DermaMNIST Classification Pipeline

Binary Classification: Malignant vs Benign

1. Clinical Motivation

False Negative (FN) = Missed Cancer

Model predicts "Benign" but patient has cancer
DANGEROUS — Patient leaves untreated

False Positive (FP) = Extra Biopsy

Model predicts "Malignant" but patient is healthy
ACCEPTABLE — Extra follow-up only

Project Goal: Minimize FN (Missed Cancers) → Maximize Recall

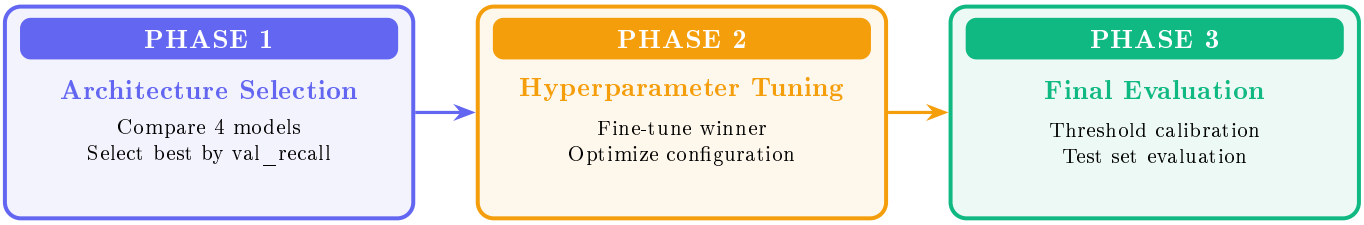
2. Evaluation Metrics

Primary Metric: Recall (Sensitivity)

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Detected Cancers}}{\text{All Actual Cancers}}$$

Precision: $\frac{TP}{TP+FP}$ **F1 Score:** $2 \cdot \frac{P \times R}{P+R}$ **AUC:** Area Under ROC

3. Three-Phase Experiment Design



Experiment Configuration

4. Data Preprocessing

Dataset: DermaMNIST (10,015 images, 28×28)
Original: 7 classes → **Binary:** 2 classes
Normalization: [0, 255] → [0, 1]

Labels: One-hot encoded
Class Imbalance: ∼9:1 (Benign:Malignant)
Split: Train / Validation / Test

7-Class to Binary Mapping:

Original Class	Description	ID	→	Binary Class	Total
nv (Melanocytic nevi)	Benign mole	0			
bkl (Benign keratosis)	Seborrheic keratosis	2			
vasc (Vascular lesions)	Blood vessel lesions	5	→	**Class 0: Benign**	**5,789**
df (Dermatofibroma)	Benign skin lesion	6		(82.6%)	
mel (Melanoma)	Malignant skin cancer	1			
bcc (Basal cell carcinoma)	Common skin cancer	3	→	**Class 1: Malignant**	**1,218**
akiec (Actinic keratoses)	Pre-cancerous	4		(17.4%)	

Note: Class imbalance ratio = 5,789 : 1,218 ≈ 4.8:1 (addressed via class weights)

5. Training Settings

Optimizer: Adam
Loss: Categorical Crossentropy
(binary as 2-class: softmax + one-hot labels)
Monitor: val_recall (mode='max')

Epochs: 30–50
Patience: 10–15
Batch Size: 32–64
Callbacks: EarlyStopping, ModelCheckpoint

6. Data Augmentation

Rotation: ±20° **Width/Height Shift:** 15% **Zoom:** 10% **Flip:** Horizontal + Vertical

7. Class Weight Strategies

Code	Weights	Description	Effect
W0	None	No class balancing	Baseline
W3	{0:1, 1:3}	3× penalty for missing malignant	Moderate recall boost
W5	{0:1, 1:5}	5× penalty (matches ∼5:1 ratio)	Higher recall boost

8. Critical Design Decision: Monitoring Metric

Why Monitor Recall, NOT Loss?

Problem with Loss:
Class weights multiply the loss for misclassifying malignant samples.
→ **Higher loss is expected!**
→ Loss no longer reflects model quality

Solution:
Monitor **val_recall** instead of val_loss for EarlyStopping and ModelCheckpoint.
→ Stop when recall stops improving
→ Save model with best recall

Callbacks: monitor='val_recall', mode='max' (not 'val_loss', mode='min')

Block 3: Model Architectures

9. Custom CNN (Baseline)

Architecture: Input → [Conv2D → BatchNorm → MaxPool → Dropout]×depth → Flatten → Dense → Softmax
Key Features: Progressive dropout (increases with depth), Filters double each block

Config	filters	depth	dropout	dense	lr	batch	Hypothesis
CNN_v1	32	3	0.5	512	0.001	64	Baseline small
CNN_v2	64	4	0.4	256	0.001	64	Deeper, wider
CNN_v3	128	4	0.3	512	0.0005	32	High capacity

10. Transfer Learning Models

Input Adaptation: 28×28 $\xrightarrow{\text{UpSampling2D}}$ 56×56 → Pre-trained Base (ImageNet) → Global Avg Pool → Custom Head
Custom Head: Dropout → Dense + BatchNorm + ReLU → Dropout → Dense(2, softmax)

ResNet50 (23.5M params, 175 layers, skip connections)

Config	unfreeze	dropout	dense	lr	Hypothesis
ResNet_v1	0 (frozen)	0.5	256	0.001	Feature extraction only
ResNet_v2	20	0.5	256	0.0001	Partial fine-tuning
ResNet_v3	None (all)	0.5	256	0.0001	Full fine-tuning
ResNet_v4	None (all)	0.7	128	0.00005	Anti-overfit + slow

VGG16 (14.7M params, 19 layers, sequential 3×3 filters)

Config	unfreeze	dropout	dense	lr	Hypothesis
VGG_v1	0 (frozen)	0.5	256	0.001	Feature extraction
VGG_v2	4	0.5	256	0.0001	Last conv block
VGG_v3	None (all)	0.6	512	0.00005	Full fine-tune

EfficientNetB0 (4.0M params, 237 layers, MBConv + squeeze-excitation)

Config	unfreeze	dropout	dense	lr	Hypothesis
EffNet_v1	0 (frozen)	0.5	256	0.001	Feature extraction
EffNet_v2	30	0.4	256	0.0001	Partial fine-tuning
EffNet_v3	None (all)	0.5	128	0.00005	Full fine-tune

Phase 2: Hyperparameter Tuning

11. Phase 2 Overview

Input: Best architecture from Phase 1
Goal: Optimize fine-tuning strategy
Selection: Best val_recall

Data Used:
- Training set (with augmentation)
- Validation set (for monitoring)

12. Fine-Tuning Strategies

Strategy	unfreeze_layers	Key Change	Hypothesis
Freeze10	10	Only top 10 layers trainable	Conservative, stable
Freeze20	20	Top 20 layers trainable	Balanced approach
HighDropout	None (all)	Full fine-tune + dropout=0.7	Anti-overfitting
LowLR	None (all)	Full fine-tune + lr=0.00005	Stability focus

Phase 3: Final Evaluation

13. Phase 3 Overview

Input: Best model from Phase 2
Goal: Final unbiased evaluation
Output: Final metrics + confusion matrix

Data Used:
- Validation set (threshold tuning)
- **Test set (final evaluation only!)**

14. Evaluation Pipeline

15. Threshold Calibration

How Threshold Calibration Works

Default: Threshold = 0.5

If P(Malignant) $\geq 0.5 \rightarrow$ Predict Malignant
If P(Malignant) $< 0.5 \rightarrow$ Predict Benign

Lower Threshold (e.g., 0.3):

$\rightarrow$ More samples predicted as Malignant
$\rightarrow$ Higher Recall (catch more cancers)
$\rightarrow$ Lower Precision (more false alarms)

Strategy: Adjust threshold to optimize Recall-Precision trade-off for clinical needs

IMPORTANT: Test set is used ONLY in Phase 3 —
never for training or model selection (prevents data leakage)