AI Breast Cancer Detection

ML Current Stage

DigitalSweep

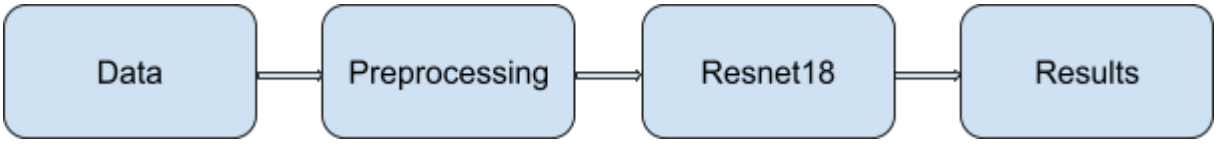
Sharp Data for Sharp Decisions

Table of Contents

Overview

This document describes the current machine learning pipeline for breast cancer detection using thermal imaging in combination with a pre-trained ResNet18 model.

The workflow follows a clinically oriented image classification approach, with a strong focus on preventing data leakage, quantifying prediction uncertainty, and generating artifacts suitable for managerial review.



Dataset

```
/MyDrive/DMR-IR/
    healthy/   # images labeled "healthy"
    sick/      # images labeled "sick"
```

This dataset comprises thermal images collected for the purpose of breast cancer detection through non-invasive thermography.

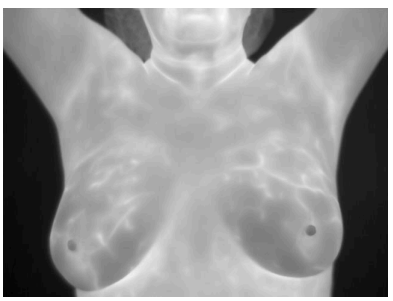
The images are categorized into two classes : **healthy** and **sick**. Each class folder contains preprocessed thermal images resized for deep learning tasks such as classification and segmentation.

Note: This is preprocessed DMR-IR dataset :

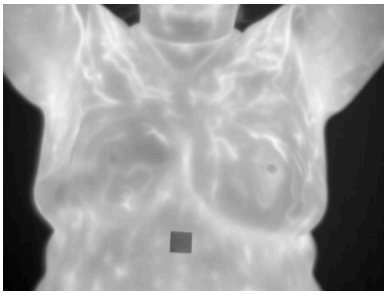
- **Balanced** (380 sick – 380 healthy) → no need for oversampling, undersampling, or class-weighted loss.
- **Normalized** → pipeline **skips global normalization** (we only convert to float [0..1] when building tensors).
- **No patient IDs** → can't do patient-aware splitting yet, but we'll keep that in mind for future work.
- **No U-Net weights** available → no segmentation is done for ROI.

Dataset Images example

Healthy



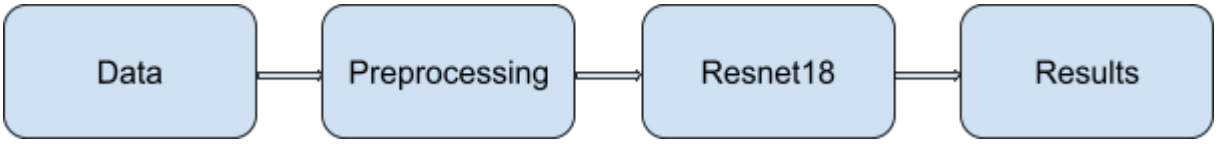
Sick



Preprocessing

- **Normalization** skipped.
- **Dataset Balancing** skipped.
- **Grayscale to RGB** : Convert to 3-channel format to match the original ResNet-18 training.
- **Resize & pad** to fixed input size (e.g., 224×224 or 512×512 depending on GPU memory).
- **Data Split:** The dataset was divided into a **training set (80%)** and a **validation set (20%)** to ensure the model's performance could be accurately evaluated on unseen data.

Model: ResNet18 (pretrained, fine tuned)



ResNet-18 :

- Robust **CNN** architecture → efficient on smaller datasets.
- **Pre-trained model (ImageNet)** to capture basic visual features using **Transfer Learning**.
- Faster convergence and improved performance over training from scratch.
- Strong feature extraction capability.
- Interpretability with Grad-CAM.
- Final layer modified for our binary classification task. (Replace final FC with single logit / two outputs)
- Classify images as 'Healthy' or 'Sick' (absence or presence of breast cancer).

Training Pipeline

- **Loss Function:** CrossEntropyLoss.
- **Optimizer:** The **Adam optimizer** (LR = 5e-5) was used to adjust the model's parameters and find the most efficient way to reduce the loss.
- **Batch Size:** The model was trained using a **batch size of 16**, which is a balanced approach for memory usage and training speed.
- **Epochs:** The model was trained for **20 epochs** to allow it to learn from the entire dataset multiple times.
- **Early Stopping:** To prevent **overfitting**, we implemented a key feature called **early stopping**. If the validation loss did not improve for three consecutive epochs, the training was halted, and the best-performing model was saved.

Initial Results

The best model was **saved at the 5th epoch**, corresponding to the lowest **validation loss (0.0134)** and consistently high validation **accuracy (99.32%).**

Below figure illustrates the training vs. validation accuracy and loss across 8 epochs out of 20. The model demonstrates rapid convergence, with training loss decreasing steadily and validation loss stabilizing after the second epoch.

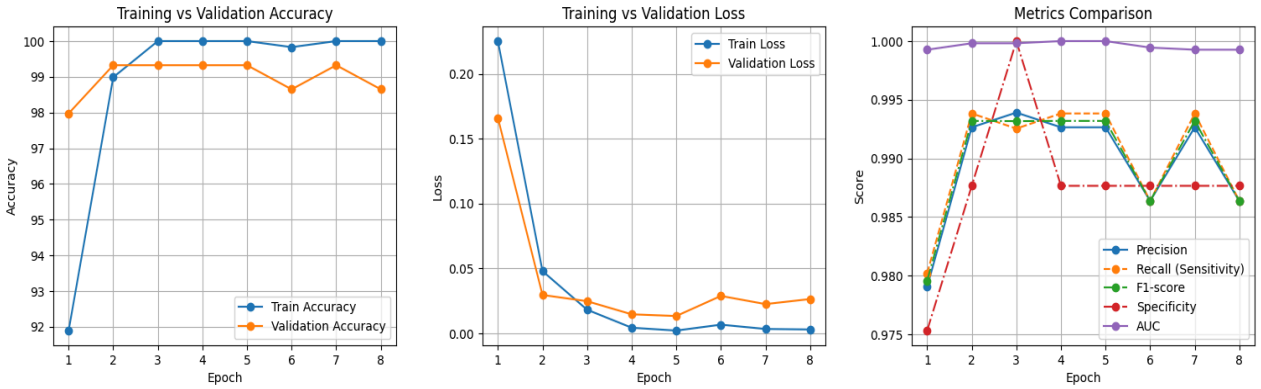


Figure 1 : Training vs Validation Graph

Below table data shows the results obtained during the training phase of the model

Epoch	Train Loss	Train Acc (%)	Val Loss	Val Acc (%)	Precision	Recall (Sens)	Specificity	F1	AUC	Best Model
1	0.2252	91.89	0.1660	97.97	0.9790	0.9802	0.9753	0.9796	0.9993	Saved model
2	0.0480	98.99	0.0296	99.32	0.9926	0.9938	0.9877	0.9932	0.9998	Saved model
3	0.0183	100.00	0.0248	99.32	0.9939	0.9925	1.0000	0.9932	0.9998	Saved model
4	0.0044	100.00	0.0147	99.32	0.9926	0.9938	0.9877	0.9932	1.0000	Saved model
5	0.0022	100.00	0.0134	99.32	0.9926	0.9938	0.9877	0.9932	1.0000	Saved model
6	0.0068	99.83	0.0289	98.65	0.9864	0.9864	0.9877	0.9864	0.9994	-
7	0.0035	100.00	0.0226	99.32	0.9926	0.9938	0.9877	0.9932	0.9993	-
8	0.0031	100.00	0.0265	98.65	0.9864	0.9864	0.9877	0.9864	0.9993	Early Stopping Triggered

Table 1 : Training vs Validation Results

On unseen data, after reloading the saved model, the ROC curve demonstrated the trade-off between sensitivity (true positive rate) and 1 - specificity (false positive rate) across multiple probability thresholds. Although the confusion matrix indicated a single false positive (a healthy case misclassified as sick), the ROC curve yielded an AUC of 1.0. This is not contradictory, as AUC evaluates the model's ability to rank positive cases higher than negative ones across all thresholds, rather than performance at a fixed cutoff. In this instance, the model consistently assigned greater probability scores to sick cases compared to healthy cases, resulting in perfect ranking separability despite a threshold-dependent misclassification.

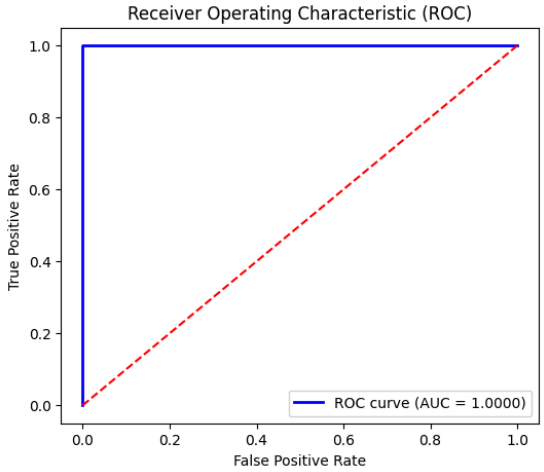


Figure 2 : ROC Graph

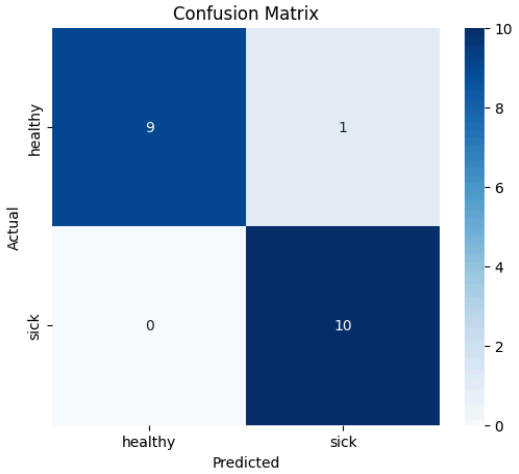


Figure 3: Confusion Matrix

On unseen data the test set confusion matrix revealed strong classification performance:

- True Positives (TP) : 10 → Sick correctly predicted as sick.
- True Negatives (TN) : 9 → Healthy correctly predicted as healthy.
- False Positives (FP) : 0 → No healthy misclassified as sick.
- False Negatives (FN) : 1 → One healthy case misclassified as sick.

From these values, the following metrics were obtained:

- **Accuracy** = (TP + TN) / Total = (10 + 9) / 20 = **95%**
- Precision (Sick class) = TP / (TP + FP) = 10 / (10+0) = 100%
- Recall (Sick class / **Sensitivity**) = TP / (TP + FN) = 10 / (10+1) ≈ **90.9%**
- **Specificity** (Healthy class) = TN / (TN + FP) = 9 / (9+0) = **100%**
- F1-score = 2 × (Precision × Recall) / (Precision + Recall) ≈ 95.2%

Why these metrics

- **Sensitivity (TPR)**: Probability of correctly detecting a sick case — the core safety metric in screening.
- **Specificity (TNR)**: Probability of correctly clearing a healthy case — reduces false alarms and downstream costs.
- **Accuracy**: Overall correctness, useful but can be misleading under class imbalance.

Enhancements (Current Pipeline):

1. **The published DMR-IR**
   - Normalisation.
   - Patient-level split.
2. **Explainability (Grad-CAM)**
   - Compute Grad-CAM heatmaps on the last conv block (e.g., layer4) to visualize high-attention regions for the "sick" class and overlay on original images.

3. **CLAHE (Contrast-Limited Adaptive Histogram Equalization)**
   - Enhances subtle, local hot-spot patterns linked to vascularity while limiting noise amplification.
   - Effect: Sensitivity improves (fewer missed positives) with minimal Specificity loss.
4. **Breast ROI Segmentation**
   - Removes torso/arm background so the network focuses on breast tissue only.
   - Effect: Fewer false positives → Specificity and Accuracy increase.
5. **Data Augmentation (train-time only)**
   - Horizontal and vertical rotations, and brightness/contrast jitter simulate posture and capture variations, MixUp & CutMix.
   - Effect: Better generalization; smoother learning curves; reduced overfitting.

6. **Training strategy & early stopping on ROC-AUC**
   - Train using AdamW (or Adam) + optional OneCycleLR. Use early stopping on validation ROC-AUC (not accuracy).

7. **Evaluation: metrics & bootstrap CIs**
   - Uncertainty: Bootstrap 95% CI for AUC (resample with replacement).
   - Provide a full, honest picture — ranking performance (AUC), precision under class imbalance (PR-AUC), and clinical operating point performance (sensitivity/spec). Bootstrap CIs quantify stability.

8. **Ablation (Additive vs LOO) + Decision rule**
   - Additive: start from baseline (CLAHE + mask + patient-aware split) and add one technique at a time to measure marginal utility.
   - LOO: start from a full set of techniques and remove one to measure necessity.
   - Additive shows marginal gains; LOO shows whether a technique is necessary.
9. **Optuna tuning block (n_trials configurable).**
10. **Auto HTML report bundling metrics + figures.**