

# Projet 3 Machine Learning

## Breast Cancer Detection

Orlane Bochart, Pauline Dumas, Simon Labracherie, Emeline Lavaux

17 Octobre 2025

# Part 1

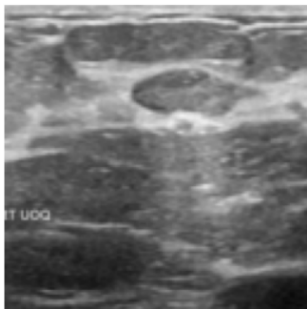
## Preprocessing

# Part 1 - Classical steps

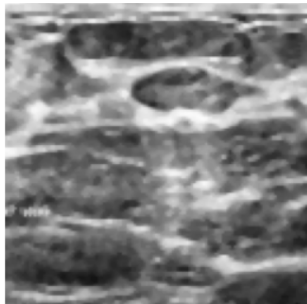
- ▶ **Scaling** : rescale pixel values to  $[0, 1]$
- ▶ **Normalization** : z-score using training set mean and std
- ▶ **Data Augmentation** : increases robustness to real-world variations
  - ▶ *Geometric* : horizontal flip, rotation  $\pm 10^\circ$ , slight zoom
  - ▶ *Photometric* : brightness ( $\times 0.8$ – $1.2$ ), contrast ( $\times 0.9$ – $1.3$ ), speckle noise

## Part 1 - Additional Tweak

- ▶ **CLAHE** (Contrast Limited Adaptive Histogram Equalization)
  - ▶ Improves local contrast
  - ▶ Outlines the mass contours
- ▶ **Median Filter**
  - ▶ Slightly reduces noise
  - ▶ Preserves tumor borders



Before (original image)



After (CLAHE + median filter)

# Part 1 - Skipped step

## ► Resizing step

- Images already **uniform** format ( $128 \times 128$ )
- Downscaling could cause loss of structural details
- Keep native resolution to **preserve diagnostic informations**

# Part 2

## Feature Engineering

## Part 2 — Method

- ▶ Extract tumor texture features from the **Gray-Level Co-Occurrence Matrix (GLCM)** (after reducing the grey levels from 255 to 32 for computational purpose) :
  - ▶ contrast, homogeneity, energy, correlation, ASM (Angular Second Moment), dissimilarity
- ▶ Capture local micro-texture using **LBP (Local Binary Patterns)** (the dominance of a particular pattern and pattern diversity)
- ▶ Measure edge density using **Canny** : proportion of “edge” pixels
- ▶ Summarize the global **intensity distribution** (after preprocessing)

## Part 2 — Features selection

- ▶ Selection of the most correlated features **Pearson correlations** with malignancy on the TRAIN set
  - ▶ We take the 4 highest in absolute value
- ▶ Features selected :
  - ▶ **contrast std and mean (glcm), dissimilarity std and mean (glcm)**

## Part 2 — Results

### ► Test performance :

- AUC : **0.784**
- Accuracy : **0.724**
- F1-score : **0.790**

### Classification Report :

Class	Precision	Recall	F1-score	Support
0	0.492	0.762	0.598	42
1	0.890	0.711	0.790	114
<b>Accuracy</b>			<b>0.724</b>	156
<b>Macro avg</b>	0.691	0.736	0.694	156
<b>Weighted avg</b>	0.783	0.724	0.739	156

## Part 2 — Method

- ▶ Add **multi-scale GLCM** features (distances =  $\{1-5\}$ ) with mean/std per distance
- ▶ Integrate **Gabor filter bank** to capture texture orientation and frequency response (4 angles  $\times$  3 frequencies)
- ▶ Include **Hu moments** (7 on Otsu mask + 7 on Canny edges) for global shape descriptors
- ▶ Add simple **feature combinations** : ratios and products between key metrics (e.g., contrast/homogeneity, edge  $\times$  contrast\_std, dissimilarity ratios)

## Part 2 — Feature selection

- ▶ Extend feature selection with **multiple methods** :
  - ▶ **ANOVA F-test** and **Mutual Information** (SelectKBest)
  - ▶ **Recursive Feature Elimination (RFE)** with Logistic Regression
- ▶ Build a **consensus selection** (union of Top-15 from correlation, ANOVA, MI, RFE).
- ▶ Create final datasets (`Xtr_sel_A`, `Xva_sel_A`, `Xte_sel_A`) based on selected features.

## Part 2 — Results

► **Test performance (threshold = 0.318) :**

- AUC : **0.823**
- Accuracy : **0.808**
- F1-score : **0.872**

### Classification Report :

Class	Precision	Recall	F1-score	Support
0	0.667	0.571	0.615	42
1	0.850	0.895	0.872	114
<b>Accuracy</b>			<b>0.808</b>	156
<b>Macro avg</b>	0.758	0.733	0.744	156
<b>Weighted avg</b>	0.801	0.808	0.803	156

# Part 3

## Modeling

## Part 3 — Classical Models

### Setup :

- ▶ Handcrafted features, standardized.
- ▶ Models : Linear SVM (calibrated) and Logistic Regression.
- ▶ Metrics on Test Set ( $n = 156$ ), balanced classes.

Model	Threshold	AUC	Precision	Recall	F1
<b>SVM (cal.)</b>	0.50	<b>0.829</b>	0.841	<b>0.930</b>	<b>0.883</b>
SVM (cal.) @F1(VAL)	0.53	0.829	0.839	0.912	0.874
LR	0.50	0.821	0.889	0.772	0.826
LR @F1(VAL)	0.41	0.821	<b>0.870</b>	0.825	0.847

### Interpretation :

- ▶ Both models achieve strong AUC ( $\sim 0.82$ – $0.83$ ), good calibration.
- ▶ SVM slightly superior overall in recall and F1.
- ▶ F1-based threshold tuning improves F1 but may slightly reduce accuracy.

## Part 3 — DenseNet121

### DenseNet121 :

- ▶ CNN with *dense connections* : each layer receives feature maps from all previous layers (feature reuse, efficient gradients).
- ▶ “121” = depth ; fewer parameters for a given accuracy vs. plain ResNets.
- ▶ Pretrained on **ImageNet** (1.2M images, 1000 classes)  $\Rightarrow$  strong generic visual features.

### How we use it :

- ▶ Grayscale inputs  $\rightarrow$  3 channels, resize  $128 \times 128$  ; light aug. (flip, small rotation, brightness, noise).
- ▶ Replace classifier head ; **freeze** features (warmup) then **fine-tune** all layers (AdamW ; separate LRs ; label smoothing + mixup ; grad clip).
- ▶ Early stopping on val AUROC ; test threshold **calibrated on validation** (FPR constraint or F1-max).

## Part 3 — DenseNet121 Results

Setup	Th	AUC	Prec.	Rec.	F1
DenseNet121 @ Max Recall ( $\text{FPR} \leq 10\%$ )	<b>0.728</b>	<b>0.908</b>	<b>0.916</b>	<b>0.956</b>	<b>0.936</b>
DenseNet121 @ F1	0.499	0.908	0.879	<b>0.956</b>	0.916

- ▶ **Best overall F1 = 0.936** at the  $\text{FPR} \leq 10\%$  operating point.
- ▶ **Recall remains high (0.956)** in both settings — suitable for screening.
- ▶ FPR-constrained threshold **improves precision** (0.916 vs. 0.879) while keeping recall.
- ▶ AUC stable at **0.908**; choose the FPR-constrained operating point for deployment.

# Part 4

## Evaluation

## Part 4 — Classical Models Evaluation

**Metrics** : Precision, Recall, F1-score, and AUC on the Test set  
(positive class = malignant case)

Model	AUC	Precision	Recall	F1
Logistic Regression @0.5	0.823	0.879	0.763	0.817
Logistic Regression @F1(VA)	0.823	0.850	0.895	0.872
SVM (calibrated) @0.5	0.829	0.841	<b>0.930</b>	<b>0.883</b>
SVM (calibrated) @F1(VA)	0.829	0.839	0.912	0.874

- ▶ **SVM (calibrated)** achieved the best trade-off — AUC=0.83, F1=0.88.
- ▶ High recall (>0.9) ensures few missed malignancies — crucial in screening.
- ▶ Logistic Regression slightly lower but stable; similar AUC  $\Rightarrow$  low model bias.

## Part 4 — DenseNet121 Evaluation

**Same pipeline**, only input resolution changes ( $128 \times 128 \rightarrow 224 \times 224$ ). Higher resolution allows finer texture and edge details in mammograms.

Model (TEST)	AUC	Precision	Recall	F1
CNN DenseNet121 ( $128 \times 128$ ) @0.5	0.913	0.891	0.930	0.910
CNN DenseNet121 ( $128 \times 128$ ) @F1(VAL)	0.913	0.904	0.912	0.908
<b>CNN DenseNet121 (<math>224 \times 224</math>) @0.5</b>	<b>0.937</b>	<b>0.894</b>	<b>0.965</b>	<b>0.928</b>
CNN DenseNet121 ( $224 \times 224$ ) @F1(VAL)	0.937	0.961	0.868	0.912

- ▶ **Higher resolution ( $224 \times 224$ )** improved AUC and F1 — better local pattern recognition.
- ▶ Recall = 0.965  $\Rightarrow$  fewer false negatives — critical for cancer detection.
- ▶ Transfer learning with ImageNet weights effectively generalizes to mammography data.

# Part 5

## Reflection

## Part 5 - Which metric should matter most ?

- ▶ **Accuracy** : Not relevant with imbalanced data (many benign, few malignant)
- ▶ **Precision** : Important, but not the priority False positives can be verified later through further tests
- ▶ **Recall : Matter the most.** Missing a malignant case (false negative) can delay treatment and harm the patient

## Part 5 — Consequences of each error type and context consideration

### ▶ **Error impact :**

- ▶ *False positive* : Inconvenience for the patient ; cost for the hospital
- ▶ *False negative* : Missed diagnosis ; serious harm to the patient

### ▶ **Context consideration :**

- ▶ If used for *automatic detection*, recall becomes critical
- ▶ If used as a *decision aid*, a balance between precision and recall may be acceptable

# Conclusion

- ▶ **AI models in cancer detection** can help identify suspicious cases faster
- ▶ **Human expertise** remains essential so models should be used as **decision-support tools**, not as replacements for radiologists