COVID-19 Detection from Chest X-Ray Images

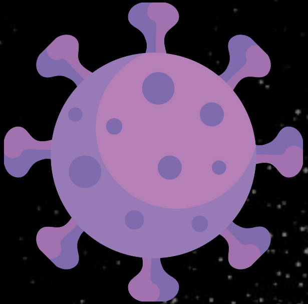
Classical ML vs Deep Learning & Transfer Learning

Diana Kalstein

Jonas Heyden

Mirco Kappel

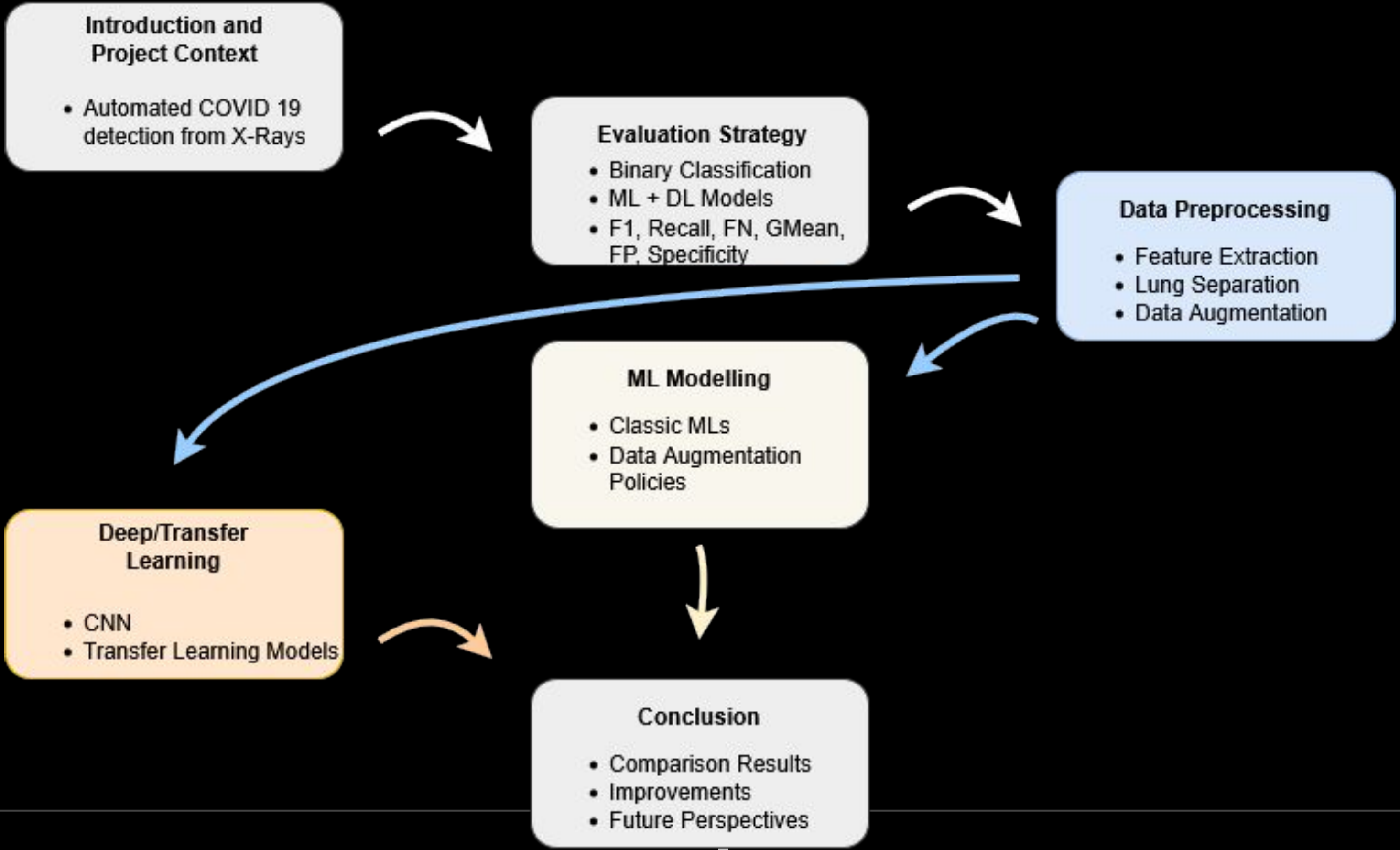
Moaoia Shatila



Data Science Bootcamp - Nov 25, 2025

Liora

Analysis of Covid-19 chest x-rays



Introduction and Project Context

- Automated COVID 19 detection from X-Rays

Evaluation Strategy

- Binary Classification
- ML + DL Models
- F1, Recall, FN, GMean, FP, Specificity

Data Preprocessing

- Feature Extraction
- Lung Separation
- Data Augmentation

ML Modelling

- Classic MLs
- Data Augmentation Policies

Deep/Transfer Learning

- CNN
- Transfer Learning Models

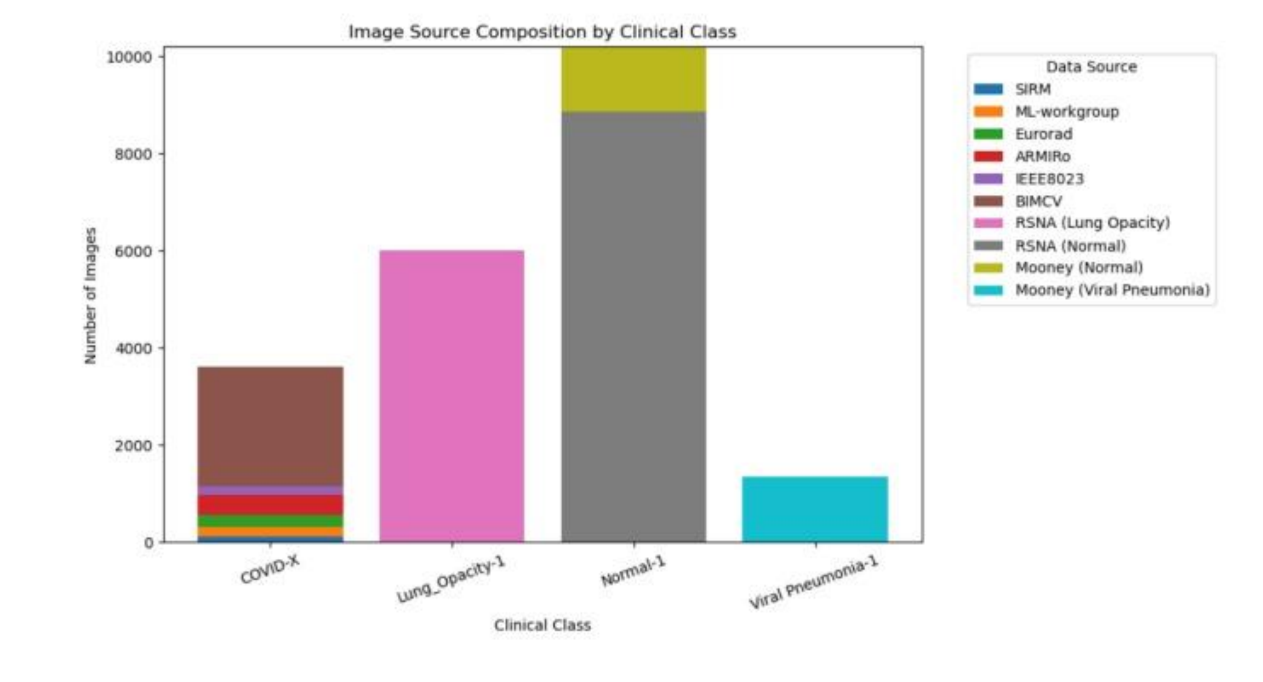
Conclusion

- Comparison Results
- Improvements
- Future Perspectives

Introduction

Dataset

- Chest X-rays + lung masks (Region of Interest)
- Strong Class Imbalance
- Multiple Data Sources



Introduction

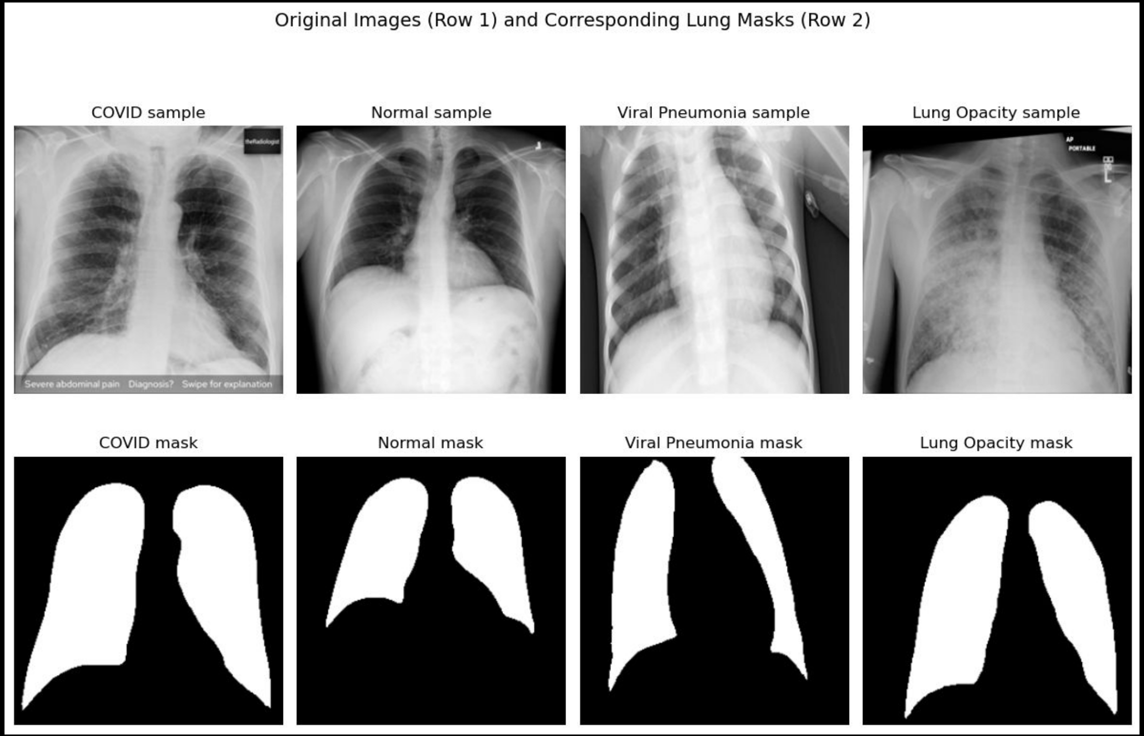
Chest X-Ray images

- 4 different classes
- Chest X-Ray + Lung mask (Kaggle)

Objective

Automated COVID19 Detection from XRays

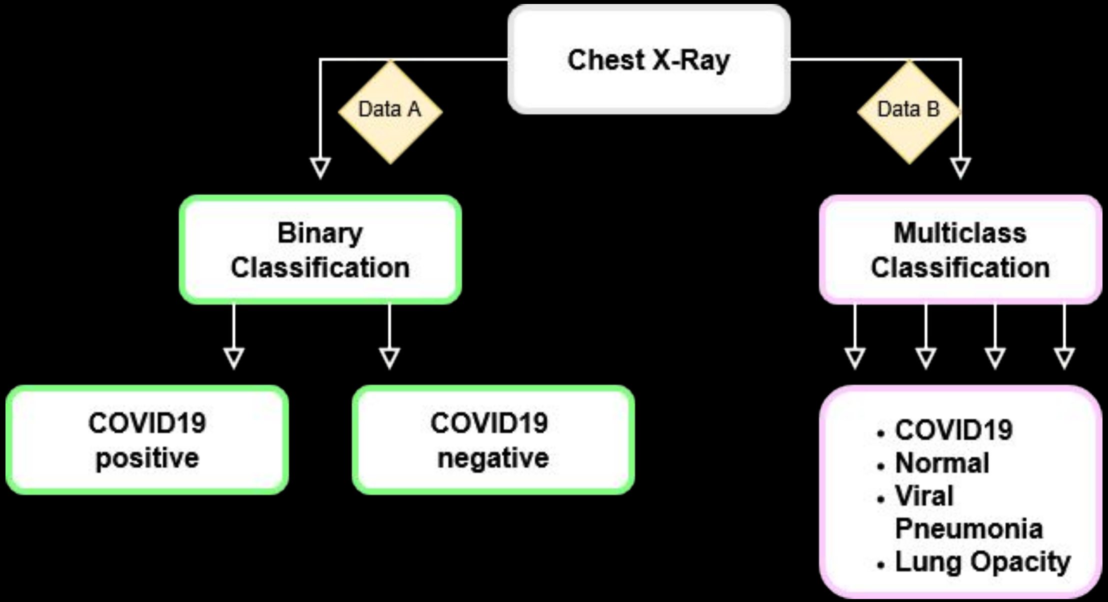
(...but why not rt-PCR?!)



Problem Definition

Classification Problem: COVID vs Non-COVID

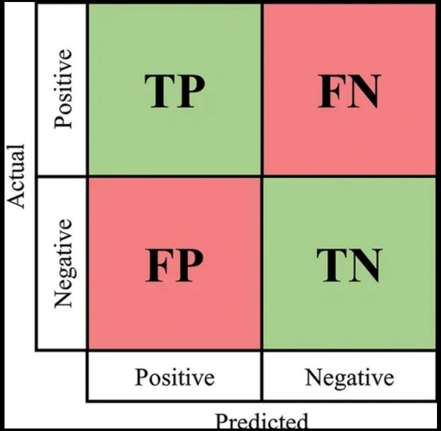
- Clinical Aspect: Most Urgent Question → (COVID yes/no?)
- Simplicity and Efficiency
- Starting Point for more Complex Problems (Multiclass Classification)



Task Framing

Most important metrics

- F1-score, Recall, False Negatives (FN), G-Mean, FP
- Secondary: Precision, Specificity
- Accuracy could be misleading (imbalanced Data)



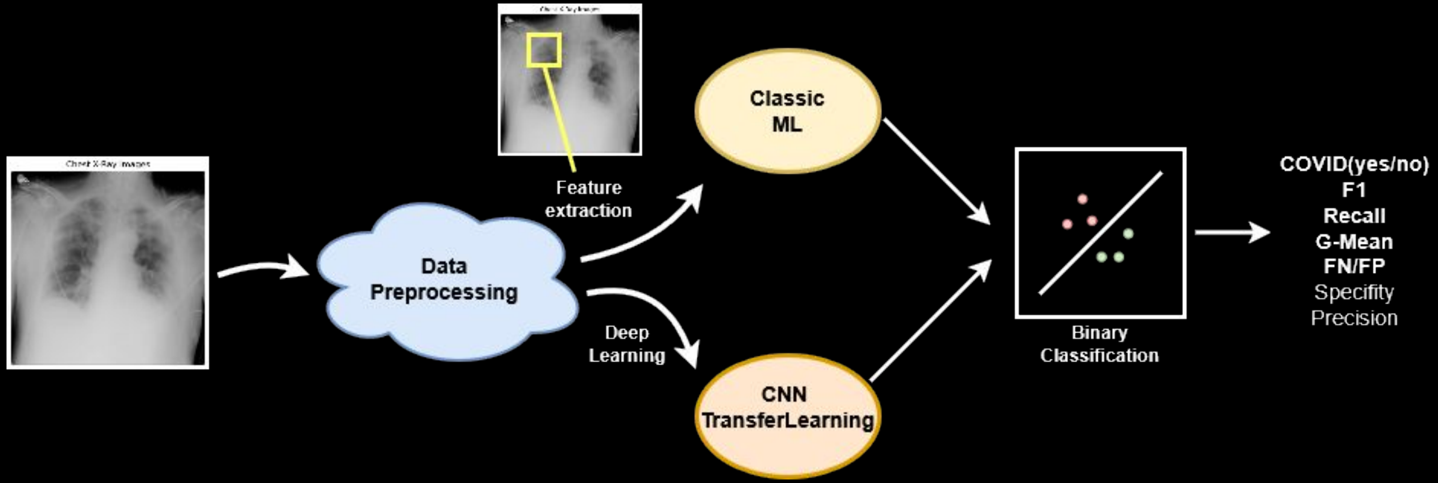
		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Measurement	Formula
F1 Score	2 x (Precision x Recall)/(Precision + Recall)
Recall	TP/(TP+FN)
G-Mean	(Recall * Specificity)^1/2
Precision	TP/(TP+FP)
Specificity	TN/(TN+FP)

Evaluation Strategy

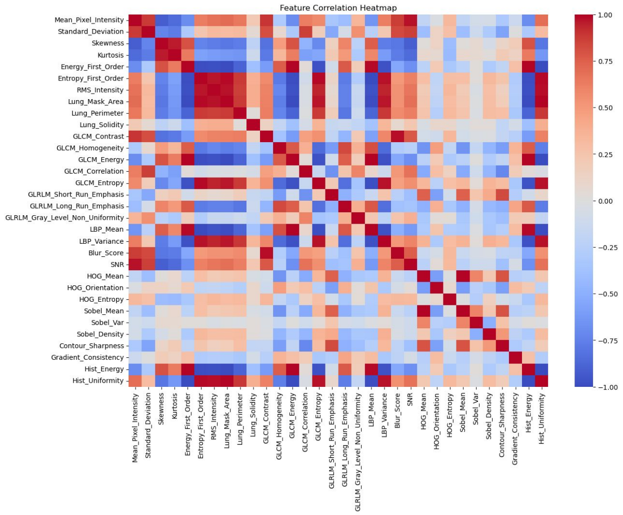
Binary Classification:

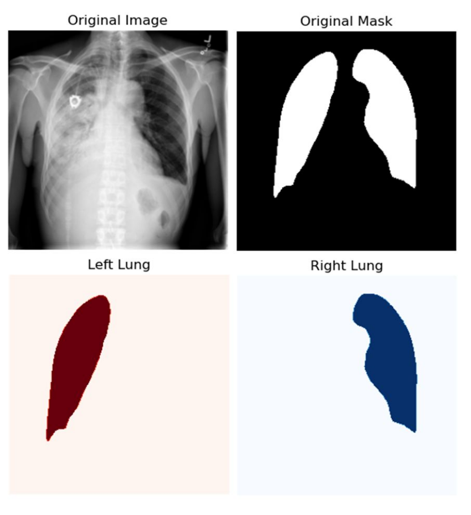
- Data Preprocessing
- Classic MLs
- Deep Learning

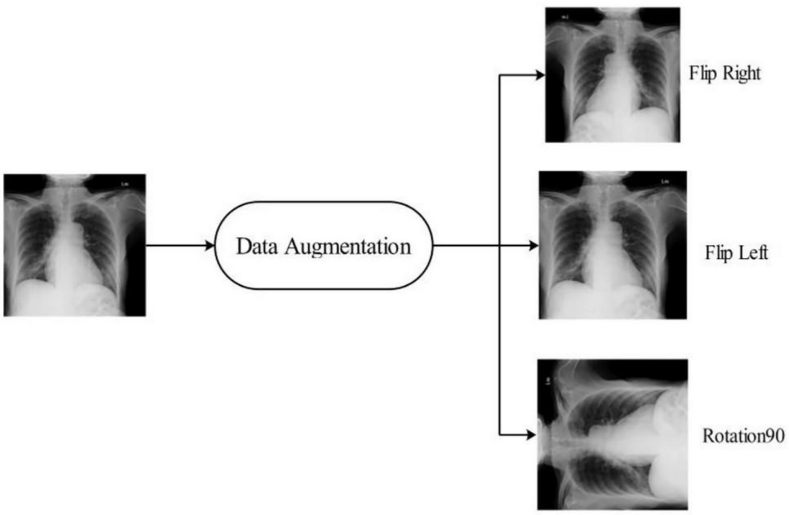


Data Preprocessing

- X-ray chest images into quantitative descriptors (Textures detect. GLCM GLRLM etc, For details see appendix)
- Lung Separation → left/ right lung features extracted separately
- Data Augmentation → Different Policies







Classical Machine Learning Baselines

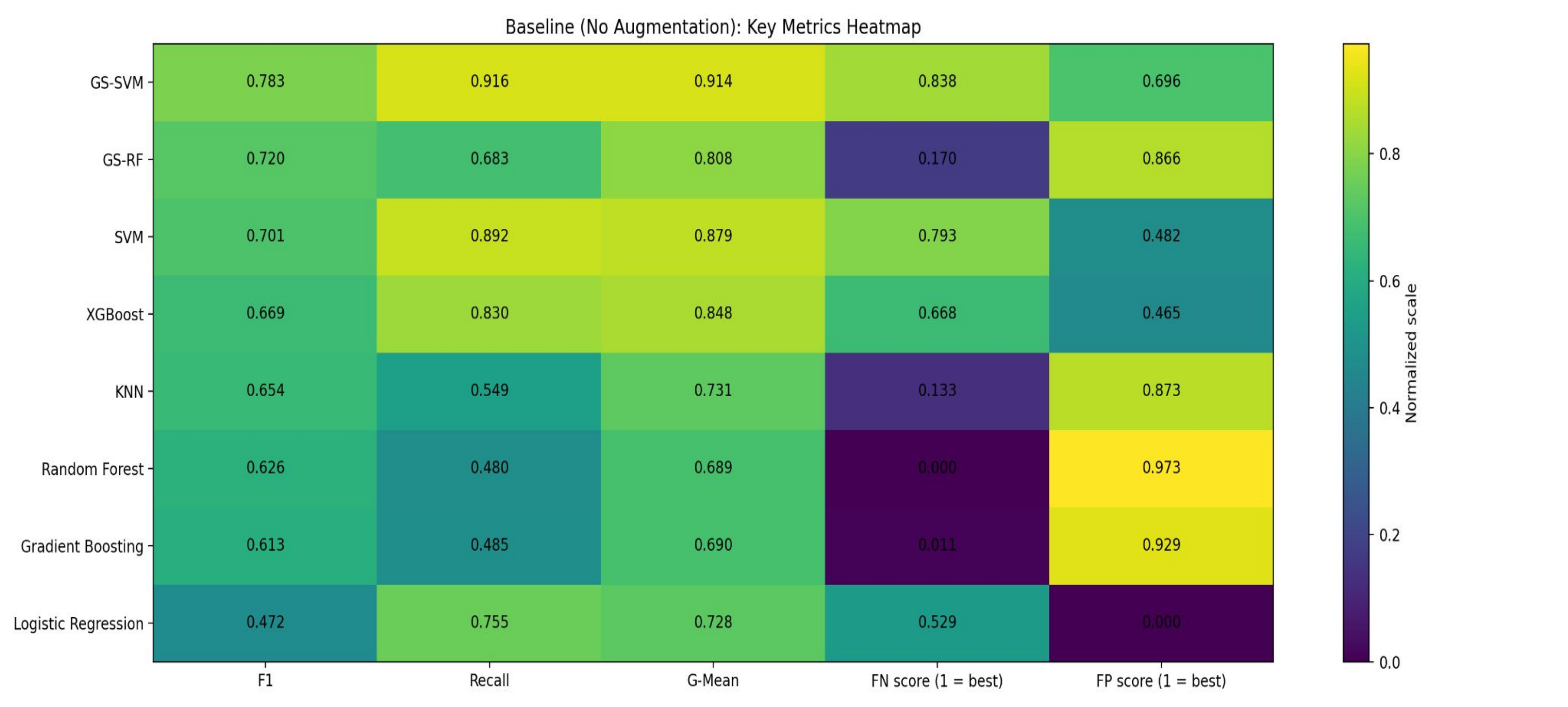
Handcrafted radiomic features from CXR + lung masks (32)

Models evaluated:

- Logistic Regression, SVM, KNN (instance-based models, work directly on the feature space geometry)
- Random Forest, Gradient Boosting, XGBoost (tree-based ensemble models, they learn decision rules)
- GS-SVM, GS-RF (optimized versions, hyperparameter tuning)

Baseline on imbalanced data (COVID ≈ 17%)

Key issue: high false negatives for most models



Baseline (No Augmentation): Key Metrics Heatmap

	F1	Recall	G-Mean	FN score (1 = best)	FP score (1 = best)
GS-SVM	0.783	0.916	0.914	0.838	0.696
GS-RF	0.720	0.683	0.808	0.170	0.866
SVM	0.701	0.892	0.879	0.793	0.482
XGBoost	0.669	0.830	0.848	0.668	0.465
KNN	0.654	0.549	0.731	0.133	0.873
Random Forest	0.626	0.480	0.689	0.000	0.973
Gradient Boosting	0.613	0.485	0.690	0.011	0.929
Logistic Regression	0.472	0.755	0.728	0.529	0.000

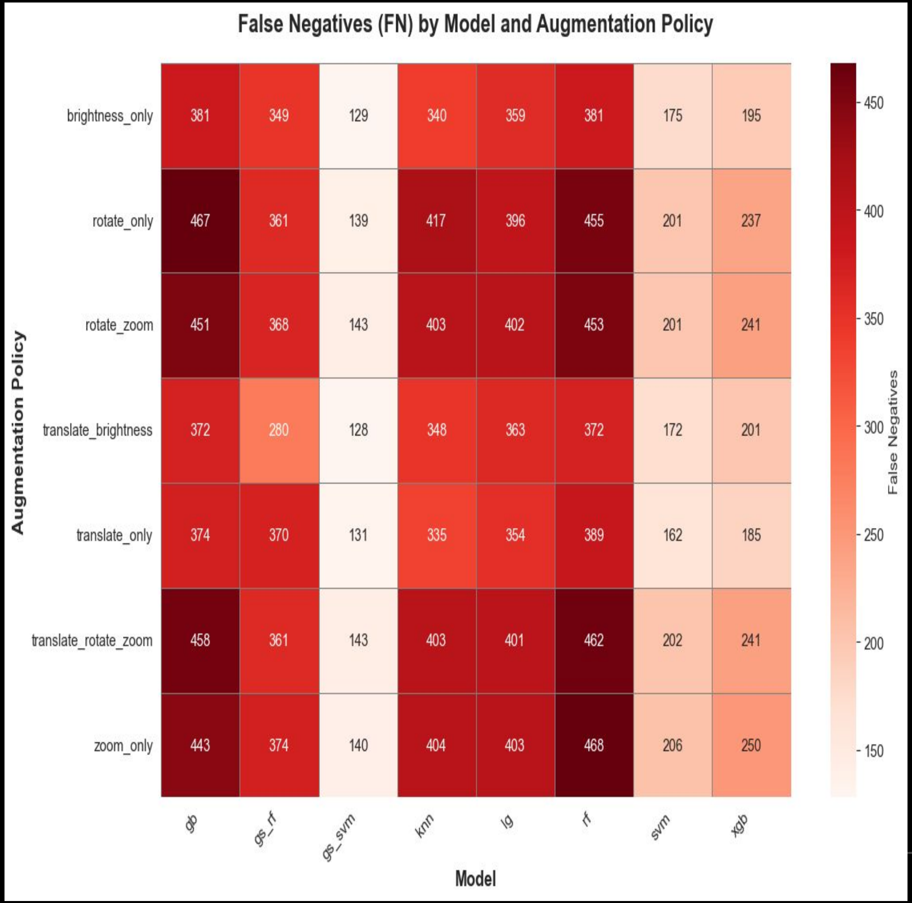
Why & How We Used Augmentation

Motivation for Augmentation (Classical ML):

- COVID class underrepresented
- Baseline models showed high FN count
- **Goal:**
  - Improve minority class representation
  - Reduce FN
  - Increase robustness
- Offline augmentation on COVID only:
  - Brightness, translation, rotation, zoom, combinations

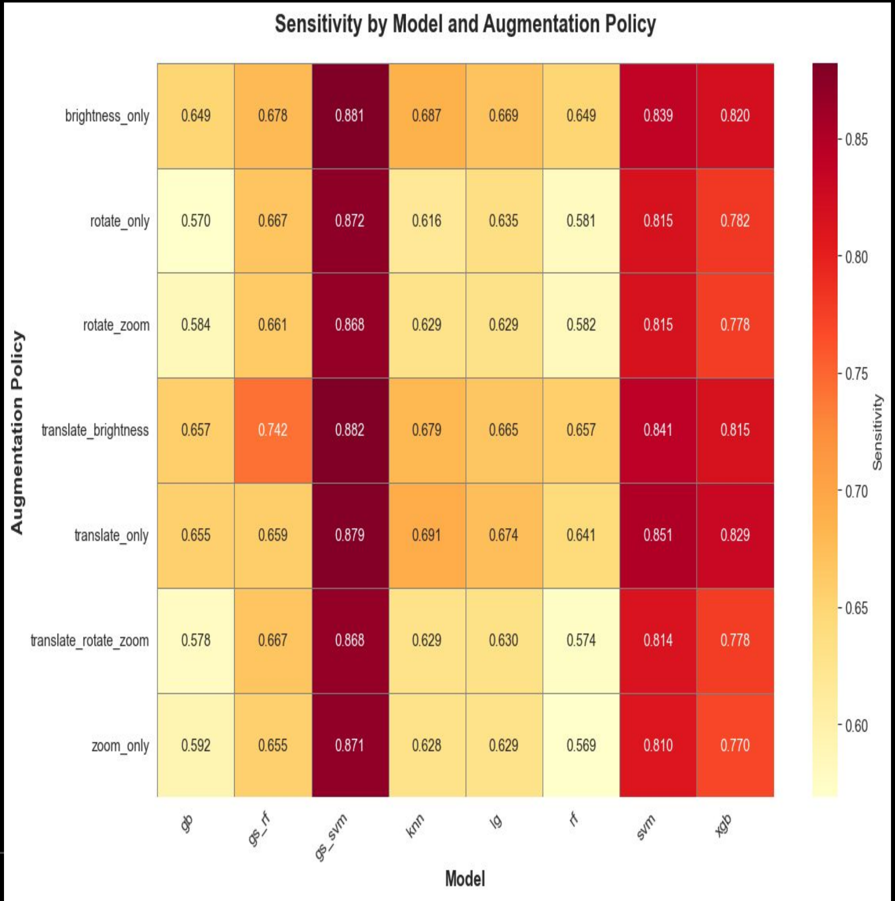
False Negatives (FN) - Number of positive cases incorrectly classified as negative (missed diagnoses).

Sensitivity (Recall) - Proportion of actual positives correctly identified: TP / (TP + FN). Measures ability to detect positive cases.



False Negatives (FN) by Model and Augmentation Policy

Augmentation Policy	gb	gs_rf	gs_svm	knn	lg	rf	svm	xgb
brightness_only	381	349	129	340	359	381	175	195
rotate_only	467	361	139	417	396	455	201	237
rotate_zoom	451	368	143	403	402	453	201	241
translate_brightness	372	280	128	348	363	372	172	201
translate_only	374	370	131	335	354	389	162	185
translate_rotate_zoom	458	361	143	403	401	462	202	241
zoom_only	443	374	140	404	403	468	206	250

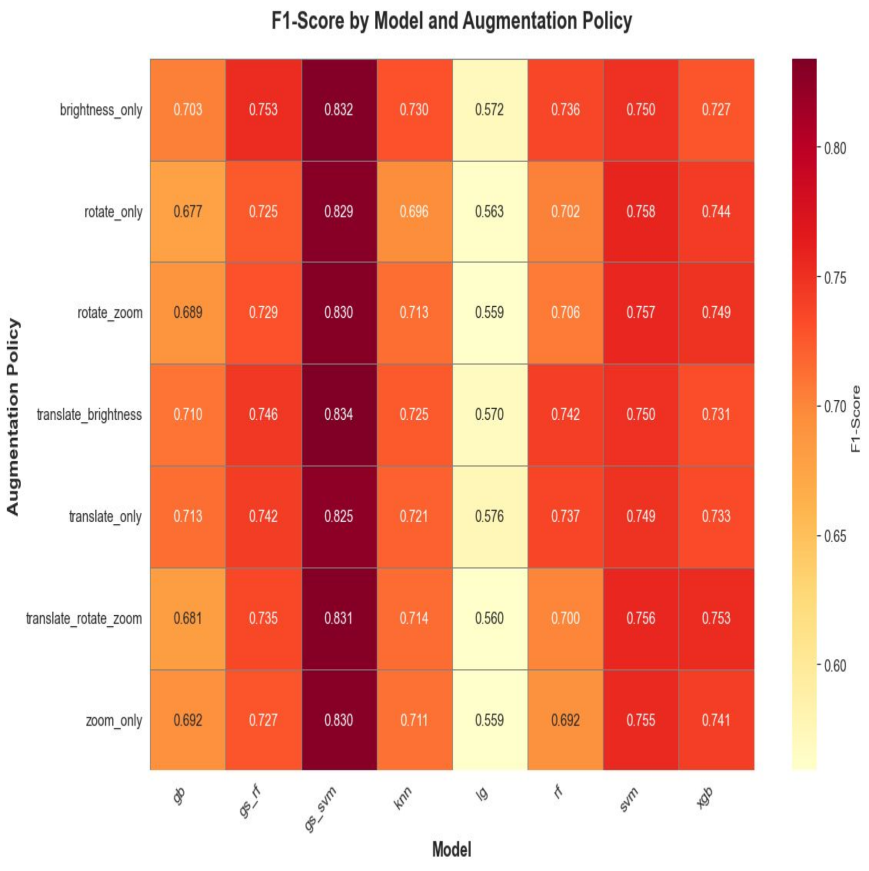


Sensitivity by Model and Augmentation Policy

Augmentation Policy	gb	gs_rf	gs_svm	knn	lg	rf	svm	xgb
brightness_only	0.649	0.678	0.881	0.687	0.669	0.649	0.839	0.820
rotate_only	0.570	0.667	0.872	0.616	0.635	0.581	0.815	0.782
rotate_zoom	0.584	0.661	0.868	0.629	0.629	0.582	0.815	0.778
translate_brightness	0.657	0.742	0.882	0.679	0.665	0.657	0.841	0.815
translate_only	0.655	0.659	0.879	0.691	0.674	0.641	0.851	0.829
translate_rotate_zoom	0.578	0.667	0.868	0.629	0.630	0.574	0.814	0.778
zoom_only	0.592	0.655	0.871	0.628	0.629	0.569	0.810	0.770

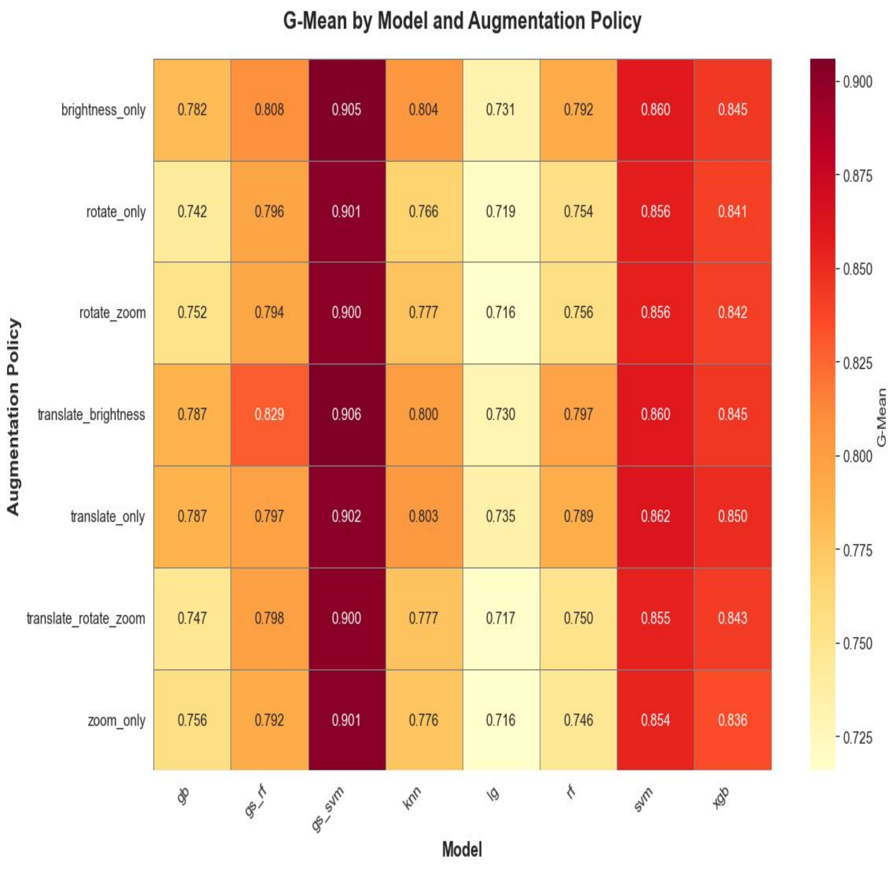
F1-Score - Harmonic mean of precision and recall: 2 × (Precision × Recall) / (Precision + Recall).

Overall classification performance

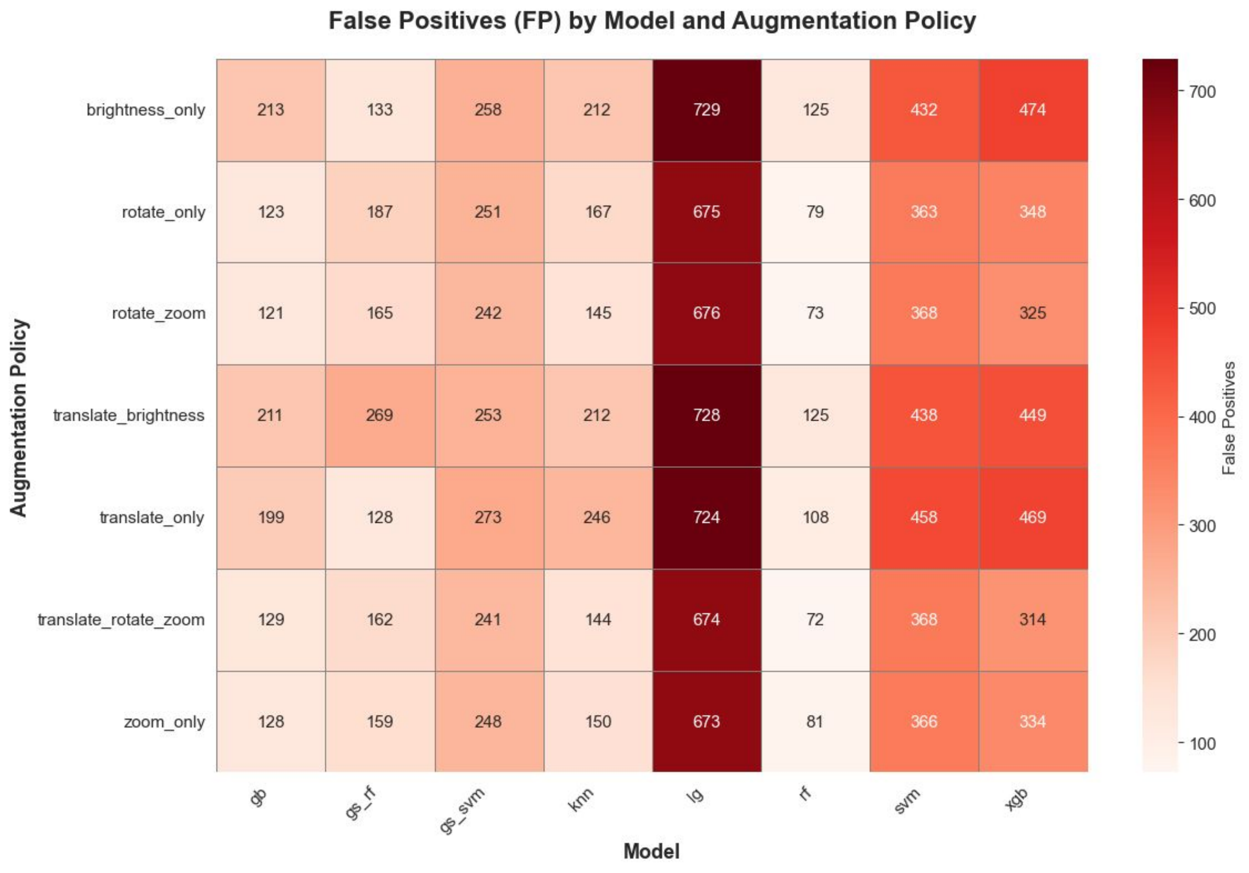


G-Mean - Geometric mean of sensitivity and specificity: √ (Sensitivity × Specificity).

Provides balanced performance measures.



False Positives (FP) - Number of negative cases (healthy patients) incorrectly flagged as positive.

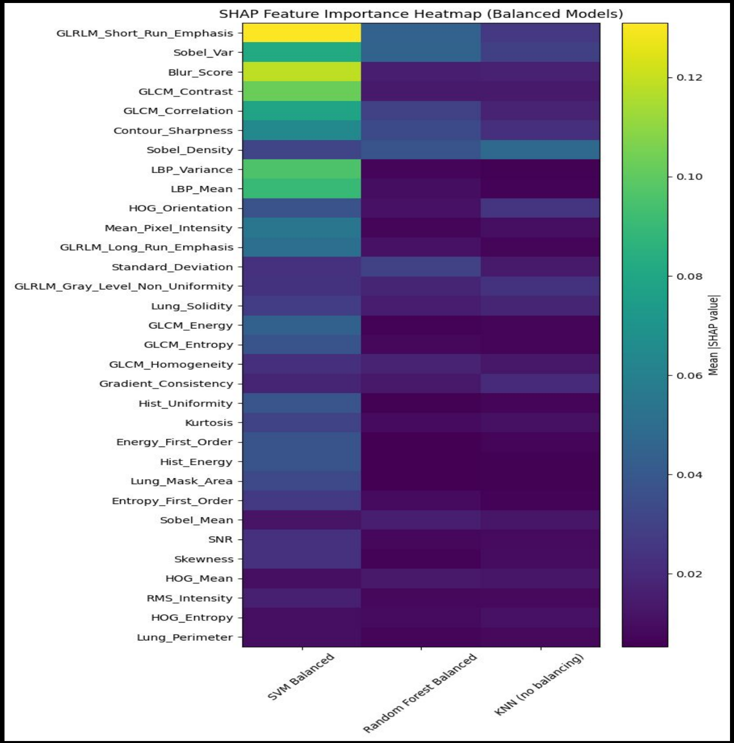


	gb	gs_rf	gs_svm	knn	lg	rf	svm	xgb
brightness_only	213	133	258	212	729	125	432	474
rotate_only	123	187	251	167	675	79	363	348
rotate_zoom	121	165	242	145	676	73	368	325
translate_brightness	211	269	253	212	728	125	438	449
translate_only	199	128	273	246	724	108	458	469
translate_rotate_zoom	129	162	241	144	674	72	368	314
zoom_only	128	159	248	150	673	81	366	334

Results & Key Takeaways (Classical ML)

Results & Limitations:

- GS-SVM best classical model across all policies
- Best config: translate + brightness
- FN reduced but still clinically high
- Performance ceiling ≈ 0.83–0.84 F1
- Tree models: high specificity, poor sensitivity
- Bonus: SHAP Enhancements
- Motivation to move to deep learning

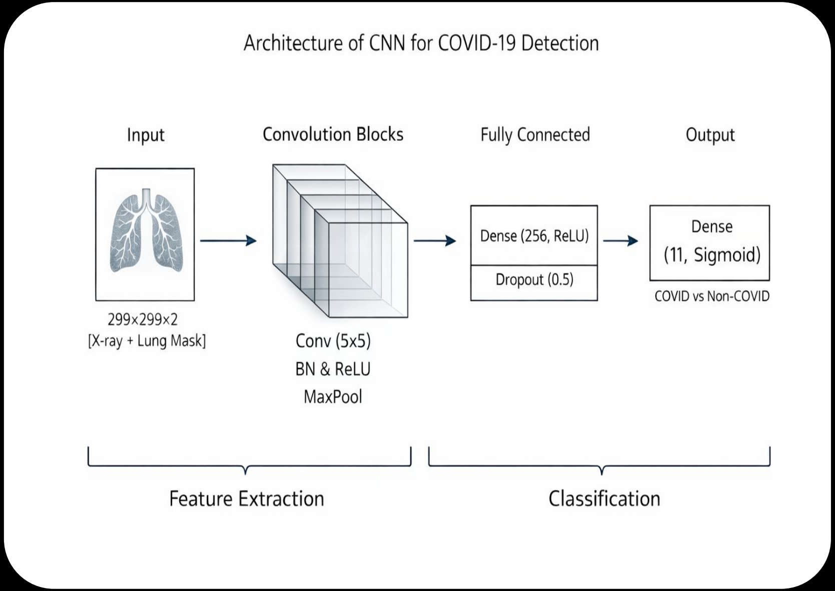


Deep Learning Phase

Data Handling

- Batch-level class balancing (32 samples / batch), 375 batches per epoch
- COVID-targeted augmentation only
  (3 augmented, 7 original)
- Train/ Val split before augmentation (no data leakage)

CNN From Scratch

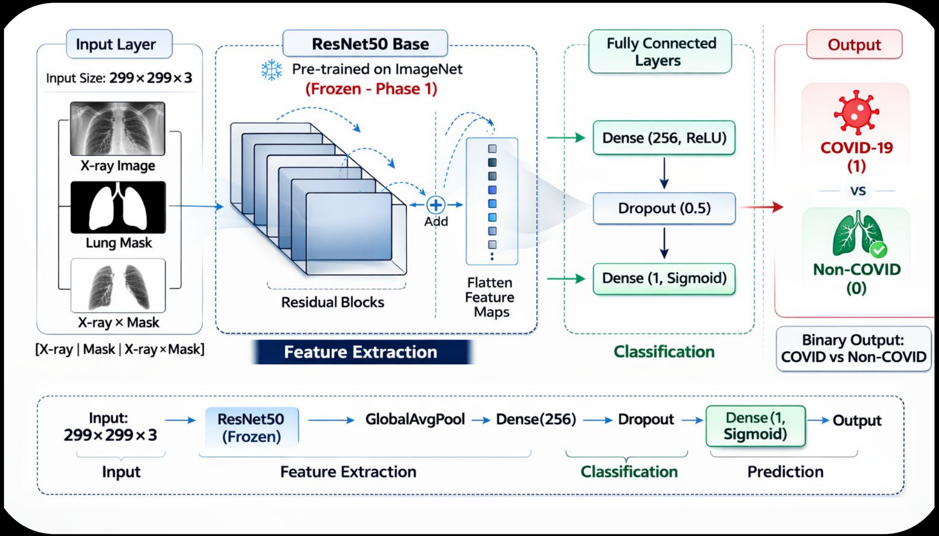


Training Strategy

- Loss: Binary Cross-Entropy | Optimizer: Adam
- Single-phase training (no frozen backbone)
- EarlyStopping on val_loss (patience = 10)
- ReduceLROnPlateau: factor 0.5, patience 3, min_lr 1e-6
- ModelCheckpoint: saves best val_loss model
- Batch size: 32 | Class-balanced batches
- COVID-targeted augmentation (3 aug : 7 original)

Transfer Learning Models

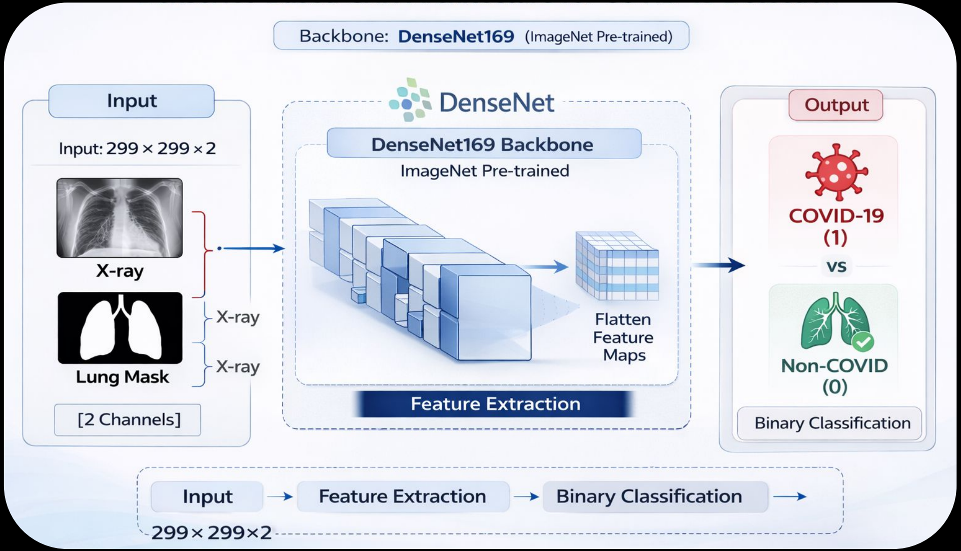
ResNet50



Training Strategy

- Two-phase transfer learning strategy:
  - Phase 1 — Frozen base | LR = 1e-3 | 10 epochs
  - Train Dense head only; ResNet weights frozen
  - EarlyStopping patience 5, Reduce LR patience 3
  - Phase 2 — Fine-tuning | LR = 5e-5 | 30 epochs
  - Unfreeze all layers except first 100
  - EarlyStopping patience 8, min_lr = 1e-7
- Loss: Binary Cross-Entropy | Optimizer: Adam
- ModelCheckpoint + CSVLogger + TensorBoard

DenseNet169



Training Strategy

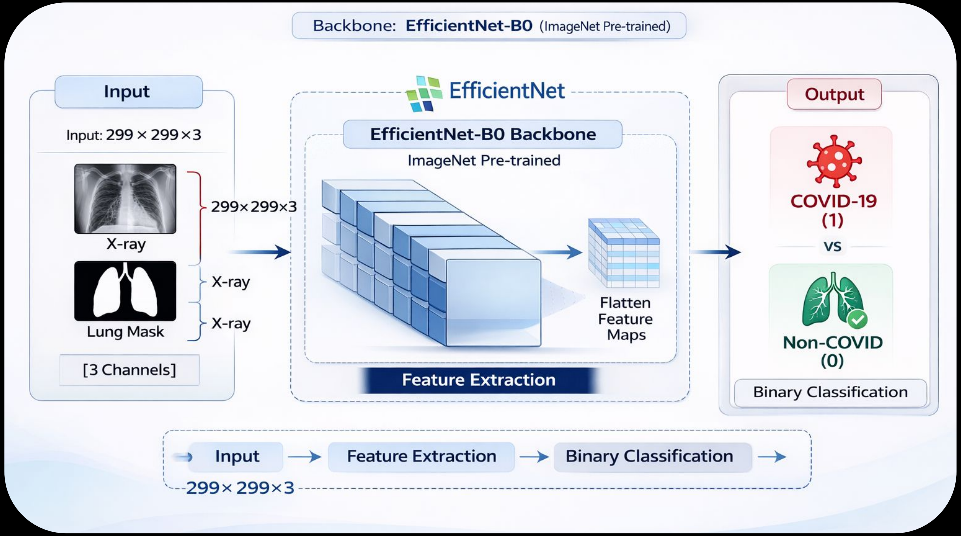
Phase 1 - Frozen backbone

- LR = 1e-3, 10 epochs
- COVID: 7 orig + 3 aug per batch
- ReduceLROnPlateau patience 3, Early stopping patience 8

Phase 2 - Fine tuning

- LR = 1e-5, 20 epochs, Unfreeze last 70 layers
- Loss: BCEWithLogitsLoss (pos_weight = 2.0)
- Augmentation: ShiftScaleRotate
- Threshold: 0.5, No Focal Loss

EfficientNet-B0



Training Strategy

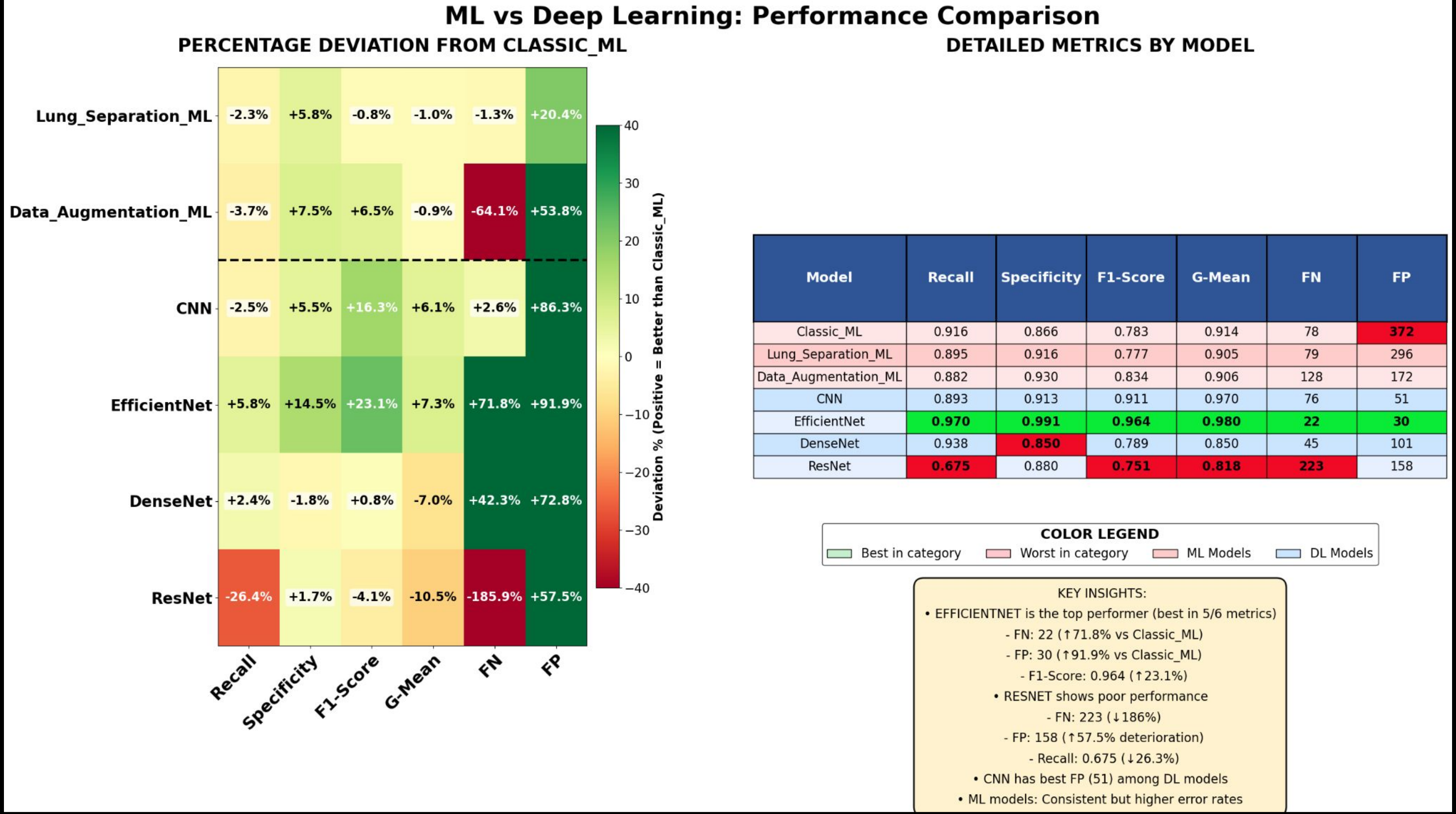
- Two-phase transfer learning
  - Phase 1: frozen backbone, train classifier head
  - Phase 2: fine-tune last 2 blocks
- Loss: BCEWithLogitsLoss ; Optimizer: Adam ; Threshold: 0.5

EfficientNet-B0

Final Clinical Performance

- Recall ≈ 0.97
- FP = 30
- F1 ≈ 0.96
- G-Mean ≈ 0.98
- FN = 22 (very low)

Conclusion



ML vs Deep Learning: Performance Comparison

PERCENTAGE DEVIATION FROM CLASSIC_ML

	Recall	Specificity	F1-Score	G-Mean	FN	FP
Lung_Separation_ML	-2.3%	+5.8%	-0.8%	-1.0%	-1.3%	+20.4%
Data_Augmentation_ML	-3.7%	+7.5%	+6.5%	-0.9%	-64.1%	+53.8%
CNN	-2.5%	+5.5%	+16.3%	+6.1%	+2.6%	+86.3%
EfficientNet	+5.8%	+14.5%	+23.1%	+7.3%	+71.8%	+91.9%
DenseNet	+2.4%	-1.8%	+0.8%	-7.0%	+42.3%	+72.8%
ResNet	-26.4%	+1.7%	-4.1%	-10.5%	-185.9%	+57.5%

DETAILED METRICS BY MODEL

Model	Recall	Specificity	F1-Score	G-Mean	FN	FP
Classic_ML	0.916	0.866	0.783	0.914	78	**372**
Lung_Separation_ML	0.895	0.916	0.777	0.905	79	296
Data_Augmentation_ML	0.882	0.930	0.834	0.906	128	172
CNN	0.893	0.913	0.911	0.970	76	51
EfficientNet	**0.970**	**0.991**	**0.964**	**0.980**	**22**	**30**
DenseNet	0.938	**0.850**	0.789	0.850	45	101
ResNet	**0.675**	0.880	**0.751**	**0.818**	**223**	158

COLOR LEGEND

🟩 Best in category 🟥 Worst in category 🟥 ML Models 🟦 DL Models

KEY INSIGHTS:

- EFFICIENTNET is the top performer (best in 5/6 metrics)
  - FN: 22 (↑71.8% vs Classic_ML)
  - FP: 30 (↑91.9% vs Classic_ML)
  - F1-Score: 0.964 (↑23.1%)
- RESNET shows poor performance
  - FN: 223 (↓186%)
  - FP: 158 (↑57.5% deterioration)
  - Recall: 0.675 (↓26.3%)
- CNN has best FP (51) among DL models
- ML models: Consistent but higher error rates

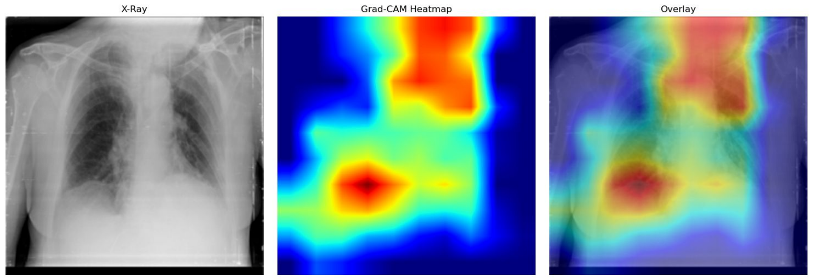
Interpretability (Grad-CAM)

Grad-CAM Heatmaps

- Models shall focus on lung regions

Exemple: Grad-CAM running on EfficientNet:

- Live demo in Streamlit follows immediately



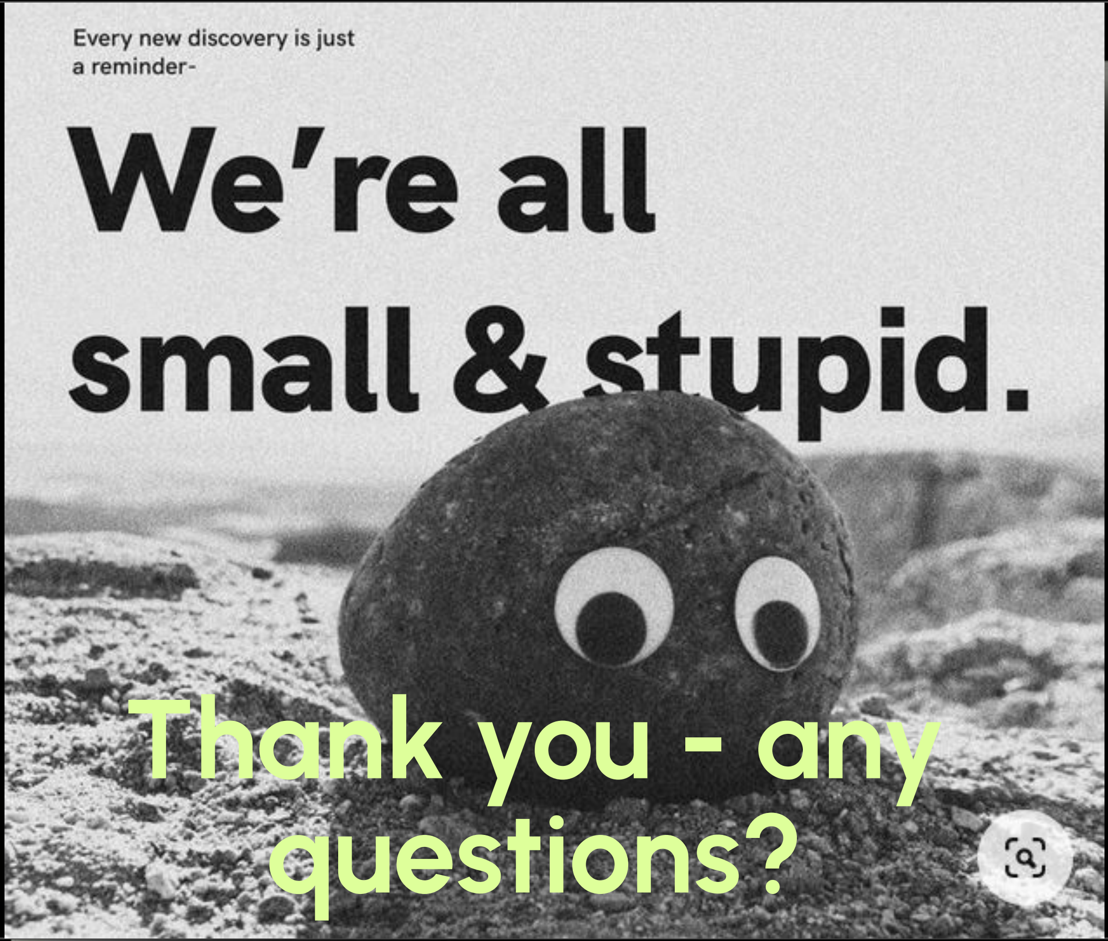
Project Conclusion

Limitations:

- **Imbalanced Dataset, multiple sources** → risk of bias, over/underfitting, and artifacts
- **Time constraints** limited hyperparameter tuning and model exploration
- But Good performance achieved with multiple efficient ML & more advanced CNN Models
- **Interpretability** SHAP only partially explored

Future work could be:

- **External validation** to assess generalizability of the best model
- **Extended classification:** multiclass or two-staged. Not only binary Covid
- **Support** for masked and unmasked images or Chest X-Ray image detection
- **Advanced methods:** Vision Transformers, improved hyperparameter optimization, SHAP-guided refinement
- **Industrialization**
- **etc.**



Thank you - any questions?

Archive, ML Feature Extraction

Base

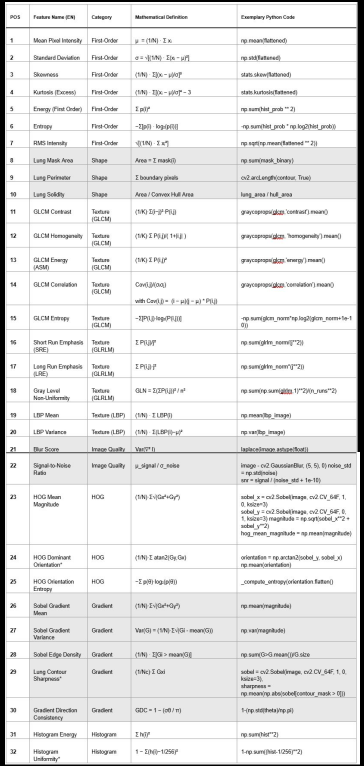
Pos	feat
1	Mean_Pixel_Intensity
2	Standard_Deviation
3	Skewness
4	Kurtosis
5	Energy_First_Order
6	Entropy_First_Order
7	RMS_Intensity
8	Lung_Mask_Area
9	Lung_Perimeter
10	Lung_Solidity
11	GLCM_Contrast
12	GLCM_Homogeneity
13	GLCM_Energy
14	GLCM_Correlation
15	GLCM_Entropy
16	GLRLM_Short_Run_Emphasis
17	GLRLM_Long_Run_Emphasis
18	GLRLM_Gray_Level_Non_Uniformity
19	LBP_Mean
20	LBP_Variance
21	Blur_Score
22	SNR
23	HOG_Mean
24	HOG_Orientation
25	HOG_Entropy
26	Sobel_Mean
27	Sobel_Var
28	Sobel_Density
29	Contour_Sharpness
30	Gradient_Consistency
31	Hist_Energy
32	Hist_Uniformity

Bilateral (extended from base, 62 features max)

Pos - feat
1 Mean_Pixel_Intensity
2 Standard_Deviation
3 Skewness
4 Kurtosis
5 Energy_First_Order
6 Entropy_First_Order
7 RMS_Intensity
8 Lung_Mask_Area_Total
9 Lung_Perimeter_Total
21 Blur_Score
22 SNR
8 Lung_Mask_Area_Left
8 Lung_Mask_Area_Right
9 Lung_Perimeter_Left
9 Lung_Perimeter_Right
10 Lung_Solidity_Left
10 Lung_Solidity_Right
11 GLCM_Contrast_Left
11 GLCM_Contrast_Right
12 GLCM_Homogeneity_Left
12 GLCM_Homogeneity_Right
13 GLCM_Energy_Left
13 GLCM_Energy_Right
14 GLCM_Correlation_Left
14 GLCM_Correlation_Right
15 GLCM_Entropy_Left
15 GLCM_Entropy_Right
16 GLRLM_Short_Run_Emphasis_Left
16 GLRLM_Short_Run_Emphasis_Right
17 GLRLM_Long_Run_Emphasis_Left
17 GLRLM_Long_Run_Emphasis_Right
18 GLRLM_Gray_Level_Non_Uniformity_Left
18 GLRLM_Gray_Level_Non_Uniformity_Right
19 LBP_Mean_Left
19 LBP_Mean_Right
20 LBP_Variance_Left
20 LBP_Variance_Right
23 HOG_Mean_Left
23 HOG_Mean_Right
24 HOG_Dominant_Orientation_Left
24 HOG_Dominant_Orientation_Right
25 HOG_Entropy_Left
25 HOG_Entropy_Right
26 Sobel_Mean_Left
26 Sobel_Mean_Right
27 Sobel_Var_Left
27 Sobel_Var_Right
28 Sobel_Density_Left
28 Sobel_Density_Right
29 Contour_Sharpness_Left
29 Contour_Sharpness_Right
30 Gradient_Consistency_Left
30 Gradient_Consistency_Right
31 Hist_Energy_Left
31 Hist_Energy_Right
32 Hist_Uniformity_Left
32 Hist_Uniformity_Right

14 GLCM_Corr_LR_Ratio
27 Sobel_Var_LR_Ratio
28 Sobel_Dens_LR_Ratio
29 Contour_Sharp_LR_Ratio
8 Area_LR_Ratio

Used calculations to extract features / image pixel data



POS	Feature Name (EN)	Category	Mathematical Definition	Exemplary Python Code
1	Mean Pixel Intensity	First-Order	$\mu = (1/N) \cdot \Sigma x_i$	np.mean(flattened)
2	Standard Deviation	First-Order	$\sigma = \sqrt{[(1/N) \cdot \Sigma(x_i - \mu)^2]}$	np.std(flattened)
3	Skewness	First-Order	$(1/N) \cdot \Sigma[(x_i - \mu)/\sigma]^3$	stats.skew(flattened)
4	Kurtosis (Excess)	First-Order	$(1/N) \cdot \Sigma[(x_i - \mu)/\sigma]^4 - 3$	stats.kurtosis(flattened)
5	Energy (First Order)	First-Order	$\Sigma p(i)^2$	np.sum(hist_prob ** 2)
6	Entropy	First-Order	$-\Sigma[p(i) \cdot \log_2(p(i))]$	-np.sum(hist_prob * np.log2(hist_prob))
7	RMS Intensity	First-Order	$\sqrt{[(1/N) \cdot \Sigma x_i^2]}$	np.sqrt(np.mean(flattened ** 2))
8	Lung Mask Area	Shape	Area = Σ mask(i)	np.sum(mask_binary)
9	Lung Perimeter	Shape	Σ boundary pixels	cv2.arcLength(contour, True)
10	Lung Solidity	Shape	Area / Convex Hull Area	lung_area / hull_area
11	GLCM Contrast	Texture (GLCM)	$(1/K) \cdot \Sigma(i-j)^2 \cdot P(i,j)$	graycoprops(glcm,'contrast').mean()
12	GLCM Homogeneity	Texture (GLCM)	$(1/K) \cdot \Sigma P(i,j)/(1+\|i,j\|)$	graycoprops(glcm, 'homogeneity').mean()
13	GLCM Energy (ASM)	Texture (GLCM)	$(1/K) \cdot \Sigma P(i,j)^2$	graycoprops(glcm,'energy').mean()
14	GLCM Correlation	Texture (GLCM)	$\mathrm{Cov}(i,j)/(\sigma_i \sigma_j)$ with $\mathrm{Cov}(i,j) = (i - \mu_i)(j - \mu_j) * P(i,j)$	graycoprops(glcm,'correlation').mean()
15	GLCM Entropy	Texture (GLCM)	$-\Sigma[P(i,j) \cdot \log_2(P(i,j))]$	-np.sum(glcm_norm*np.log2(glcm_norm+1e-10))
16	Short Run Emphasis (SRE)	Texture (GLRLM)	$\Sigma P(i,j)/j^2$	np.sum(glrlm_norm/(j**2))
17	Long Run Emphasis (LRE)	Texture (GLRLM)	$\Sigma P(i,j) \cdot j^2$	np.sum(glrlm_norm*(j**2))
18	Gray Level Non-Uniformity	Texture (GLRLM)	GLN = $\Sigma(\Sigma P(i,j))^2 / n^2$	np.sum(np.sum(glrlm,1)**2)/(n_runs**2)
19	LBP Mean	Texture (LBP)	$(1/N) \cdot \Sigma$ LBP(i)	np.mean(lbp_image)
20	LBP Variance	Texture (LBP)	$(1/N) \cdot \Sigma(\mathrm{LBP}(i)-\mu)^2$	np.var(lbp_image)
21	Blur Score	Image Quality	$\mathrm{Var}(\nabla^2 I)$	laplace(image.astype(float))
22	Signal-to-Noise Ratio	Image Quality	μ_signal / σ_noise	image - cv2.GaussianBlur, (5, 5), 0) noise_std = np.std(noise) snr = signal / (noise_std + 1e-10)

23	HOG Mean Magnitude	HOG	$(1/N) \cdot \Sigma\sqrt{(Gx^2+Gy^2)}$	sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
				sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3) magnitude = np.sqrt(sobel_x**2 + sobel_y**2)
				hog_mean_magnitude = np.mean(magnitude)

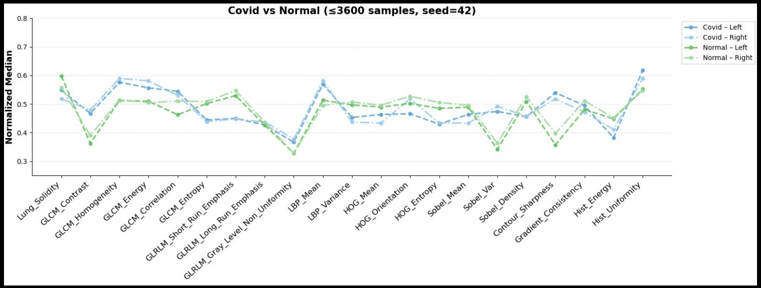
24	HOG Dominant Orientation*	HOG	$(1/N) \cdot \Sigma$ atan2(Gy,Gx)	orientation = np.arctan2(sobel_y, sobel_x)
				np.mean(orientation)

25	HOG Orientation Entropy	HOG	$-\Sigma p(\theta) \cdot \log_2(p(\theta))$	_compute_entropy(orientation.flatten()
26	Sobel Gradient Mean	Gradient	$(1/N) \cdot \Sigma\sqrt{(Gx^2+Gy^2)}$	np.mean(magnitude)
27	Sobel Gradient Variance	Gradient	$\mathrm{Var}(G) = (1/N) \cdot \Sigma\sqrt{(Gi - \mathrm{mean}(G))}$	np.var(magnitude)
28	Sobel Edge Density	Gradient	$(1/N) \cdot \Sigma[Gi > \mathrm{mean}(G)]$	np.sum(G>G.mean())/G.size

29	Lung Contour Sharpness*	Gradient	$(1/Nc) \cdot \Sigma$ Gxi	sobel = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3),
				sharpness = np.mean(np.abs(sobel[contour_mask > 0]))

30	Gradient Direction Consistency	Gradient	GDC = $1 - (\sigma\theta / \pi)$	1-(np.std(theta)/np.pi)
31	Histogram Energy	Histogram	$\Sigma h(i)^2$	np.sum(hist**2)
32	Histogram Uniformity*	Histogram	$1 - \Sigma(h(i)-1/256)^2$	1-np.sum((hist-1/256)**2)

Lung-Split Visualisation, Median Trends on Lung Split Data Before ML Training

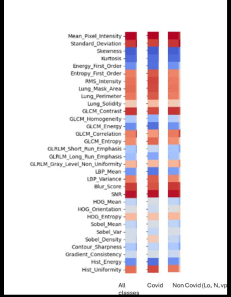


- White - ‘Global’ feats, calculations over both lungs, stayed in bilateral the same
Yellow - Extension of ‘Global’ feats were splitted, calculation on each lung separately
Turquoise - Feats, “divisions of lungs R/L” from existing yellow

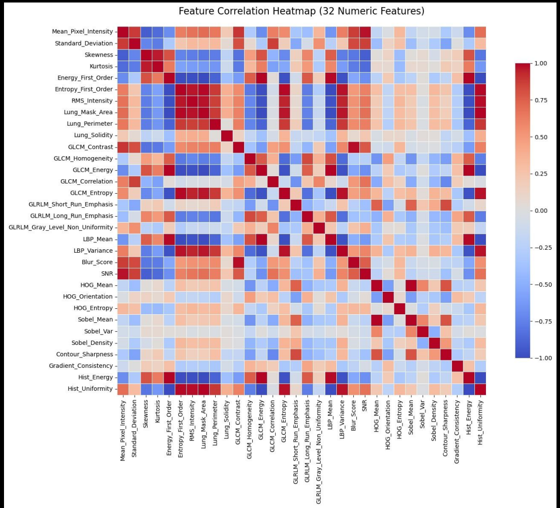
Archive

Feature Engineering for Radiomis

Converts X-ray chest images into **quantitative descriptors,** extracted **within the lungs** using the segmentation mask



- **First-Order** - global intensity statistics (1-7)
- **Shape** - anatomical geometry and contours, edges (8-10)
- **Texture (GLCM)** - local spatial intensity structure (11-15)
- **Texture (GLRLM)** - run-length texture patterns (16-18)
- **Texture (LBP)** - robust micro-textures (19-20)
- **Gradient / HOG** - edge strength and orientation (23-30)
- **Image Quality** - sharpness and noise(21-22)
- **Histogram** - global intensity distribution(31-32)



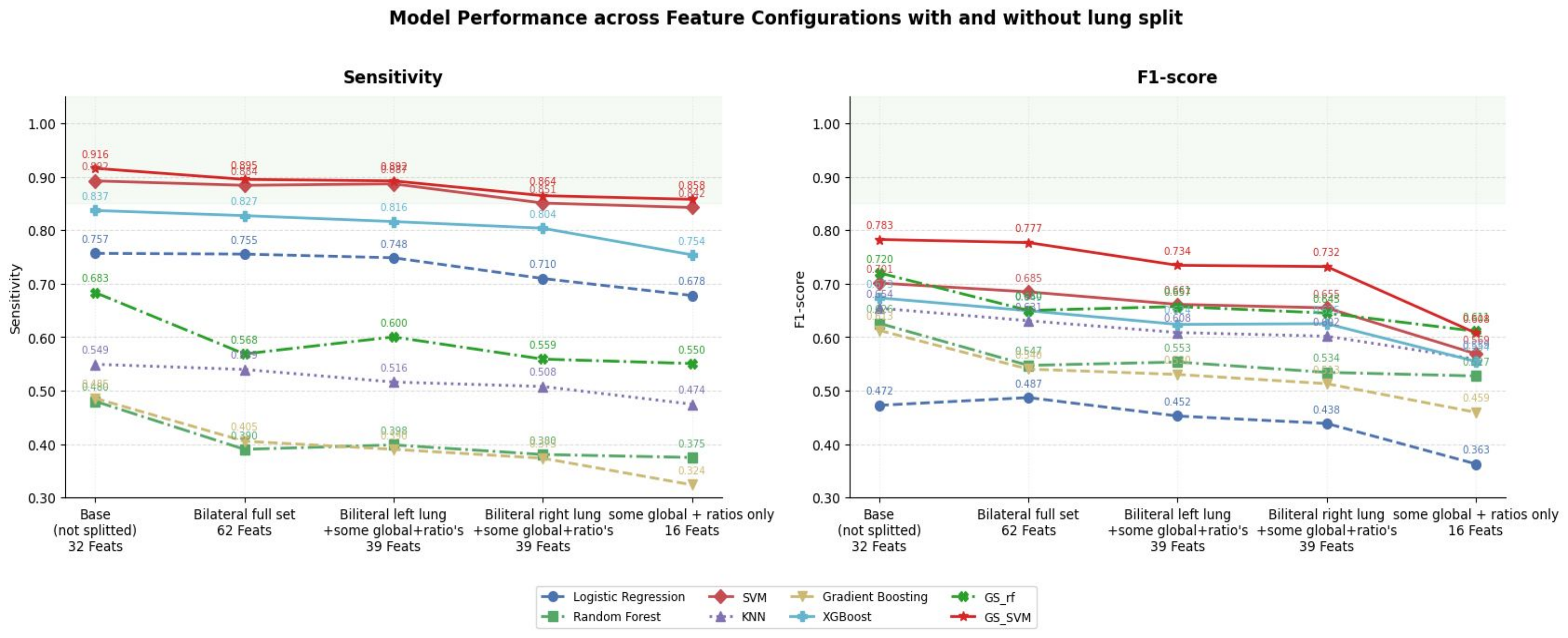
form a **compact, interpretable, ML representation** that quantifies subtle deviations from normal

Quick first validation:

- **Potentially multicollinearity analysis with Correlation Heatmap**

- **GLCM** Gray Level Co occurrence Matrix, **GLRLM** Gray Level Run Length Matrix, **LBP** Local Binary Patterns

Archive, ML Lung-Split Performance Comparison

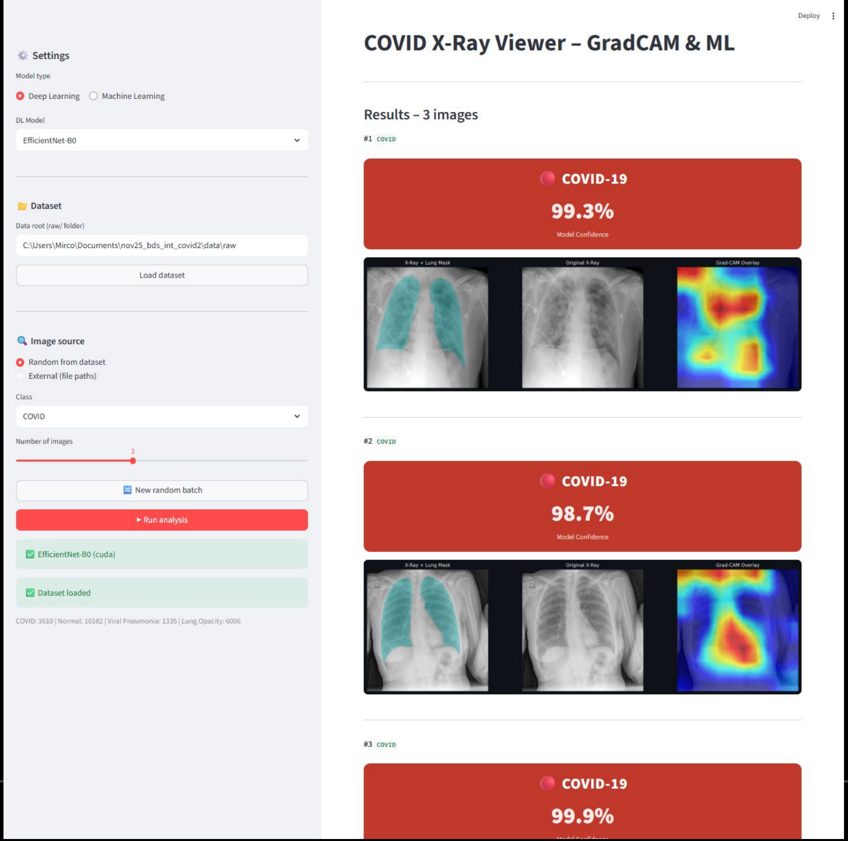


Model Performance across Feature Configurations with and without lung split

Sensitivity

F1-score

Backup SL, DL & ML



⚙ Settings

Model type

● Deep Learning ○ Machine Learning

DL Model

EfficientNet-B0

📁 Dataset

Data root (raw/ folder)

C:\Users\Mirco\Documents\nov25_bds_int_covid2\data\raw

🔍 Image source

● Random from dataset
○ External (file paths)

Class

COVID

Number of images

3

☑ EfficientNet-B0 (cuda)

☑ Dataset loaded

COVID: 3610 | Normal: 10182 | Viral Pneumonia: 1335 | Lung Opacity: 6006

COVID X-Ray Viewer – GradCAM & ML

Results – 3 images

#1 COVID

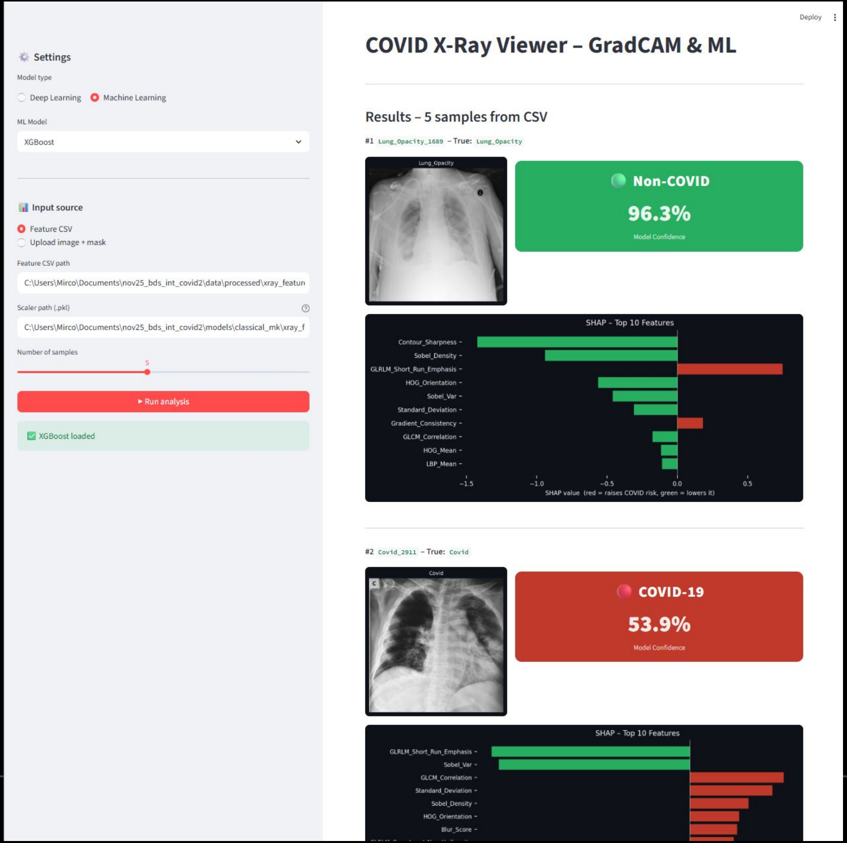
COVID-19
99.3%
Model Confidence

#2 COVID

COVID-19
98.7%
Model Confidence

#3 COVID

COVID-19
99.9%



⚙ Settings

Model type

○ Deep Learning ● Machine Learning

ML Model

XGBoost

📊 Input source

● Feature CSV
○ Upload image + mask

Feature CSV path

C:\Users\Mirco\Documents\nov25_bds_int_covid2\data\processed\xray_featur

Scaler path (.pkl)

C:\Users\Mirco\Documents\nov25_bds_int_covid2\models\classical_mk\xray_f

Number of samples

5

☑ XGBoost loaded

COVID X-Ray Viewer – GradCAM & ML

Results – 5 samples from CSV

#1 Lung_Opacity_1689 – True: Lung_Opacity

Non-COVID
96.3%
Model Confidence

#2 Covid_2911 – True: Covid

COVID-19
53.9%
Model Confidence