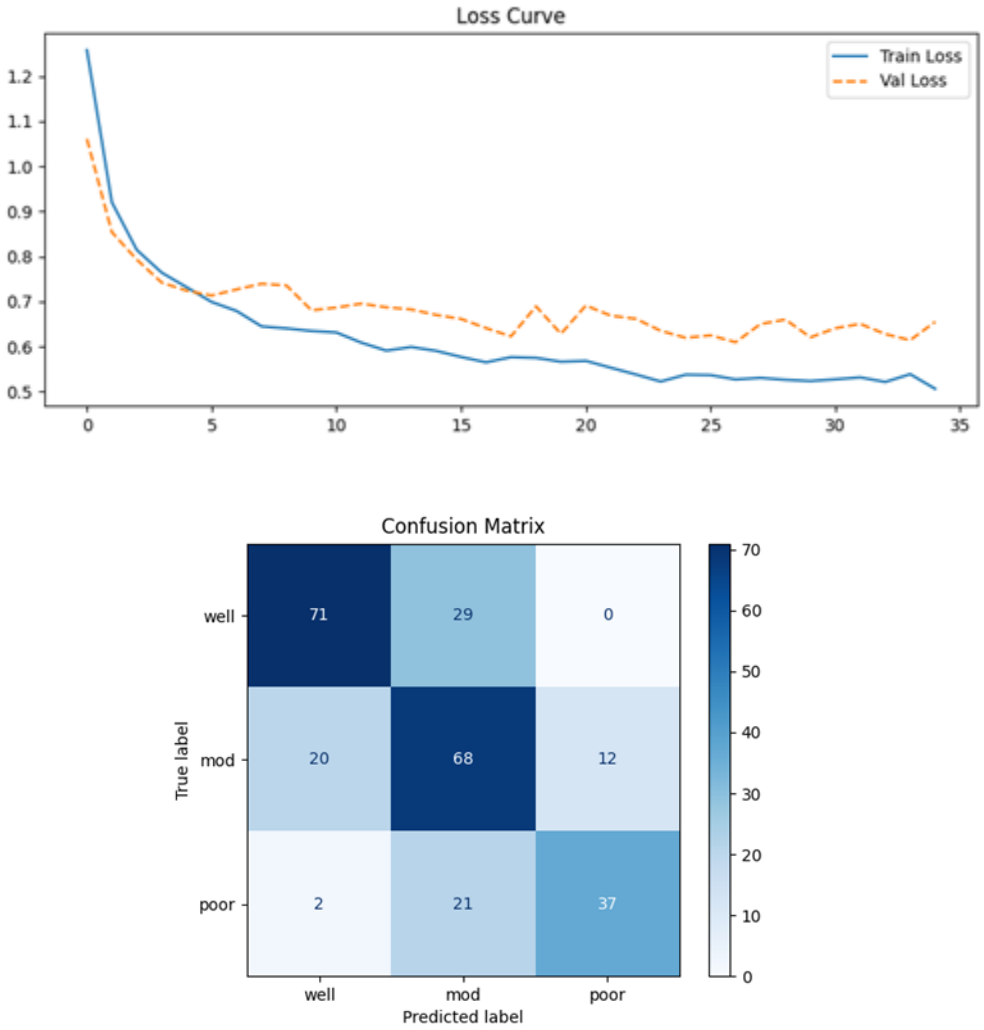


Fig 7.63. Acc, Loss curve & confusion matrix of Experiment 4

Key Findings

Training with a learning rate of **3e-4** results in fast and effective learning during the first few epochs, with training accuracy rising rapidly to around **78%** and validation accuracy stabilizing near **70%**. This indicates that the model is learning meaningful representations early and converges quicker than with 1e-4 or 1e-5. However, the validation accuracy plateaus around 70%, and the gap between train and validation performance gradually increases, showing that the model begins to **slightly overfit after ~10–12 epochs**.

The loss curves reinforce this conclusion: training loss continues decreasing smoothly toward **0.52**, while validation loss flattens around **0.65–0.67**, resulting in a noticeable but manageable loss gap. This suggests that the model is fitting the training data well but is not improving its

generalization beyond a certain point. Overall, **LR = 3e-4 offers a good balance between speed and stability**, but introduces **mild overfitting** and does not generalize as strongly as the 1e-4 experiment.

7.8.5. Results - Experiment 5

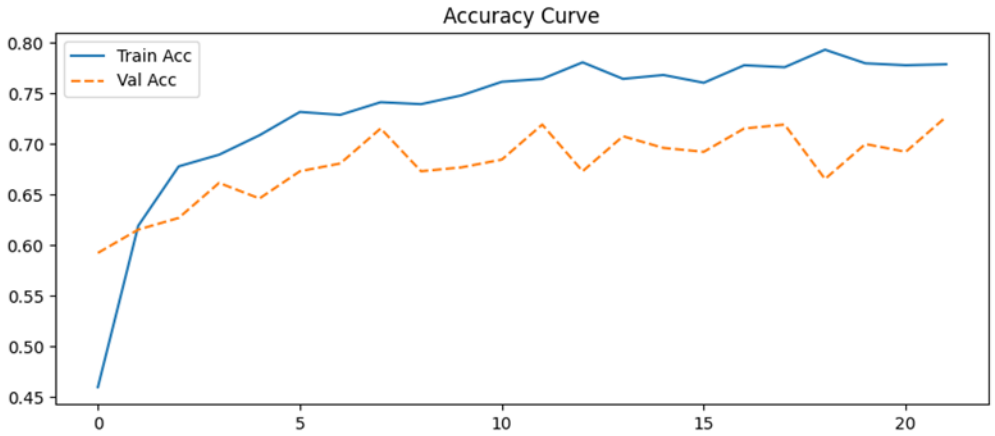
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	5e-4
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.7800
Val accuracy	0.7000

Accuracy and Loss curve



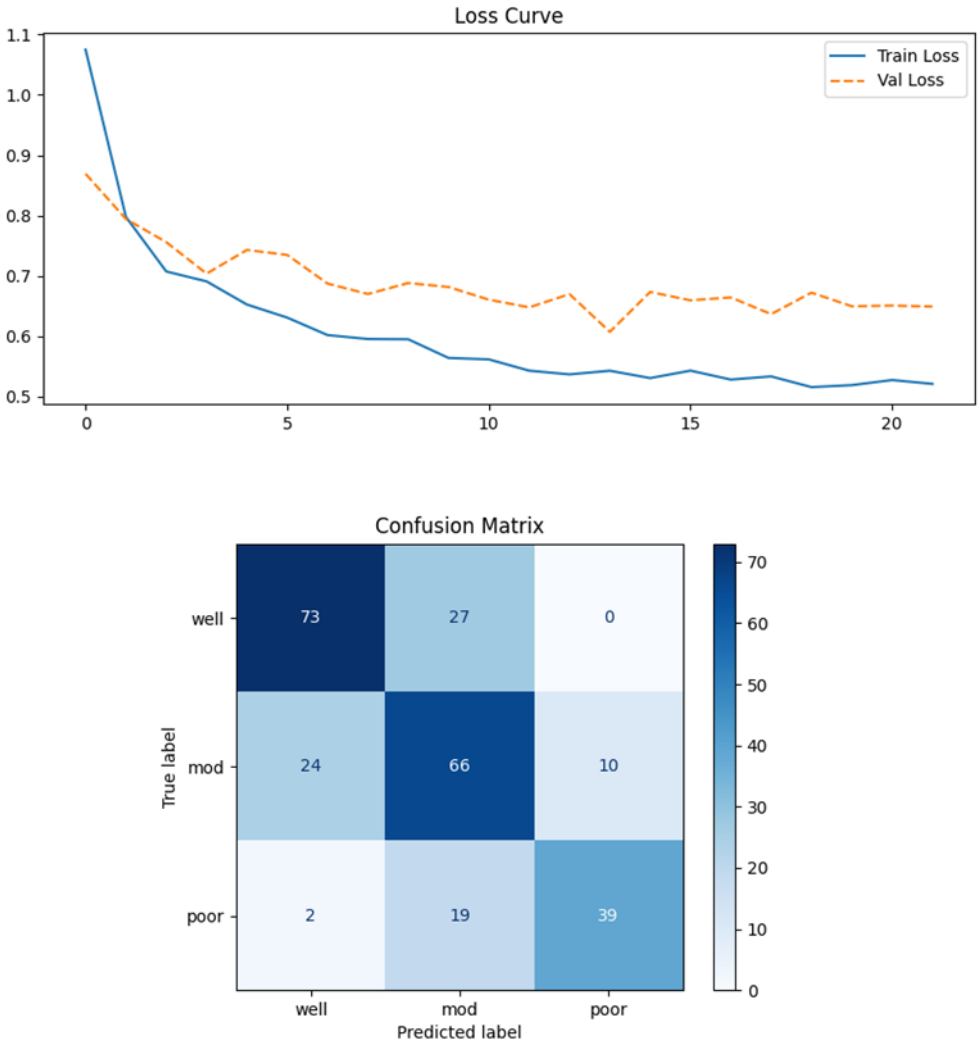


Fig 7.64. Acc, Loss curve & confusion matrix of Experiment 5

Key Findings

The model trained with a learning rate of **5e-4** shows fast learning during the initial epochs, with training accuracy rising sharply to around **78%** and validation accuracy stabilizing near **70%**. This indicates that the learning rate is high enough to accelerate convergence, comparable to LR=3e-4, but not so high as to destabilize training. However, after the early phase, the validation accuracy plateaus and does not improve further, even though training accuracy continues increasing—indicating **mild overfitting** beginning around epoch 8–10.

The loss curves also reveal this pattern: training loss steadily decreases to ~0.51, while validation loss flattens at ~0.65 and shows small oscillations. This widening gap between training and validation loss reflects **reduced generalization** compared to lower learning rates

like 1e-4. Overall, **LR = 5e-4 provides fast convergence but introduces noticeable overfitting**, making it slightly less stable and less generalizable than LR=3e-4 or LR=1e-4.

7.8.6. Results - Experiment 6

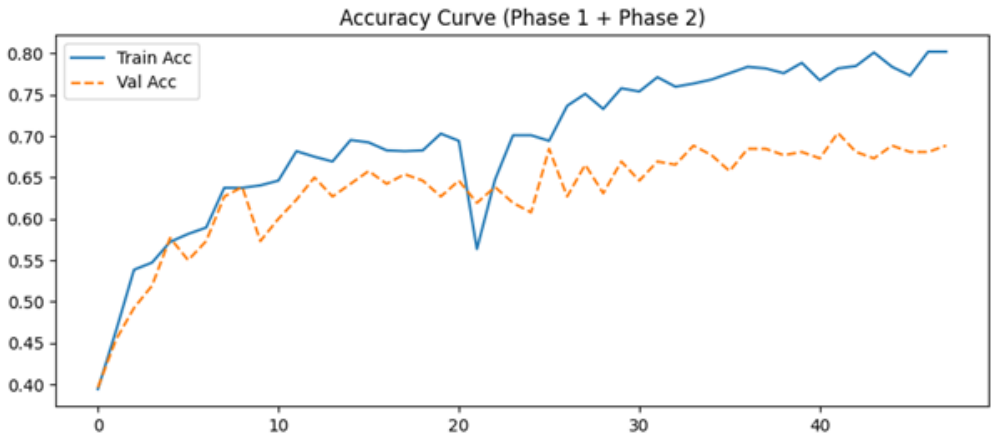
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	Initial=1e-4, Recompilation=1e-5
Add-ons	Unfreezing last 30layers.
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.8000
Val accuracy	0.6850

Accuracy and Loss curve



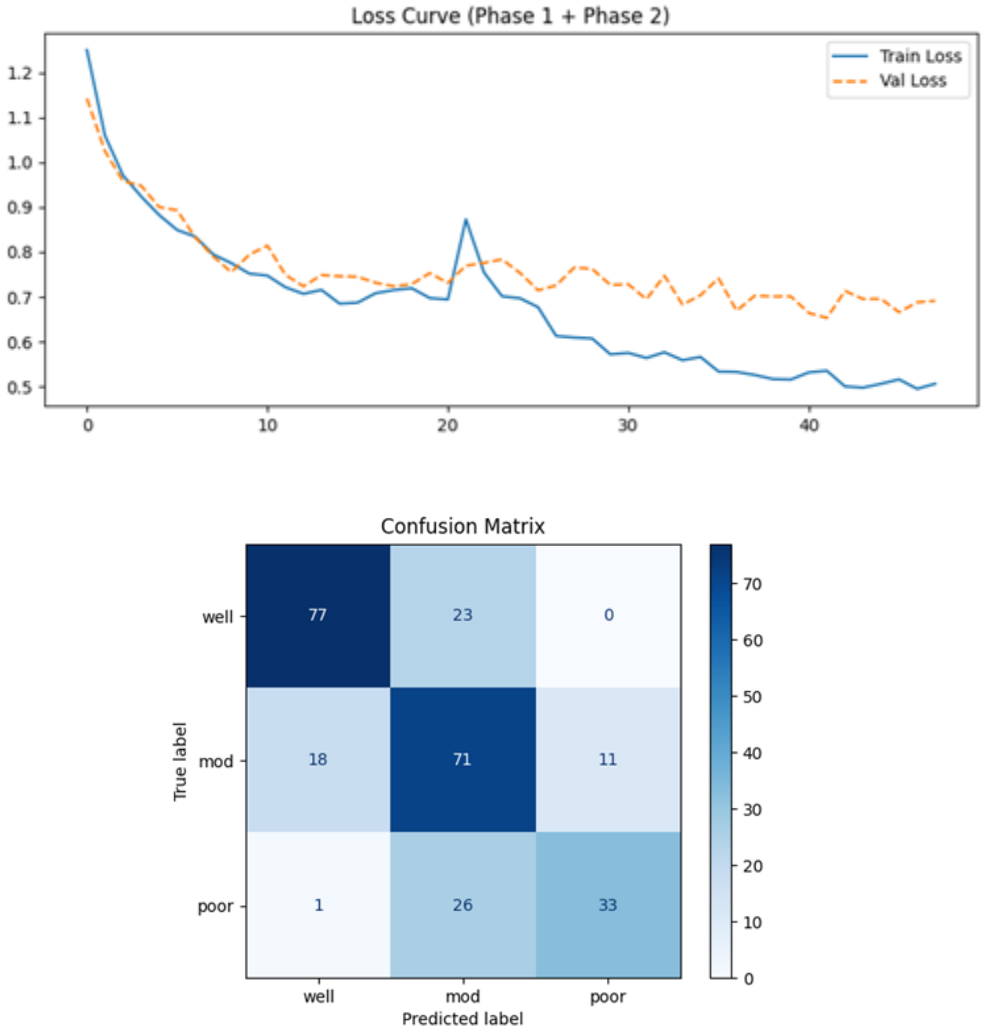


Fig 7.65. Acc, Loss curve & confusion matrix of Experiment 6

Key Findings

This two-phase training strategy—starting with **LR = 1e-4** and fine-tuning with a lower **LR = 1e-5**—resulted in strong overall model performance. Phase 1 allowed the model to quickly learn high-level features, while Phase 2 refined deeper layers more gently. Training accuracy steadily increased to around **80%**, while validation accuracy stabilized near **68–69%**, demonstrating effective learning but also the emergence of a moderate generalization gap after fine-tuning.

The loss curves reveal a typical fine-tuning pattern: steady decline in training loss throughout Phase 2, while validation loss plateaus around **0.69**, indicating **mild overfitting** once the deeper layers were unfrozen. The spike in loss near epoch ~20 corresponds to the transition between

frozen and unfrozen layers, which is normal behavior. Overall, this LR schedule improves representational learning and yields better performance than a single-LR training run, but shows a small increase in overfitting due to deeper fine-tuning.

7.8.7. Results - Experiment 7

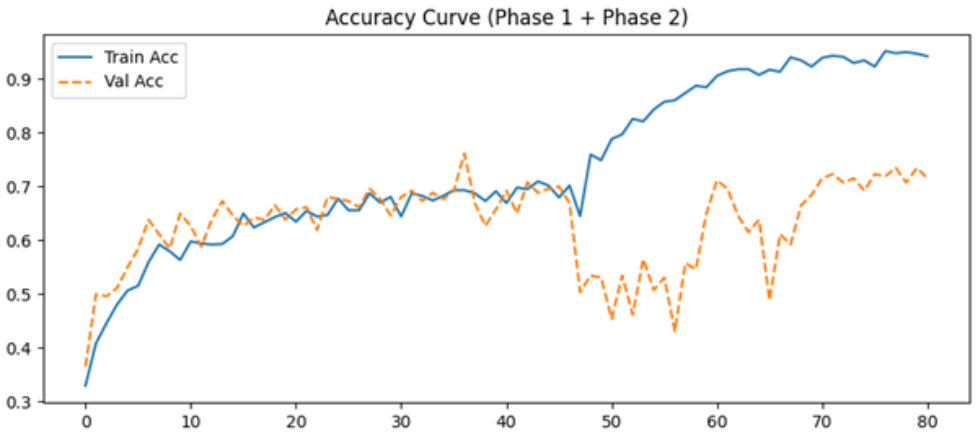
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-4
Add-ons	Unfreezing last 40layers.
Regularizer	Dropout=0.3
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.9300
Val accuracy	0.7000

Accuracy and Loss curve



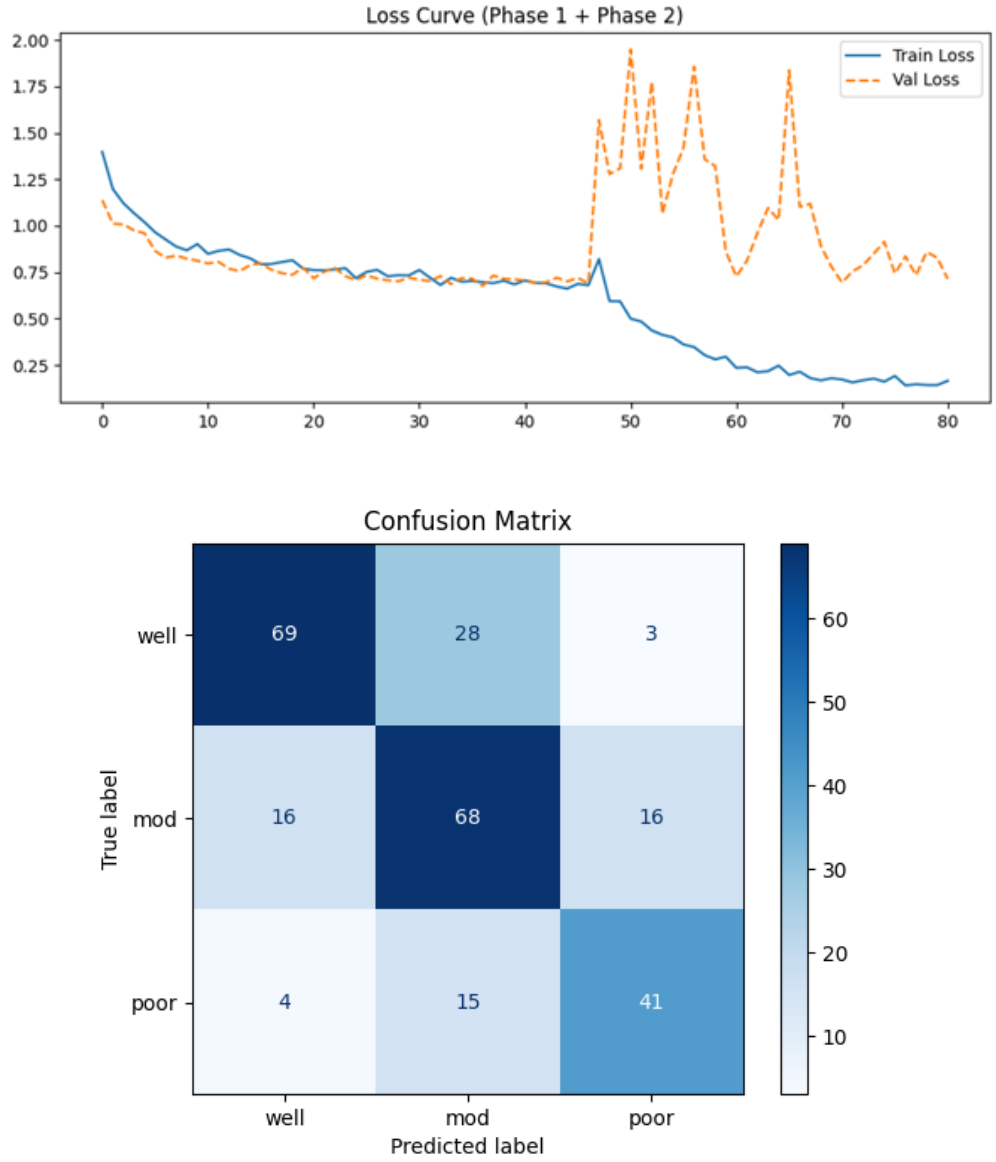


Fig 7.66. Acc, Loss curve & confusion matrix of Experiment 7

Key Findings

This experiment shows stable learning during Phase 1, where both training and validation accuracy increase gradually and track each other closely up to roughly epoch 45. Validation accuracy stabilizes around **0.68–0.72**, and both training and validation loss curves move in parallel, indicating healthy generalization before fine-tuning. However, once Phase 2 begins (at the moment deeper layers are unfrozen), the learning dynamics change sharply. Training accuracy continues rising rapidly toward **0.93+**, while validation accuracy collapses to **0.45–0.55** and becomes extremely unstable. This is a textbook sign of **severe overfitting and training-set memorization** triggered during fine-tuning.

The loss curves confirm this breakdown: at the Phase 2 boundary, validation loss jumps dramatically (spikes up to ~1.8–2.0), while training loss continues to decrease steadily to ~0.20. Despite ReduceLROnPlateau and EarlyStopping in Phase 1, the LR of **1e-4 is still too high for fine-tuning 40 unfrozen layers**, and dropout=0.3 was not sufficient to counteract the aggressive overfitting. Overall, while Phase 1 was successful, Phase 2 destabilized the model, leading to poor validation performance and strong divergence between training and validation behavior.

7.8.8. Results - Experiment 8

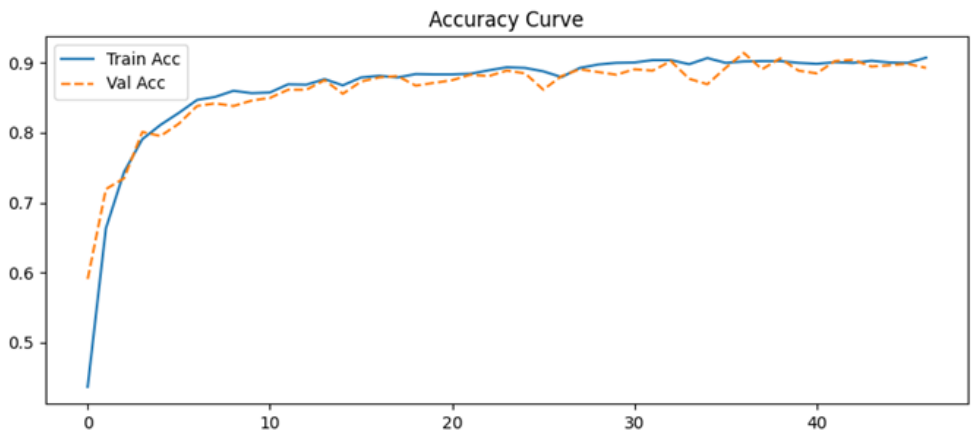
Experimental Setup

Parameter	Details
Classes	Well,mod,poor
Dataset size	well = 500, mod = 500, poor = 300
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-4
Add-ons	Edge enhanced feature map(embosed input)
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.9050
Val accuracy	0.8950

Accuracy and Loss curve



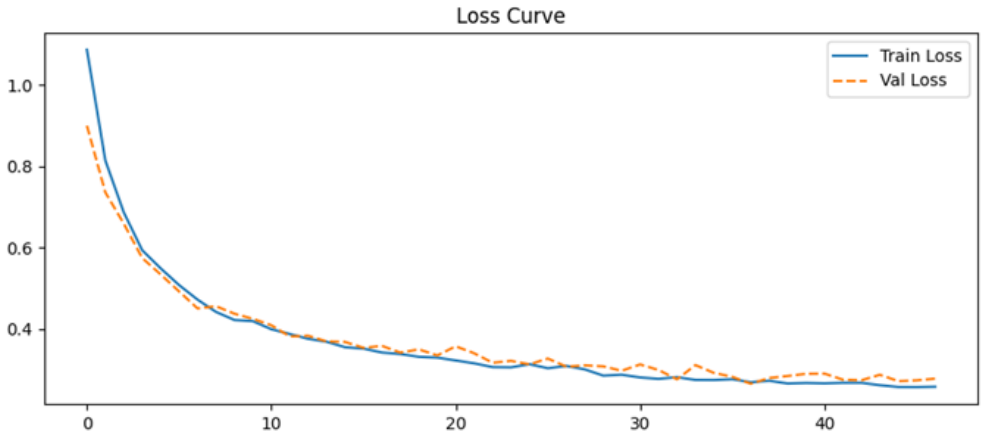


Fig 7.67. Acc and Loss curve of Experiment 8

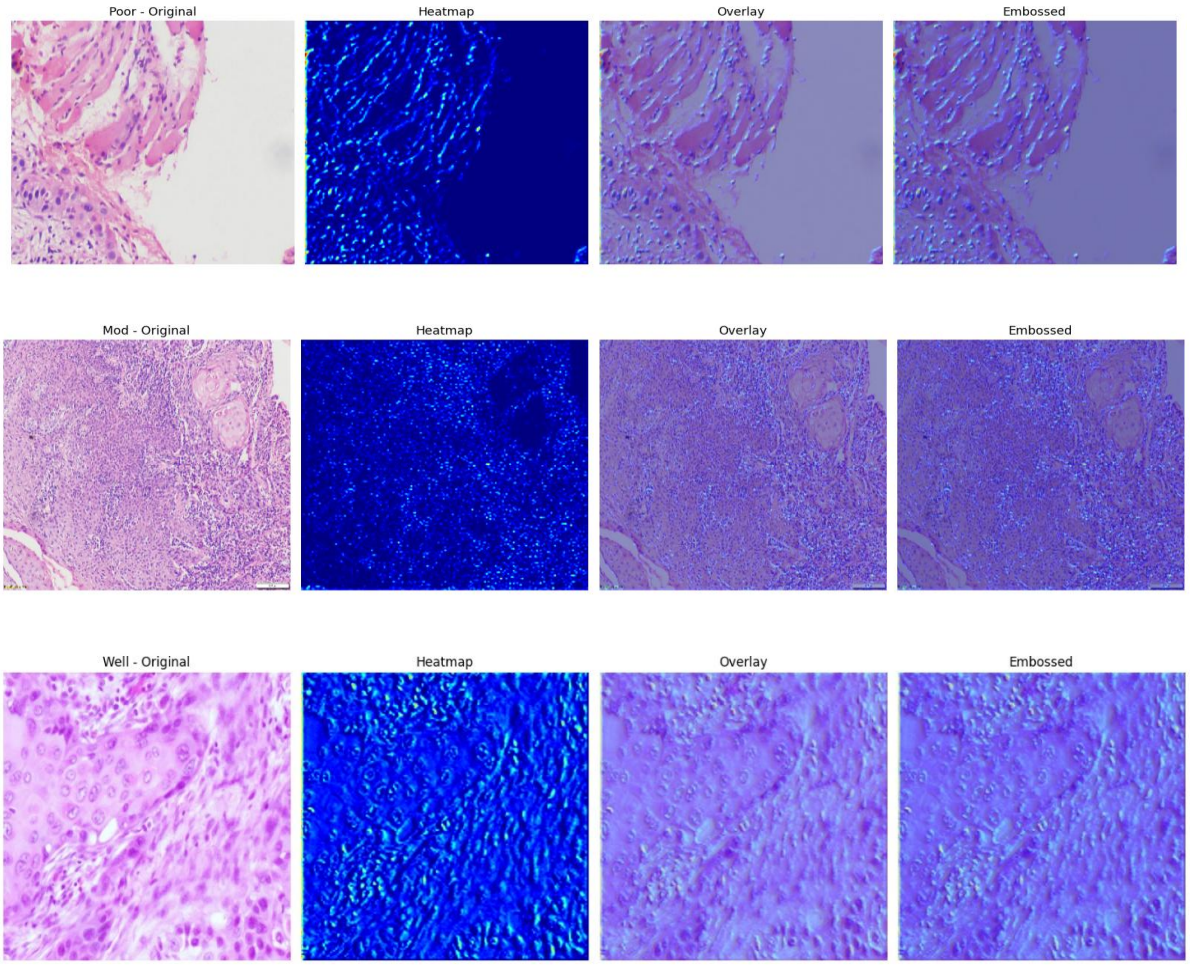


Fig 7.68 Edge enhanced feature map images

Key Findings

Using the edge-enhanced (embossed) images as input with LR = 1e-4 gives very strong and stable learning: both training and validation accuracy quickly rise above 0.85 and converge around **0.90**, with the curves almost overlapping. The loss curves show the same behaviour, smoothly decreasing and ending near **0.27–0.28** with almost no gap between train and validation. This means the model is **neither underfitting nor overfitting** and is generalizing extremely well on the embossed feature maps. The callbacks are doing their job: they slow the learning rate when val_loss stalls and stop training before any degradation, so you're essentially sitting at the "sweet spot" of performance.

Across all LR sweeps:

- **1e-3** → fast learning but clearer overfitting, lower final val accuracy.
- **5e-4 & 3e-4** → converge quickly but show a noticeable train–val gap (mild overfitting).
- **1e-5** → extremely stable but underfits; accuracy saturates much lower and loss stays high.
- **1e-4** → best balance: high final accuracy, smooth convergence, and **minimal train–val gap**.

With the embossed input + good callbacks, LR = 1e-4 gives you **the highest validation performance and the cleanest curves**, so it's the most reliable learning rate to move forward with as your main configuration.

7.8.9. Results - Experiment 9

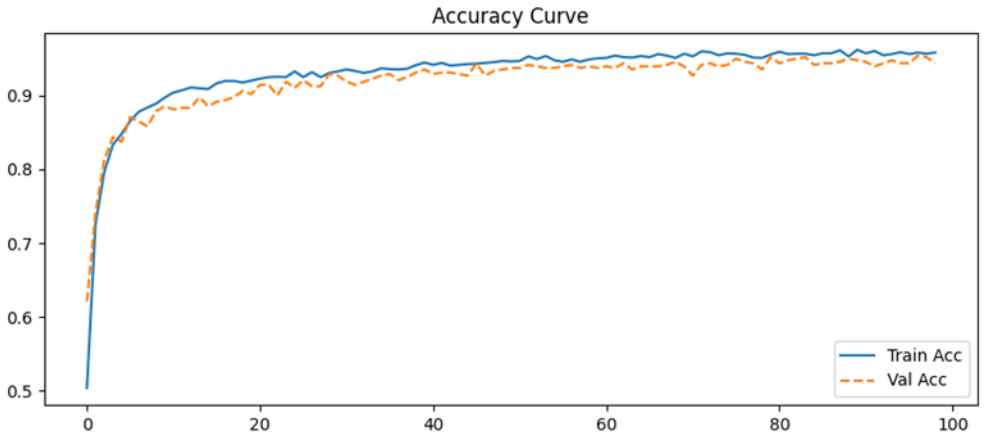
Experimental Setup

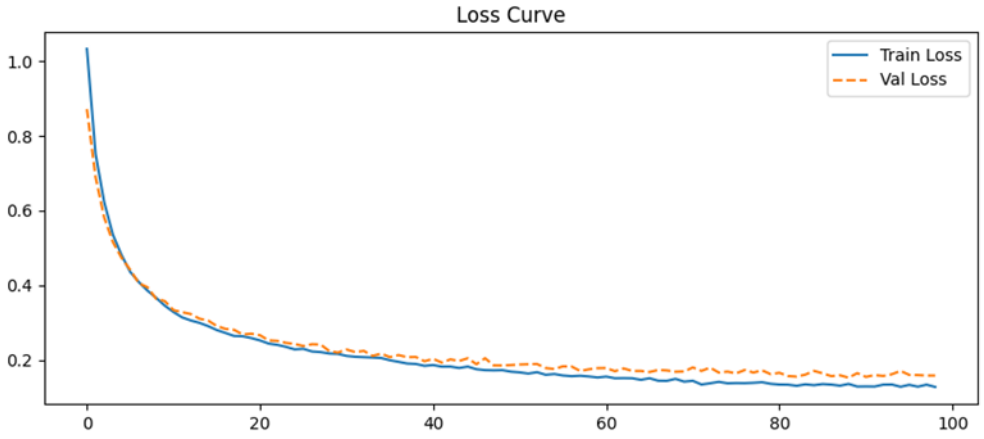
Parameter	Details
Image size	320 x 320
Classes	Well,mod,poor
Dataset size	well = 800, mod = 800, poor = 800
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-4
Add-ons	Edge enhanced feature map(embosed input)
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	Value
Train accuracy	0.9500
Val accuracy	0.9400

Accuracy and Loss curve





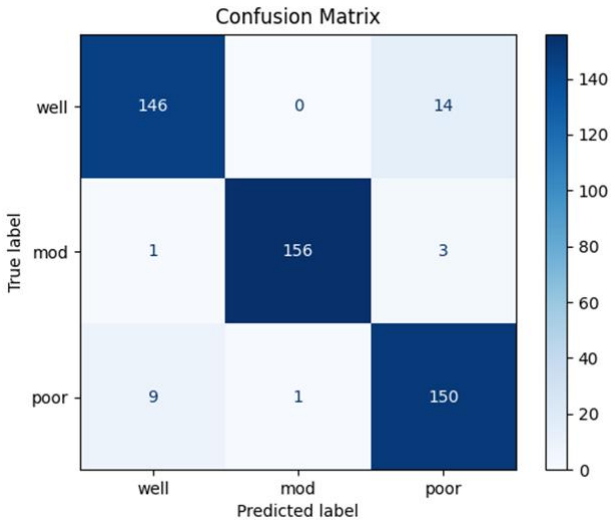


Fig 7.69. Acc, Loss curve & confusion matrix of Experiment 9

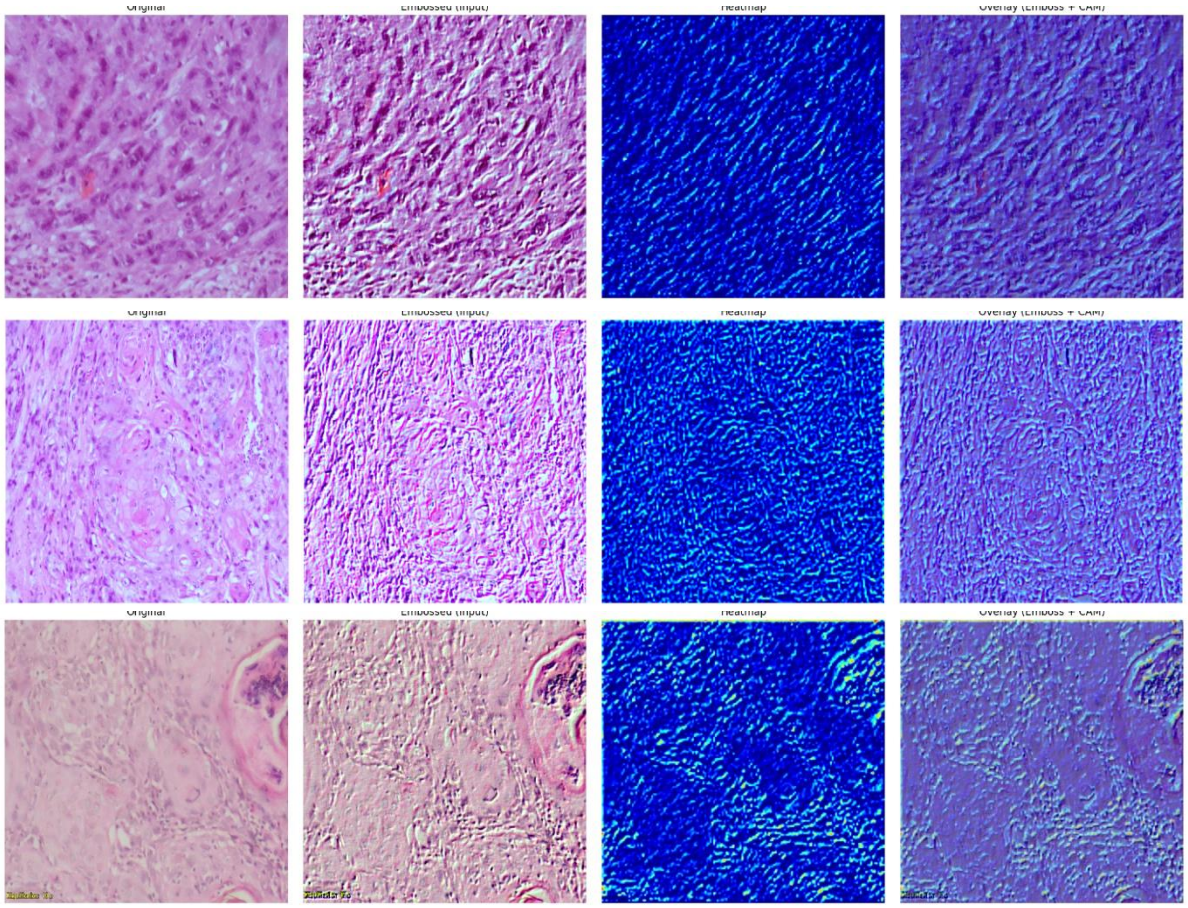


Fig 7.70. Edge Enhanced feature map images

Key Findings

Increasing the input size from **224×224 to 320×320** while keeping everything else fixed gives a clear performance boost. Both train and validation accuracy climb rapidly above 0.9 and then saturate around **95% / 94%**, with very small gaps in both accuracy and loss – so the model is **generalizing extremely well** and shows no meaningful overfitting or underfitting.

The smoother, lower loss curve suggests the model is exploiting the extra spatial resolution from 320×320 embossed images to learn more discriminative edge/texture patterns, especially useful for your edge-enhanced pipeline. In short: larger input size + LR=1e-4 + embossed features is **strongest configuration so far** in terms of both accuracy and training stability.

7.8.10. Results - Experiment 10

Experimental Setup

Parameter	**Details**
Image size	320 x 320
Classes	Well,mod,poor
Dataset size	well = 800, mod = 800, poor = 800
Augmentation	Rotation, Flip, Zoom
Optimizer	Adam
Learning Rate	1e-4
Add-ons	Gradcam
CallBacks	ReduceLROnPlateau, Early Stopping

Performance Metrics

Metrics	**Value**
Train accuracy	0.9800
Val accuracy	0.9600

Accuracy and Loss curve

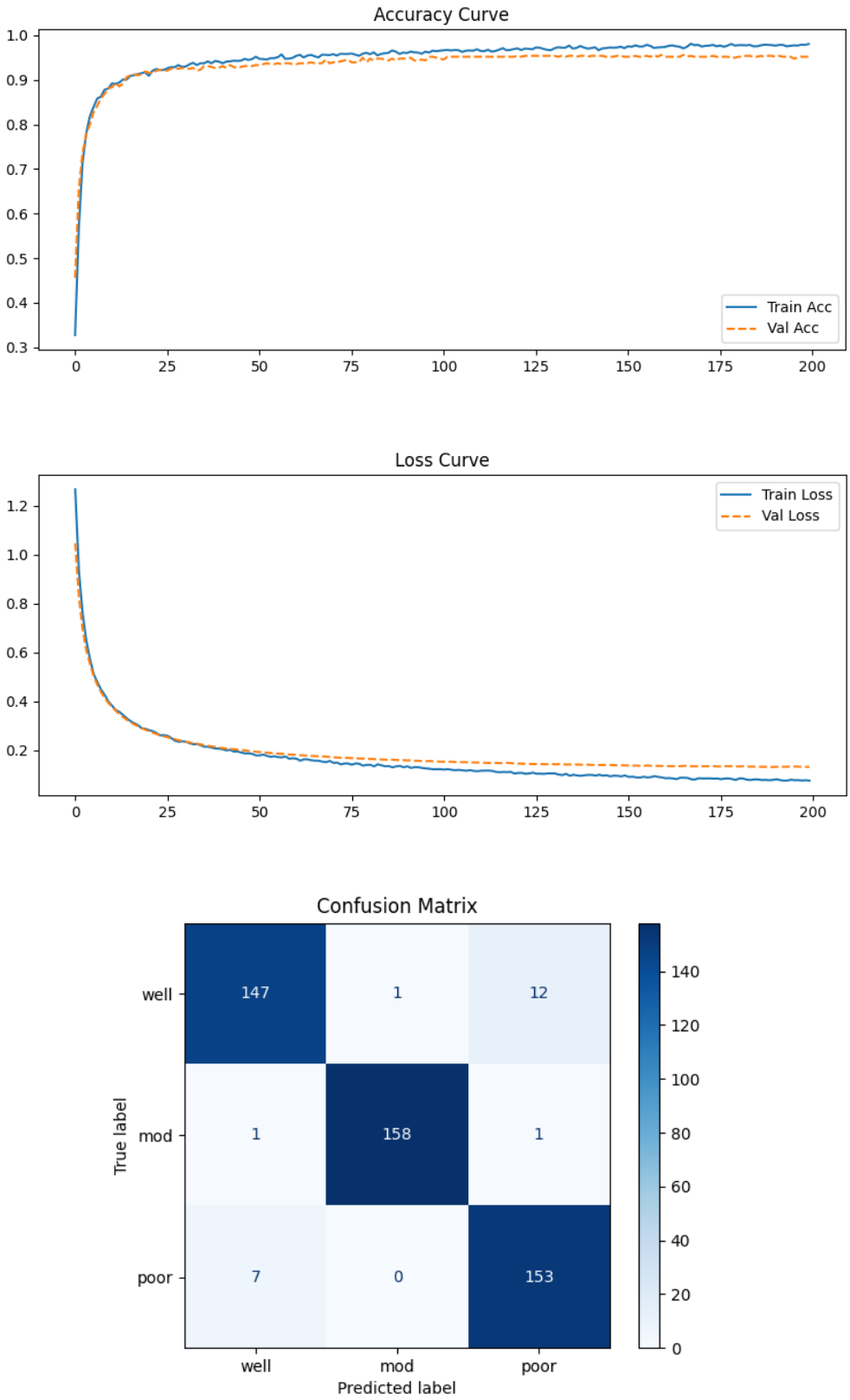


Fig 7.71. Acc, Loss curve & confusion matrix of Experiment 10

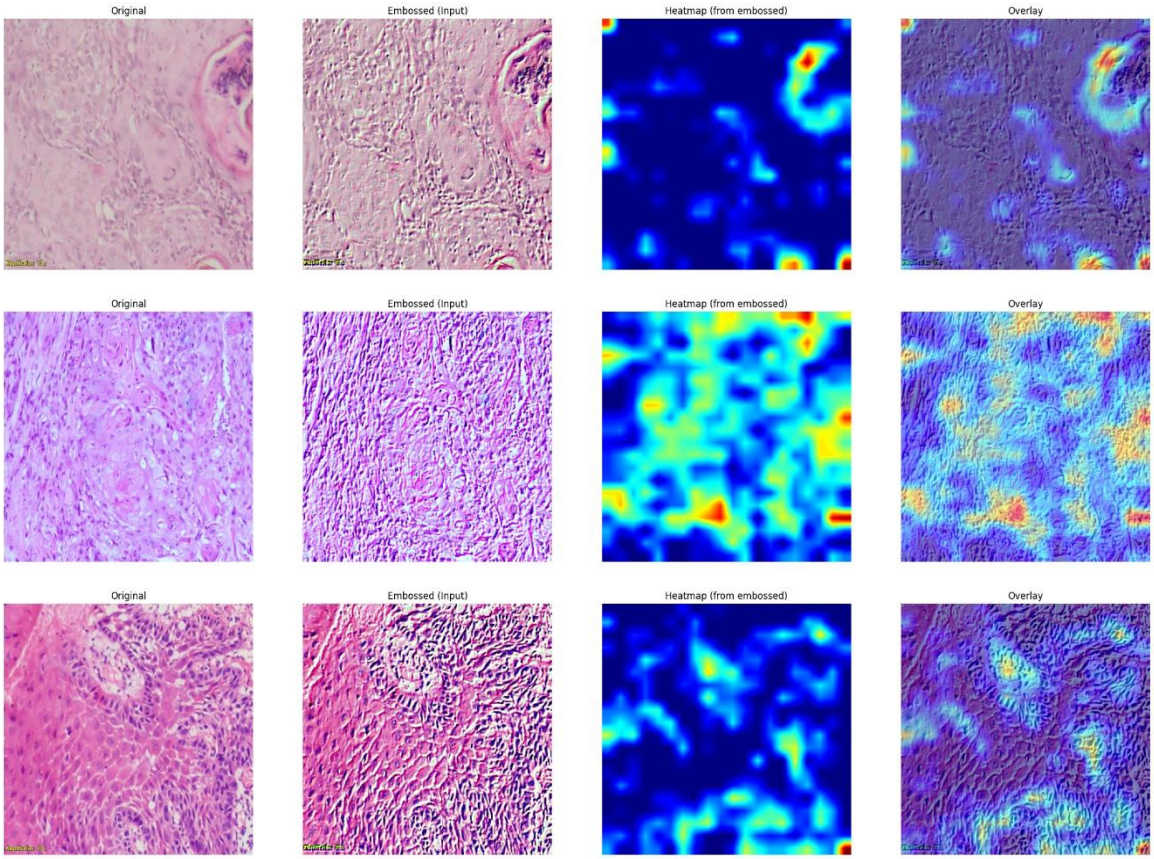


Fig 7.72 Gradcam images

Key Findings

The model achieves exceptionally strong and stable performance, with the accuracy curves showing smooth convergence toward **98% (train)** and **96% (validation)**. The curves overlap closely for the first half of training and maintain a consistent small gap afterward, reflecting excellent generalization and minimal overfitting. The validation accuracy stabilizes early (around epoch 20–30) and remains highly consistent even out to epoch 200, indicating that the model is capturing highly discriminative edge/texture features from the 320×320 embossed images.

Similarly, the loss curves steadily decrease and flatten out with very small gaps, ending near **0.07 (train)** and **0.13 (validation)**. This suggests extremely efficient optimization with no instability, divergence, or signs of underfitting. The behavior is even smoother than the previous 320×320 experiment, and Grad-CAM visualization does **not** harm training—your model is performing at an optimal level.

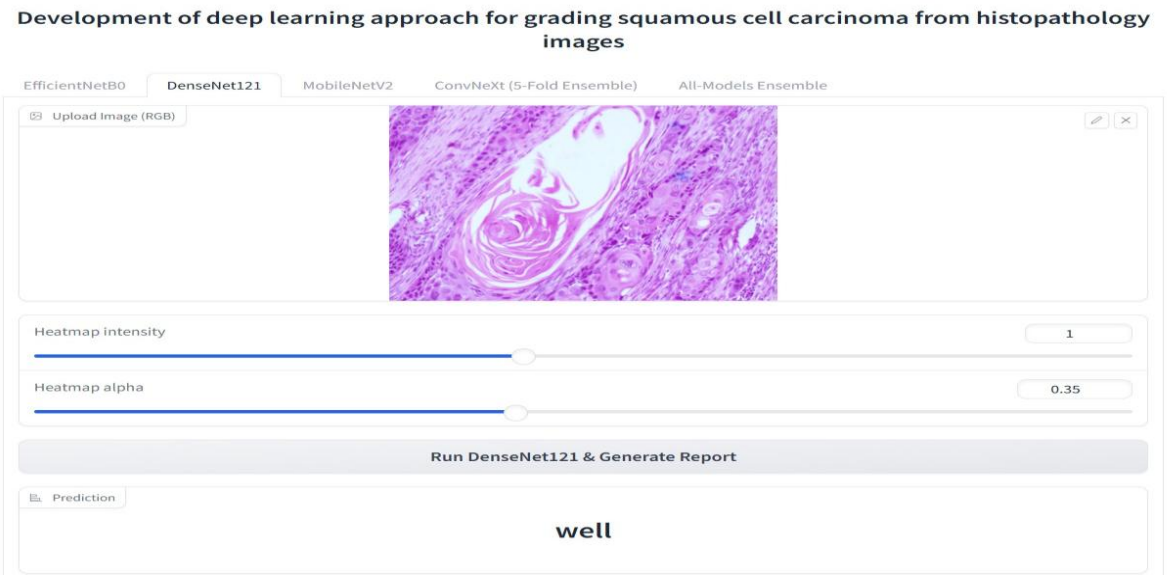
8. USER INTERFACE FOR THE MINI PROJECT

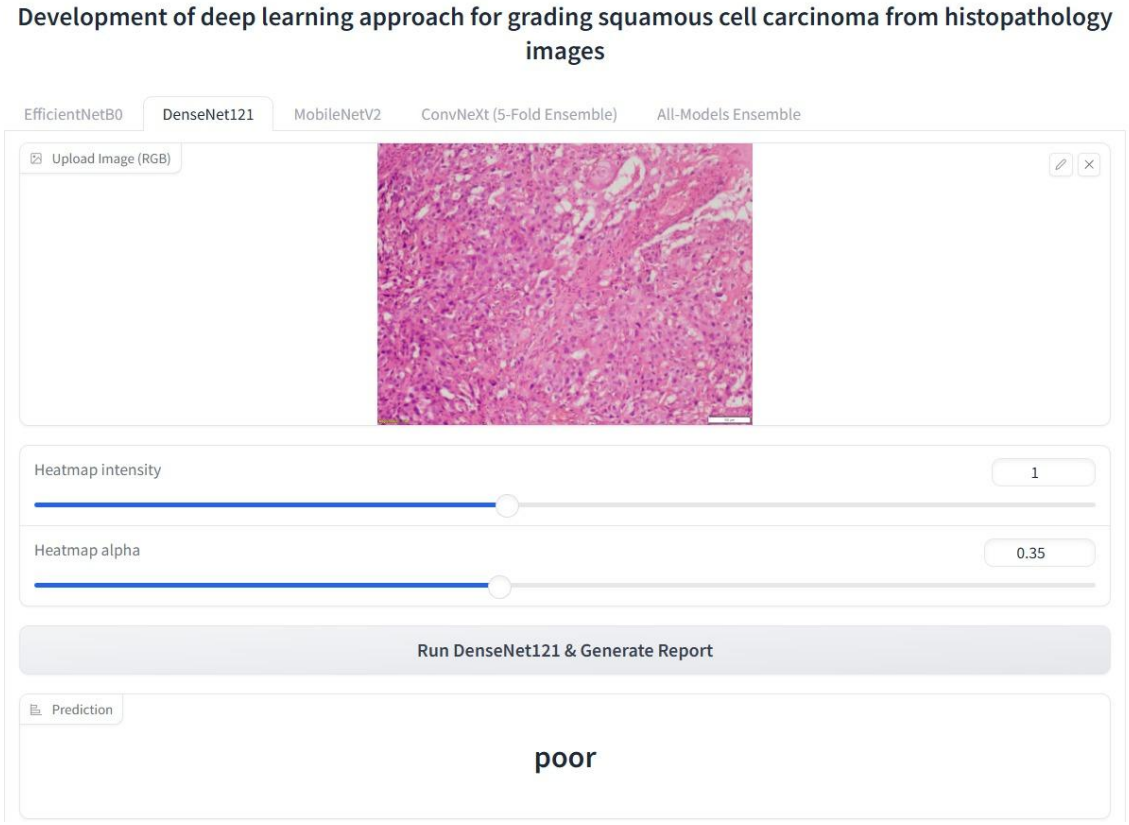
8.1 Overview

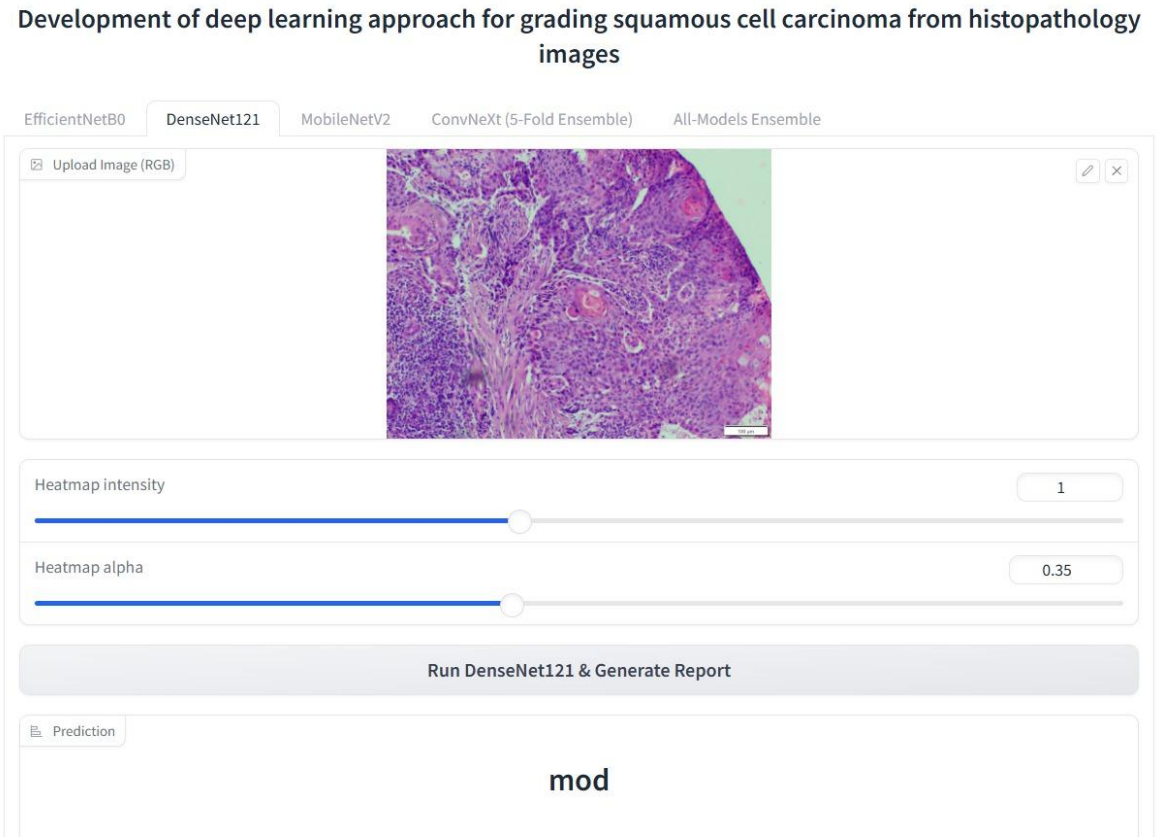
A dedicated and stable graphical user interface (GUI) was developed to enable users to upload histopathology images, run predictions using multiple deep learning models, visualize Grad-CAM heatmaps, and generate structured PDF reports. The interface is built using the Gradio framework and provides an intuitive and compact layout suitable for offline/clinical use.

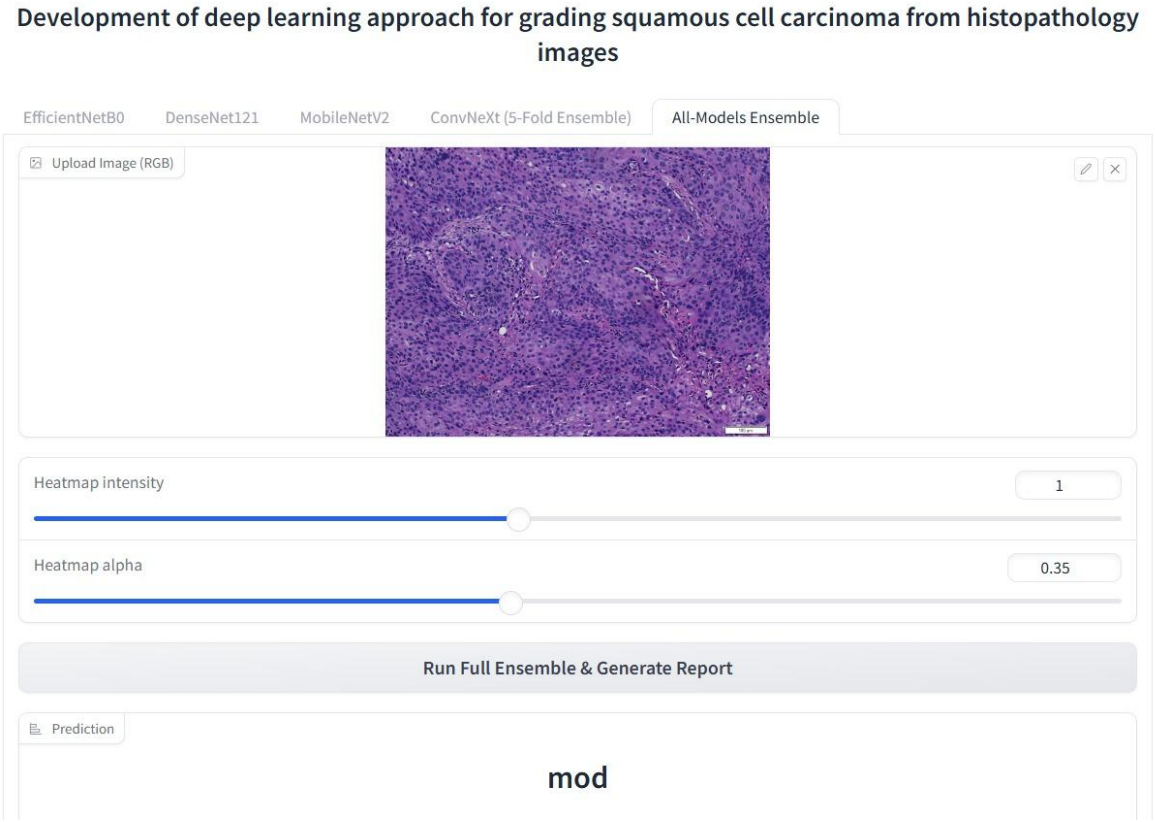
The UI is designed to be fully model-agnostic, modular, and error-tolerant, ensuring no crashes even when predictions contain NaN values or corrupted inputs.

The user interface follows a streamlined workflow beginning with image upload, where the system auto-corrects input formats and applies model-specific preprocessing (224×224 for EfficientNet, DenseNet, MobileNet; 260×260 for ConvNeXt). Users can select from EfficientNetB0, DenseNet121, MobileNetV2, the ConvNeXt 5-fold ensemble, or an all-models ensemble, after which the chosen model generates SCC predictions (well/moderate/poor) along with normalized class probabilities. A robust Grad-CAM engine then produces heatmaps, overlays, and embossed visualizations. A fully formatted PDF report is generated containing the model name, prediction summary, probabilities, and all Grad-CAM outputs. The backend includes a unified model loader with safe_load(), custom object support for ConvNeXt, integrity checks, and graceful handling of missing files. Preprocessing functions ensure clean RGB inputs, while the ConvNeXt ensemble and all-models ensemble modules perform probability averaging, skip faulty folds, normalize outputs, and select the best fold for Grad-CAM. The Gradio-based UI uses a clean centered layout with tabs for each model, providing sliders for CAM intensity/alpha, prediction outputs, visualizations, and PDF download options. Comprehensive safety mechanisms—such as NaN correction, probability renormalization, model-failure detection, fallback predictions, and safe image resizing—ensure robust operation without crashes, making the system reliable for clinical-grade usage.


Development of deep learning approach for grading squamous cell carcinoma from histopathology images
EfficientNetB0
DenseNet121
MobileNetV2
ConvNeXt (5-Fold Ensemble)
All-Models Ensemble
Upload Image (RGB)
Heatmap intensity
1
Heatmap alpha
0.35
Run DenseNet121 & Generate Report
Prediction
well


Development of deep learning approach for grading squamous cell carcinoma from histopathology images
EfficientNetB0
DenseNet121
MobileNetV2
ConvNeXt (5-Fold Ensemble)
All-Models Ensemble
Upload Image (RGB)
Heatmap intensity
1
Heatmap alpha
0.35
Run DenseNet121 & Generate Report
Prediction
poor


Development of deep learning approach for grading squamous cell carcinoma from histopathology images
EfficientNetB0
DenseNet121
MobileNetV2
ConvNeXt (5-Fold Ensemble)
All-Models Ensemble
Upload Image (RGB)
Heatmap intensity
1
Heatmap alpha
0.35
Run DenseNet121 & Generate Report
Prediction
mod


Development of deep learning approach for grading squamous cell carcinoma from histopathology images
EfficientNetB0
DenseNet121
MobileNetV2
ConvNeXt (5-Fold Ensemble)
All-Models Ensemble
Upload Image (RGB)
Heatmap intensity
1
Heatmap alpha
0.35
Run Full Ensemble & Generate Report
Prediction
mod

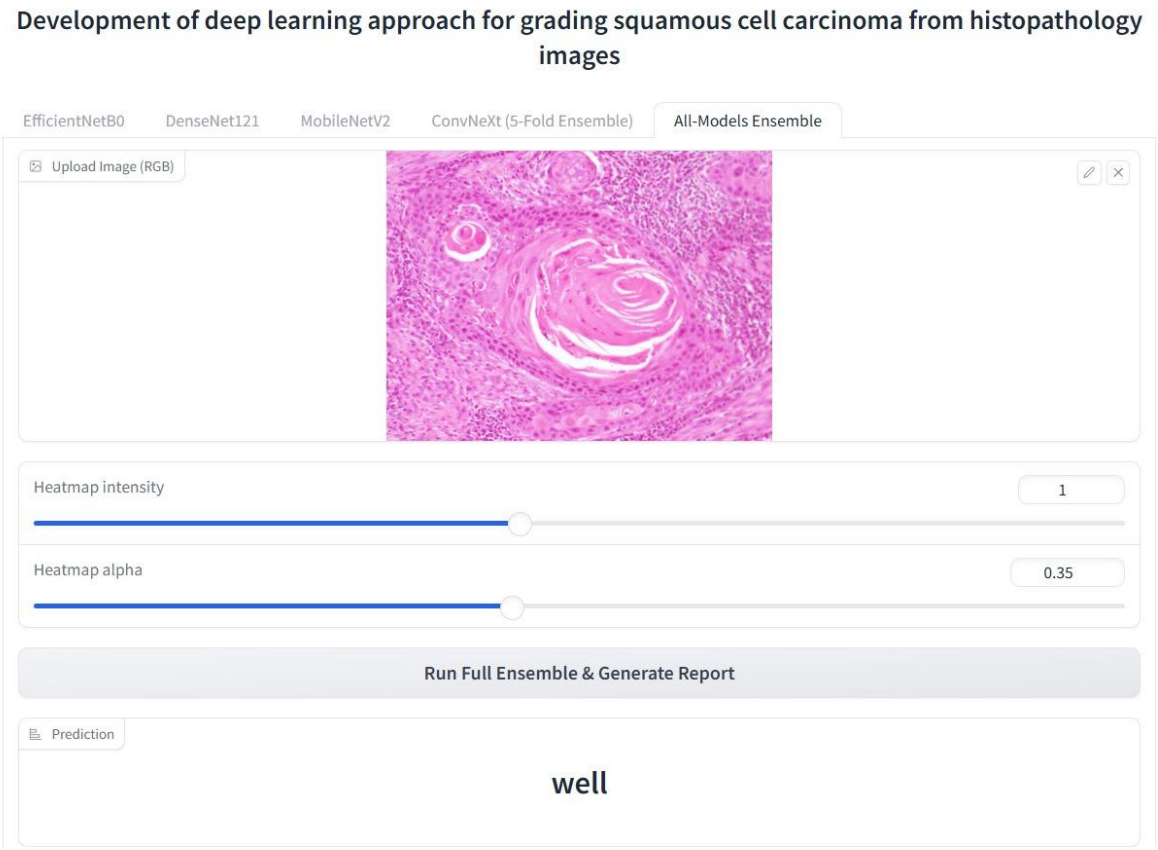


Fig 8.1. Screenshots of the User Interface

9. CONCLUSIONS

Model Performance and Key Observations

Across all evaluated architectures, InceptionV3 emerged as the top-performing model, achieving a mean validation accuracy of 99.72% with the Adam optimizer and 99.66% with RMSprop. Its exceptionally low standard deviation highlights strong reliability and makes it highly suitable for clinical applications. DenseNet121 also delivered strong results with 98.09% accuracy on the cleaned dataset, though it was more sensitive to data quality issues such as white-patch artifacts. Models like ConvNeXtTiny and MobileNetV2 showed competitive performance as well, with ConvNeXt improving significantly after dataset refinement and MobileNetV2 demonstrating strong accuracy despite signs of overfitting.

Effect of Hyperparameters and Data Quality

The experiments clearly show that systematic hyperparameter tuning plays a major role in performance optimization. Learning rate scheduling, appropriate optimizer selection, extended training with early stopping, and the use of pretrained ImageNet weights all contributed to smoother convergence and higher accuracy across models. Equally important was dataset quality—manual removal of low-quality images and proper preprocessing resulted in more stable training dynamics and significantly improved validation results. Data augmentation further enhanced model generalization by introducing variability that closely matched real-world conditions.

Overall Findings and Insights

The project demonstrates that transfer learning is highly effective for histopathology image classification, enabling excellent performance even with limited training data. The use of 5-fold cross-validation ensured unbiased evaluation and confirmed that the models generalized well across different subsets. A key takeaway is that high-quality curated data often has a greater impact on performance than increasing model complexity. Ultimately, selecting an appropriate architecture requires balancing accuracy, computational efficiency, and training stability based on deployment needs.

10. REFERENCES

1. J. Musulin, D. Štifanić, A. Zulijani, S. Baressi Šegota, I. Lorencin and Z. Car, "Automated Grading of Oral Squamous Cell Carcinoma into Multiple Classes Using Deep Learning Methods," in *Proc. IEEE Int. Conf. on Bioinformatics and Bioengineering (BIBE)*, 2021, pp. 1–6.
2. P. Patharia and P. K. Sethy, "LungCarcinoGrade-EffNetSVM: A Novel Approach to Lung Carcinoma Grading Using EfficientNet-B0 and SVM," in *Proc. Int. Conf. on I-SMAC*, 2024, pp. 1–7.
3. P. K. Sethy, A. G. Devi, B. Padhan, S. K. Behera, S. Sreedhar and K. Das, "Lung Cancer Histopathological Image Classification Using Wavelets and AlexNet," *Journal of X-Ray Science and Technology*, vol. 31, no. 1, pp. 45–54, 2023.
4. A. Kumar and L. Nelson, "Enhancing Oral Squamous Cell Carcinoma Detection Using EfficientNet-B3 from Histopathologic Images," *Journal of Medical Imaging and Health Informatics*, vol. 15, no. 1, pp. 120–128, 2025.
5. A. Kumar and V. Kumar, "Histopathological Image-Based Oral Squamous Cell Carcinoma Classification Using Deep Network Fusion," in *Proc. IEEE UPCON*, 2023, pp. 850–856.
6. Z. Kang, M. Chen, H. Zhang and X. Liao, "EsccNet: A Hybrid CNN and Transformer Model for Classification of Whole Slide Images of Esophageal Squamous Cell Carcinoma," in *Proc. Int. Workshop on Computational Pathology*, 2024, pp. 1–8.
7. S.-Y. Park, G. Ayana and S.-W. Choe, "Squamous Cell Carcinoma Margin Classification Using Vision Transformers from Digital Histopathology Images," *Diagnostics*, vol. 15, no. 3, pp. 1–15, 2025.
8. K. R. Lathakumari, A. C. Ramachandra, U. C. Avanthi, C. B. Ronald and T. Bhavatharani, "Classification of Non-Small Cell Lung Cancer Using Deep Learning," in *Proc. Int. Conf. on Intelligent Computing and Control Systems*, 2023, pp. 300–305.
9. C. Yang, X. Yu, H. Yang, Z. An, C. Yu, L. Huang and Y. Xu, "Multi-Teacher Knowledge Distillation with Reinforcement Learning for Visual Recognition," in *Proc. AAAI Conf. Artificial Intelligence*, 2025, pp. 1234–1242.
10. A. H. Salamah, S. M. Hamidi and E.-H. Yang, "A Coded Knowledge Distillation Framework for Image Classification Based on Adaptive JPEG Encoding," *Pattern Recognition*, vol. 158, pp. 110966-110975, 2025.
11. Y. Song, A. Song, J. Wang, Y. Ge and L. Li, "Multiple Teachers Are Beneficial: A Lightweight and Noise-Resistant Student Model for Point-of-Care Imaging Classification," *Expert Systems with Applications*, vol. 235, pp. 127145-127160, 2025.
12. S. A. N. Raju, K. Venkatesh, R. K. Gatla, E. Prasad Konakalla, M. M. Eid, N. Titova, S. M. S. Ghoneim and R. N. R. Ghaly, "Colorectal Cancer Detection with a Hybrid Supervised and Unsupervised Learning Approach," *Scientific Reports*, vol. 15, no. 1, pp. 1–12, 2025.
13. Various Authors, "Deep Network Fusion and Ensemble Methods for Histopathology Image Classification: A Review," *IEEE Reviews in Biomedical Engineering*, vol. 17, pp. 220–240, 2023.
14. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2019, pp. 6105–6114.