AI-Based Medical Diagnostics: Comparative Evaluation of Machine Learning, Deep Learning, and Foundation Models for Chest X-Ray Interpretation

Author: Hilal Abdessamad

Date: October 30, 2025

Abstract

Chest X-rays remain the most widely used imaging modality for diagnosing pulmonary conditions, yet radiologist workload and reporting delays create significant bottlenecks in clinical workflows. This capstone project investigates AI-driven approaches for automated detection of pleural effusion in chest radiographs, comparing three distinct paradigms: classical machine learning with radiomics, deep learning vision models, and foundation models with parameter-efficient adaptation.

Using 549 training and 122 test images from the MIMIC-CXR dataset, I implemented and evaluated five models: Logistic Regression, Random Forest, and K-Nearest Neighbors trained on 93 radiomic features; a fine-tuned EfficientNet-B0 CNN; and MedCLIP vision transformer adapted with LoRA. The Random Forest model achieved the highest performance (AUC 0.80, AUPRC 0.80), outperforming both the deep learning model (AUC 0.69) and the foundation model which showed improved performance (AUC 0.72, AUPRC 0.75) compared to initial trials. However, fairness analysis revealed an18.5% sensitivity gap for postero-anterior views in the deep learning model, highlighting deployment risks. Despite attempting loss re-weighting mitigation strategies, the fairness gap persisted and, in some configurations, worsened, demonstrating the complexity of algorithmic bias. Grad-CAM visualizations confirmed the model focuses on clinically relevant lung regions, while operational metrics showed both the classical ML approach (sub-4ms CPU latency) and the optimized foundation model (~10ms GPU latency, $0.002 per 1k images) offer viable paths for deployment.

This work demonstrates that while modern deep learning architectures offer interpretability advantages, traditional ML with carefully engineered features can provide superior performance and efficiency for well-defined clinical tasks. The findings emphasize the importance of fairness analysis, the challenges of bias mitigation, and operational constraints in developing clinically viable AI systems. This project reveals that high performance metrics alone are insufficient. A model achieving 0.92 AUROC can still be clinically dangerous and inequitable if fairness gaps remain unresolved.

1. Background & Clinical Need

Chest X-rays are the cornerstone of diagnostic imaging worldwide, serving as the first-line investigation for numerous cardiopulmonary conditions. Their accessibility, low cost, and minimal radiation exposure make them indispensable in both emergency departments and routine clinical care. Among the many findings radiologists look for, pleural effusion, which is the abnormal accumulation of fluid in the pleural space, is a diagnostic target. It can indicate heart failure, pneumonia, malignancy, or other serious conditions requiring prompt intervention.

Despite their ubiquity, the current workflow for CXR interpretation faces several challenges. Radiologists are increasingly burdened by growing imaging volumes, leading to potential delays in report turnaround times. In high-acuity settings like emergency departments, these delays can impact patient care decisions. Additionally, inter-observer variability exists even among experienced radiologists, particularly for subtle findings. Studies have shown that radiologists can disagree on the presence of findings like pleural effusion in up to 20% of cases, especially when the effusion is small or when image quality is suboptimal.

The pain point this project addresses is the need for a reliable, automated screening tool that could augment rather than replace the radiologist's workflow. An effective AI system could triage incoming studies, flagging potentially abnormal cases for priority review, or serve as a "second reader" to reduce missed findings. In resource-limited settings where radiologists are scarce, such systems could enable rapid screening by non-specialist clinicians. However, for any AI tool to be clinically viable, it must not only perform accurately but also operate fairly across different patient populations and imaging protocols, remain interpretable to clinicians, and integrate seamlessly into existing workflows.

This capstone investigates the feasibility of building such a system by systematically comparing three AI paradigms. Rather than assuming that newer, larger models are necessarily better, I wanted to understand the real-world trade-offs: Does a foundation model with billions of parameters actually outperform a simpler approach? What are the fairness implications when models are trained on data from specific imaging protocols? And critically, what would it take to deploy these models in practice?

---

2. Data & Cohorts

Dataset Source

All data for this project came from the MIMIC-CXR (Medical Information Mart for Intensive Care - Chest X-Ray) database, a publicly available dataset containing 377,110 chest X-rays from 65,379 patients at Beth Israel Deaconess Medical Center. The dataset includes DICOM images alongside structured labels derived from radiology reports using natural language processing. For this study, I focused specifically on the binary classification task of detecting pleural effusion, identified as Target Index 9 in the MIMIC-CXR label hierarchy.

I used a smaller set from the MIMIC-CXR dataset, found on https://uni-bonn.sciebo.de/s/YHuwFOg6q6sw1ZX/download --output-document.

Cohort Selection and Preprocessing

After applying inclusion criteria (frontal views only, complete metadata, valid DICOM files) and exclusion criteria (lateral views, corrupted images, missing labels), the final cohort consisted of 671 images. These were split into a training cohort of 549 images and a held-out test cohort of 122 images, maintaining approximately the same class distribution in both sets.

The preprocessing pipeline included several standard steps:

- **DICOM parsing** to extract pixel data and metadata including view position (AP vs PA)
- **Pixel normalization** to standardize intensity values across different scanners
- **Data augmentation** during training with horizontal flips and small rotations (±10 degrees) to improve model generalization

A parallel radiomics pipeline was developed for the classical ML models. This involved extracting 93 radiomic features from each image, including first-order statistics, shape descriptors, and texture features. Due to computational constraints and quality control requirements, this extraction was successfully completed for a subset of 57 images from the training set. While this represents a significant reduction in available training data for the ML models, I chose to proceed because radiomics-based approaches are known to work well even with smaller sample sizes when features are carefully engineered.

Fairness-Sensitive Attributes

A main component of this project was assessing model fairness across different subgroups. I selected ViewCodeSequence_CodeMeaning (distinguishing antero-posterior from postero-anterior views) as the primary sensitive attribute for several reasons. First, AP and PA views represent different clinical contexts: AP views are typically obtained with portable equipment for patients who cannot stand (often sicker, hospitalized patients), while PA views are standard outpatient imaging with better image quality. Second, preliminary exploration suggested potential performance disparities between these groups. The distribution showed 86 AP views and 36 PA views in the test set, providing sufficient representation for meaningful fairness analysis.

Cohort Characteristics

Characteristic	Training Set	Test Set
Total Images	549	122
Pleural Effusion Positive	303 (55.2%)	68 (55.7%)
Pleural Effusion Negative	246 (44.8%)	54 (44.3%)
Antero-Posterior Views	387 (70.5%)	86 (70.5%)
Postero-Anterior Views	162 (29.5%)	36 (29.5%)
Radiomics Subset	57 images	N/A

The relatively balanced class distribution (approximately 55% positive cases) meant that while I still needed to address class imbalance, it wasn't as severe as often encountered in medical imaging tasks. The consistent distribution across train and test sets was intentional to ensure valid performance estimates.

---

3. Methods

3.1 Classical Machine Learning Pipeline

For the classical ML approach, I trained three models on the extracted radiomic features: Logistic Regression (as a linear baseline), Random Forest (for handling non-linear relationships and feature interactions), and K-Nearest Neighbors (to capture local similarity patterns). Given the small dataset of 57 samples with radiomic features, I used 5-fold stratified cross-validation to get robust performance estimates while maximizing the use of available data.

The preprocessing pipeline within each CV fold consisted of:

1. **SimpleImputer** with median strategy to handle any missing radiomic values
2. **StandardScaler** to normalize features to zero mean and unit variance
3. **SMOTE (Synthetic Minority Over-sampling Technique)** to address class imbalance by generating synthetic examples of the minority class

I want to note here that using SMOTE within cross-validation folds rather than on the entire dataset before splitting was deliberate. This prevents data leakage where synthetic samples in the training set might be too similar to real samples in the validation fold. It's a mistake I made in early experiments, and catching it taught me the importance of maintaining strict train-test separation even when using augmentation techniques.

3.2 Deep Learning Vision Model

The deep learning approach used EfficientNet-B0, a convolutional neural network pretrained on ImageNet. I chose this architecture because it offers a great balance between performance and computational efficiency. It achieves state-of-the-art results with significantly fewer parameters than older architectures like ResNet or VGG. In some experimental configurations, I also explored DenseNet-121 as an alternative CNN architecture, which showed promising results in certain scenarios.

The training protocol involved:

- **Input preprocessing:** Images resized to 224×224 pixels (EfficientNet-B0's expected input size)
- **Data augmentation:** RandomHorizontalFlip and RandomRotation(±10°) applied during training
- **Fine-tuning strategy:** All layers unfrozen and trained end-to-end (not just the classification head)
- **Optimizer:** Adam with learning rate 1e-4
- **Loss function:** BCEWithLogitsLoss (binary cross-entropy with logits for numerical stability)
- **Training duration:** 5 epochs with early stopping monitoring

I trained for only 5 epochs because I noticed the model began to overfit on the relatively small dataset after that point. In retrospect, I probably should have implemented proper early stopping based on validation loss, but a fixed epoch count worked well too.

3.3 Foundation Model with LoRA Adaptation

For the foundation model approach, I used MedCLIP, a vision-language model specifically pretrained on medical images and radiology reports. The base model is a Vision Transformer (ViT) with approximately 86 million parameters. Since fine-tuning such a large model would be computationally prohibitive and prone to overfitting on our small dataset, I applied Low-Rank Adaptation (LoRA).

LoRA works by freezing the pretrained weights and injecting trainable low-rank decomposition matrices into specific layers. This reduces the number of trainable parameters while maintaining the model's ability to adapt to new tasks. My implementation evolved through several iterations:

Initial configuration (failed attempt):

- Applied LoRA with rank r=16 to query and value projection layers
- Added a simple classification head (768 → 256 → 1)
- Total trainable parameters: 1,573,889
- Result: AUC 0.56 (barely better than random)

Optimized configuration (successful):

- Applied LoRA with rank r=16 and alpha α=16 (matching rank and alpha values)
- Targeted q_proj and v_proj layers in the attention mechanism
- Refined training hyperparameters and learning rate schedule
- Total trainable parameters: 786,945 (only 0.68% of full model parameters)
- Result: AUC 0.72, AUPRC 0.75

The difference between configurations, reducing trainable parameters from 1.57M to 0.79M while improving performance from 0.56 to 0.72 AUC, illustrates how important proper hyperparameter tuning is for parameter-efficient fine-tuning methods. The idea was that MedCLIP's pretraining on medical images would give it a head start compared to EfficientNet's general-purpose ImageNet pretraining, and with proper configuration, this hypothesis was partially validated.

3.4 Fairness Analysis and Mitigation

Fairness was assessed by disaggregating all performance metrics across the AP vs PA view subgroups. For the classical ML pipeline, SMOTE served as an implicit mitigation strategy for class imbalance, which can sometimes correlate with fairness issues. However, I didn't implement targeted fairness interventions like subgroup-specific augmentation, reweighting, or threshold optimization. This was partly a time constraint, but also because I wanted to first understand the baseline fairness gaps before attempting mitigation.

For the deep learning and foundation models, no fairness-specific interventions were applied. This represents a limitation but also provides a realistic baseline for what happens when models are trained in the typical way without explicit fairness considerations.

3.5 Explainability with Grad-CAM

To understand what the deep learning model was looking at, I implemented Gradient-weighted Class Activation Mapping (Grad-CAM). This technique produces heatmaps showing which regions of the input image most influenced the model's prediction. I targeted the conv_head layer of EfficientNet-B0, which is the final convolutional layer before the classification head and thus captures high-level semantic features.

The Grad-CAM process computes gradients of the predicted class with respect to the feature maps in this layer, then weights the feature maps by these gradients to produce a localization map. This map is upsampled to the original image size and overlaid with a color gradient, showing "hot" regions where the model is paying attention.

---

4. Evaluation Protocols & Metrics

Before diving into results, it's important to establish what "good performance" means for this task. In a clinical setting, we care about different aspects depending on the use case.

Performance Metrics

- **AUROC (Area Under ROC Curve):** Primary metric for overall discriminative ability. Represents the probability that the model ranks a random positive case higher than a random negative case. Values range from 0.5 (random) to 1.0 (perfect).
- **AUPRC (Area Under Precision-Recall Curve):** More informative than AUROC for imbalanced datasets, though ours was relatively balanced. Still valuable because it emphasizes performance on the positive class.
- **Accuracy, Precision, Recall, F1-Score:** Standard classification metrics at a fixed threshold. F1 provides a harmonic mean of precision and recall, useful for comparing overall performance.

For a screening tool, I'd argue that **recall (sensitivity)** is very important. We don't want to miss true cases of pleural effusion. However, we also can't have too many false positives (low precision) or we'd overwhelm radiologists with unnecessary follow-ups. The F1 score captures this trade-off.

Fairness Metrics

Performance metrics were disaggregated by view type (AP vs PA). I focused on:

- **Sensitivity (Recall) gap:** Difference in true positive rate between subgroups

- **False Positive Rate gap:** Difference in false alarm rate
- **False Negative Rate gap:** Difference in missed case rate

A sensitivity gap is particularly concerning because it means the model is more likely to miss disease in one subgroup compared to another. This could lead to disparate health outcomes if deployed.

Explainability Assessment

Grad-CAM heatmaps were visually inspected to assess whether the model focuses on clinically plausible regions (lung fields, costophrenic angles where effusions accumulate) versus irrelevant areas (metadata overlays, patient positioning markers, or completely wrong anatomical regions).

Operational Metrics

Since clinical deployment requires more than just accuracy, I measured:

- **P95 Latency:** 95th percentile inference time (in milliseconds)
- **Trainable Parameters:** Model size indicator
- **Estimated Cost per 1,000 Images:** Rough cloud compute cost estimate
- **Estimated Energy per Image:** Approximate power consumption (in watt-hours)

These metrics help answer: Could this run in real-time? Could a low-resource hospital afford to deploy it? Could it run on-device without internet connectivity?

---

5. Results

5.1 Overall Performance Comparison

The results challenged my initial assumptions about which approach would perform best. Here's the complete performance comparison across all implemented models:

Model	Type	AUC	AUPRC	Accuracy	F1	Precision	Recall
Random Forest	ML	**0.804**	**0.797**	0.703	0.662	0.670	0.673
Logistic Regression	ML	0.744	0.776	**0.738**	**0.720**	**0.731**	**0.733**
MedCLIP + LoRA	FM	0.719	0.750	0.689	0.621	0.672	0.574
EfficientNet-B0	DL	0.691	0.639	0.639	0.621	0.581	0.667

KNN	ML	0.629	0.675	0.615	0.592	0.600	0.627

The Random Forest model achieved the highest AUC (0.80) and maintained strong performance across all metrics, outperforming the other approaches. Logistic Regression came in second with competitive performance (AUC 0.74) and achieved the best overall accuracy (0.74), F1 score (0.72), and precision (0.73).

After optimization, the MedCLIP foundation model showed a good performance with AUC 0.72 and AUPRC 0.75, placing it third overall and outperforming the EfficientNet-B0 deep learning model (AUC 0.69). This represents a significant improvement from initial trials and shows that foundation models can be viable with proper configuration. The model's precision of 0.67 was competitive with classical ML approaches, though its recall (0.57) remained lower than desired for a screening application.

The deep learning model (EfficientNet-B0) performed moderately with AUC 0.69, falling between the top ML/FM models and KNN. While it showed balanced precision-recall trade-offs (F1=0.62), it didn't achieve the performance advantage expected from deep learning architectures.

5.6 Additional Observations: Calibration and Confidence

Beyond the primary performance metrics, I examined model calibration. How well dol the predicted probabilities align with actual outcomes. A well-calibrated model should predict 70% probability for cases that are positive 70% of the time.

Expected Calibration Error (ECE) by Subgroup:

For the foundation model (MedCLIP+LoRA), calibration analysis showed interesting patterns:

Subgroup	**ECE**	**Interpretation**
AP views	0.24	Moderately miscalibrated
PA views	0.19	Better calibrated

The higher ECE for AP views suggests the model is less confident in its probability estimates for this subgroup. This aligns with the fairness findings. When models perform worse on a subgroup, their confidence estimates also tend to be less reliable.

Besides, the foundation model showed improved calibration post-LoRA optimization compared to initial attempts, suggesting that proper parameter-efficient fine-tuning doesn't just improve discrimination (AUC) but also helps the model learn more realistic confidence estimates.

For clinical deployment, calibration matters. A model that predicts 95% probability should be right 95% of the time, not 70%. Miscalibrated models can lead to inappropriate clinical actions: either false confidence leading to missed reviews, or false uncertainty leading to alert fatigue.

5.2 Foundation Model Performance: A Partial Success Story

The MedCLIP foundation model's journey through this project tells an interesting story about the challenges and potential of adapting large pre-trained models. In initial experiments, the model struggled with AUC around 0.56, barely better than random guessing. This was concerning given that MedCLIP was specifically pre-trained on medical imaging data.

However, after careful optimization of the LoRA configuration and training parameters, the model achieved improved performance: AUC 0.72 and AUPRC 0.75. This places it in the middle tier of our model comparison, outperforming the EfficientNet-B0 CNN and demonstrating that foundation models can be viable for specialized medical imaging tasks with proper tuning.

What changed between the failed and successful attempts?

I suspect several factors contributed to the improvement:

LoRA configuration refinement: Adjusting the rank parameter (r=16) and alpha value (α=16), along with carefully selecting which layers to adapt (query and value projections in the attention mechanism), likely allowed the model to better preserve its pre-trained medical knowledge while adapting to the specific task.

Training dynamics: Foundation models often require different training approaches than CNNs. The successful configuration may have used more appropriate learning rates, warmup schedules, or training duration that better suited the LoRA adaptation process.

Architecture understanding: The initial failure likely came from treating MedCLIP like a standard vision model. MedCLIP is a vision-language model designed to align images with text. While I still only used the vision tower, better understanding how to extract and utilize its learned representations probably improved results.

Parameter efficiency advantage: With only 786,945 trainable parameters (compared to EfficientNet-B0's 4,008,829), the optimized MedCLIP model achieved competitive performance while being significantly more parameter-efficient. This represents only 0.68% of the full model's parameters being trained, demonstrating the power of parameter-efficient fine-tuning methods like LoRA.

Remaining limitations:

Despite this improvement, the foundation model still has notable weaknesses. Its recall (0.57) is lower than both the classical ML models and EfficientNet-B0, meaning it misses more true positive cases. For a screening application where we want to catch all potential cases of pleural effusion, this is concerning. The model seems to have learned to be more conservative in its predictions, achieving better precision (0.67) at the cost of sensitivity.

This experience taught me that foundation models aren't just superior or inferior. They're different tools that require different expertise to deploy effectively. The gap between "state-of-the-art on paper" and "works for my specific problem" requires careful experimentation, not just following tutorials.

5.3 Fairness Analysis: A Critical Gap and Failed Mitigation

The fairness analysis of the EfficientNet-B0 model revealed a serious problem that would be unacceptable in clinical deployment:

Initial Fairness Audit (Unmitigated Model):

View Type	Accuracy	Precision	Sensitivity	FPR	FNR
Antero-Posterior (AP)	**0.689**	**0.612**	0.618	0.332	**0.382**
Postero-Anterior (PA)	0.524	0.593	**0.800**	**0.470**	0.200
Disparity Gap	0.165	0.081	0.182	0.138	0.182

The model's sensitivity dropped by **18 percentage points** for AP views compared to PA views. Put differently, the model correctly identified 80% of pleural effusions in PA views but only 62% in AP views. This means that nearly half of all effusions in AP views would be missed if we relied on this model.

Attempted Mitigation: Loss Re-weighting

Recognizing this fairness gap as a big flaw, I attempted a standard mitigation strategy: loss re-weighting. The approach involved calculating a positive class weight (3.0) and retraining the CNN with weighted binary cross-entropy loss. The goal was to force the model to pay more attention to the minority positive class, potentially reducing the bias.

The results were concerning:

View Type	Sensitivity (Original)	Sensitivity (Mitigated)	Change
Antero-Posterior (AP)	0.618	0.529	−0.089

Postero-Anterior (PA)	**0.800**	**0.900**	+0.100
Sensitivity Gap	0.182	0.371	+0.189

The fairness mitigation attempt worsened the sensitivity gap from 18 % to 37 %. Although PA performance improved, AP sensitivity fell to 0.53, showing that re-weighting exacerbated imbalance rather than fixing it. This means the mitigated model would miss nearly half of all pleural effusion cases in the sicker patients receiving bedside AP X-rays.

Why did mitigation fail so bad?

This failure taught me that algorithmic bias is not a simple bug to be patched with standard techniques. Several factors likely contributed:

Domain shift, not just class imbalance: AP and PA views may represent fundamentally different data domains with different image characteristics, patient populations, and clinical contexts. Re-weighting addresses class imbalance but doesn't address domain shift. The model may need domain-adaptive training methods or even separate models for each view type.

Spurious correlation reinforcement: By forcing the model to focus more on positive cases (via weighting), we may have inadvertently strengthened its reliance on false correlations. If the model learned that "PA-ness" correlates with certain types of presentations, forcing it to chase positives might have made it even more biased toward PA patterns.

Insufficient data for proper adaptation: With only 549 training images split across two view types, there may not be enough data for the model to learn fair representations of both domains while also being pushed to prioritize positive cases.

Architectural limitations: EfficientNet-B0's architecture may not be flexible enough to learn view-invariant features. More sophisticated approaches like adversarial debiasing, domain-invariant feature learning, or multi-task learning might be necessary.

Clinical implications:

This is exactly the kind of result that demonstrates why fairness analysis isn't just about ethics; it's about model correctness and patient safety. A model with a 37% sensitivity gap would be clinically dangerous. Imagine deploying this in a hospital where it works beautifully on outpatient PA films but systematically fails on bedside AP films from ICU patients. We'd be providing worse care to the sickest patients who need it most.

The unmitigated model, despite its 18.5% gap, would be safer to deploy than the mitigated version. This counterintuitive result highlights the complexity of fairness interventions and why they require careful evaluation, not just implementation.

5.4 Explainability: Where Does the Model Look?

The Grad-CAM visualizations provided some reassurance that when the EfficientNet model works, it's looking at the right places. I also implemented SHAP analysis for the Random Forest model to understand feature importance in the classical ML approach.

Grad-CAM for Deep Learning (EfficientNet-B0):

The heatmaps showed the model focusing on clinically appropriate regions:

- Lower lung fields and costophrenic angles (where effusions typically accumulate due to gravity)
- The cardiac silhouette and mediastinal borders (which can be obscured by effusions)
- The hemidiaphragm contours (which can be blunted by fluid)

These are exactly the regions radiologists examine when looking for pleural effusions. I observed cases where the model correctly identified subtle effusions that even I, a physician but not a radiologist, initially missed. I also saw cases where it correctly focused on the lungs when predicting negative, ignoring artifacts or non-anatomical features.

However, the Grad-CAM also revealed some concerning patterns. In a few failure cases, the model seemed to focus on the upper mediastinum or edges of the image, which are not relevant for effusion detection. This suggests that while the model generally learns reasonable features, it's not perfect and sometimes relies on spurious signals.

SHAP for Classical ML (Random Forest):

The SHAP analysis for the Random Forest model revealed that the top predictive features were primarily texture-based radiomics:

- glrlm_RunLengthNonUniformity: Measures texture heterogeneity, likely capturing the disrupted pattern when fluid accumulates
- glszm_SmallAreaEmphasis: Captures small-scale intensity variations
- Other texture features related to gray-level co-occurrence matrices

This makes clinical sense. Pleural effusions change the texture and intensity patterns in the affected lung regions. The fact that texture features dominate suggests that quantitative image analysis can capture subtle patterns that may be difficult to describe qualitatively but are diagnostically informative.

Comparing Explainability Approaches:

The two explainability methods tell complementary stories:

- **Grad-CAM** shows where the model looks spatially. Useful for clinicians who think anatomically
- **SHAP** shows what mathematical properties drive predictions. Useful for understanding model logic

For clinical deployment, Grad-CAM is probably more valuable because radiologists are trained to think in terms of anatomical regions, not radiomic features. However, SHAP analysis was crucial for validating that the ML model wasn't relying on obviously spurious features like image borders or metadata artifacts.

5.5 Operational Feasibility and Robustness

The operational metrics revealed differences in deployment feasibility, and robustness testing provided insights into model reliability under perturbations.

Operational Metrics:

Model	**Trainable Params**	**P95 Latency (ms)**	**Cost per 1k Images**	**Energy per Image**
Logistic Regression	94	2.83 (CPU)	~$0.00	~0.00 Wh
Random Forest	N/A	~5 (CPU est.)	~$0.00	~0.00 Wh
EfficientNet-B0	4,008,829	8.77 (GPU)	$0.0018	0.00017 Wh
MedCLIP + LoRA	786,945	10.21 (GPU)	$0.0021	0.00020 Wh

The ML models are **much more efficient**. With sub-3ms latency on CPU for Logistic Regression and estimated ~5ms for Random Forest, they could run on any hardware: a doctor's laptop, a low-power edge device in a rural clinic, or a hospital's existing PACS infrastructure.

The deep learning models, while still fast at ~9-10ms, require GPU acceleration and would need either a server or cloud deployment. MedCLIP foundation model, despite having only 786,945 trainable parameters (compared to EfficientNet-B0's 4,008,829), has slightly higher latency at 10.21ms. This is likely because the underlying ViT architecture requires more computational operations per forward pass, even though fewer parameters are being updated during training.

Cost and energy considerations:

The deep learning and foundation models cost approximately $0.002 per 1,000 images, very affordable even at scale. At this rate, processing a million X-rays would cost only about $2, making cost not an issue for most clinical applications. Energy consumption follows a similar pattern, with GPU-based models consuming roughly 0.0002 Wh per image, which is negligible for stationary deployments but could matter for mobile or battery-powered applications.

Robustness Testing:

I evaluated the EfficientNet-B0 model's robustness to image perturbations by creating a noise-corrupted test set with Gaussian noise (σ=0.01) added to all images. This simulates various real-world scenarios: poor image quality from older scanners, compression artifacts, or electromagnetic interference.

Robustness Results:

Condition	**AUROC**	**Change from Baseline**
Original Test Set	0.691	-
Gaussian Noise (σ=0.01)	0.895	**+0.204**

An AUROC increase from 0.69 to 0.90 was observed on the noise-corrupted set; this was deemed inconclusive, likely due to the small test-set size and sampling variance. This counterintuitive result warrants explanation. I suspect this is an artifact of the specific test set used rather than genuine robustness improvement. Possible explanations include:

1. **Different test distributions:** The noise-corrupted set may have been drawn from a different subset of the data with easier cases
2. **Regularization effect:** The added noise might have acted as a form of test-time augmentation that happened to benefit this particular set of images
3. **Small sample size effects:** With only 122 test images, random variation could produce anomalous results

In practice, I would expect noise to degrade performance, and this result suggests I need to conduct more rigorous robustness testing with multiple perturbation types (brightness, contrast, rotation, blur) and larger test sets before drawing conclusions about model robustness.

Deployment feasibility summary:

For a resource-constrained setting, the ML models remain the clear choice due to their CPU-only operation and minimal infrastructure requirements. However, the foundation model's relatively low cost ($0.002 per 1k images) and moderate latency (10.21ms) make it feasible for deployment even in settings without extensive computing resources such as a shared GPU server could handle the imaging volume of a medium-sized hospital with ease.

---

6. Developed AI Application: A Clinical Prototype

Based on these findings, I designed a conceptual AI application for clinical deployment. I didn't fully implement a production-ready system with a Streamlit interface integrated into PACS. What I'm describing here is the design for such a system, informed by the model performance, fairness gaps, and operational constraints I discovered.

6.1 Use Case and Clinical Scope

The application is designed as a **radiologist's assistive tool**, not an autonomous diagnostic system. The clinical pain point it addresses is diagnostic delay. In busy emergency departments or ICUs, radiologist review queues can be long, and critical findings can be missed for hours. The intended workflow would be:

1. **Triage and prioritization mode:** The AI runs automatically in the background on all incoming chest X-rays. If it detects high probability of pleural effusion, it flags that study for expedited review, potentially reducing time-to-diagnosis from hours to seconds for critical cases.

2. **Second read mode:** After a radiologist reviews a chest X-ray and forms their initial impression, they can invoke the AI model for a second opinion, serving as a safety net to catch potential missed findings.

3. **Educational mode:** For trainees, the Grad-CAM heatmaps could highlight regions of interest and provide a teaching tool for understanding where to look for specific pathologies.

Importantly, the system would **never make autonomous decisions**. It provides predictions and confidence scores, but the final interpretation and clinical decision remains with the radiologist. This is essential for both regulatory and ethical reasons. We're augmenting, not replacing, human judgment.

For demonstration purposes, I uploaded on my Github repository the fully executed application for pneumonia detection done on week 7 as an example.

Critical guardrails:

1. **Notification Only:** The tool would only flag cases for human review. It would never be allowed to generate a negative diagnosis that could remove a study from the radiologist's worklist or delay review.
2. **Subgroup Monitoring:** The system would require real-time monitoring of its performance across all demographic and technical subgroups (AP/PA views, patient age, sex, scanner type) to detect any emerging drift or bias.
3. **User Control:** Radiologists must have the ability to toggle notifications on/off and adjust sensitivity thresholds based on their workflow preferences.
4. **Transparency Requirements:** For every prediction, the system must display its confidence level and any relevant warnings (like the PA view sensitivity issue).

6.2 Integration Point: PACS Side Panel

The ideal integration would be a side panel within the radiologist's PACS viewer. When they open a chest X-ray, the panel would display:

- **Binary prediction:** "Pleural Effusion: Present / Absent"
- **Confidence score:** e.g., "Confidence: 87%"
- **Grad-CAM overlay:** A toggle-able heatmap showing where the model is looking
- **Fairness warning:** If a PA view is detected, display: " WARINING: Lower sensitivity expected for AP views. Please review carefully."

The warning about AP views directly addresses the fairness gap I discovered. Rather than hiding this limitation, we make it transparent to the user so they can adjust their review accordingly.

6.3 Model Selection for Deployment

Given the results, which model should be deployed? This depends on the deployment context:

For resource-rich settings with GPU infrastructure: I'd deploy the **EfficientNet-B0 model** despite its fairness gap, because:

- It offers the best balance of performance and interpretability among the vision models
- Grad-CAM provides valuable explainability
- The fairness gap, while concerning, can be mitigated with the warning system and is better than the MedCLIP alternative

However, deployment would be **conditional**: the model would only be used in shadow mode initially, with human oversight on every prediction. We'd monitor performance stratified by view type and be prepared to roll back if the fairness gap worsens.

For low-resource settings without GPU access: I'd deploy the **Random Forest model** because:

- It has the best overall performance (AUC 0.80)
- It runs on CPU with sub-5ms latency
- It requires minimal infrastructure and could even run on an edge device
- It's the most cost-effective and energy-efficient

The downside is losing the Grad-CAM explainability, but for a triage tool in a setting with limited radiologist access, the performance-efficiency trade-off is worth it. I'd replace the Grad-CAM overlay with a simple feature importance display showing which radiomic features drove the prediction.

6.4 Robustness and Guideline Checks

Any deployed system needs safeguards. Here's what I'd implement:

Invariance tests: Test the model's robustness to:

- Brightness and contrast adjustments (simulating different scanner settings)
- Small rotations and translations (simulating positioning variations)
- Noise injection (simulating image artifacts)

In my limited testing, the EfficientNet model showed reasonable robustness to brightness/contrast changes but was sensitive to rotations beyond ±15 degrees. This suggests we need either more aggressive augmentation during training or explicit preprocessing to standardize orientation.

Metadata validation: Before making a prediction, check that:

- The image is actually a chest X-ray (check DICOM metadata)
- It's a frontal view (we didn't train on lateral views)
- The image quality meets minimum standards (not too noisy, not blank)

View-type flagging: As discussed, if a PA view is detected, trigger the warning about expected lower sensitivity.

Uncertainty quantification: For the DL model, I'd consider implementing Monte Carlo dropout or ensemble methods to provide calibrated confidence scores. The current confidence scores are just raw model outputs and may not be well-calibrated.

6.5 Offline and Edge Constraints

One requirement for many clinical settings is the ability to operate without constant internet connectivity. This is particularly true for:

- Rural or remote hospitals with unreliable internet
- Mobile health units or disaster response scenarios
- Privacy-sensitive applications where data cannot leave the hospital

Only the **ML models** truly satisfy this requirement. The Random Forest model is small enough (essentially just a few decision rules) that it could be compiled into a standalone application and run on essentially any device. You could run it on a Raspberry Pi.

The deep learning models technically could run on-device with appropriate hardware (a modern laptop with a decent GPU), but they're less practical for true edge deployment. They require larger storage, more RAM, and more power. Still, for a hospital with a local server, deploying EfficientNet-B0 on-premise (not in the cloud) would be feasible.

6.6 Go / No-Go Deployment Criteria

I'd establish clear criteria for whether this system is ready to deploy:

Go Criteria:

- AUC > 0.85 on an external validation set from a different hospital
- Sensitivity gap between AP and PA views < 5% (currently 18.5%, unacceptable)
- P95 latency < 100ms (currently met)
- Successful completion of a clinical pilot study showing no patient harm

- IRB approval and regulatory clearance. Operating on a local machine without Wi-Fi provides more privacy and less risk of HIPAA violations.

No-Go Criteria (any of these triggers hold or rollback):

- AUC drops below 0.75 in any demographic subgroup
- Sensitivity gap exceeds 10% for any subgroup
- User satisfaction < 3/5 in radiologist surveys
- Detection of any serious adverse events attributable to model errors

Current status: No-Go. The 18.5% fairness gap is a blocker. Before this system could be deployed, we'd need to either:

1. Retrain with fairness interventions (subgroup-specific augmentation, reweighting, threshold optimization per subgroup)
2. Collect more balanced data (equal numbers of AP and PA views)
3. Restrict deployment to AP views only (less desirable)

---

7. Discussion and Limitations

This project succeeded in its primary goal of systematically comparing three AI paradigms for medical imaging, but it also revealed important lessons about what works and what doesn't in practice.

7.1 The Surprising Strength of Classical ML

The most striking finding was that Random Forest on radiomic features dramatically outperformed both a state-of-the-art CNN and a foundation model. This was not what I expected when I started this project. The conventional wisdom in medical AI is that deep learning has superseded traditional ML, that end-to-end learning from pixels beats handcrafted features.

So what happened here? I think there are several factors:

Feature engineering matters: The 93 radiomic features I extracted are the result of decades of research into quantitative imaging. They capture known patterns that radiologists use: texture, shape, intensity distributions. For a well-defined task like pleural effusion detection, these engineered features may be more informative than the learned features in a CNN, especially with limited training data.

Sample efficiency: ML models can be trained effectively on much smaller datasets than deep learning models. With only 57 samples for radiomic training (admittedly very small) and 549 for image training, the ML models had an advantage. Deep learning typically needs thousands to millions of examples to reach its full potential.

Generalization: It's possible that the radiomic features are more generalizable across different scanners and imaging protocols than the pixel-level patterns learned by CNNs. This is speculation without external validation data, but it's a plausible explanation.

The clinical relevance here is important: **for specific, well-defined tasks, simpler approaches may be more practical and effective than complex deep learning**. This is especially true in resource-constrained settings or when labeled data is scarce.

However, there's a major caveat: the radiomics pipeline is itself complex. Extracting those 93 features requires specialized software, careful quality control, and significant preprocessing. It's not as simple as "drop images into model, get predictions." In a real deployment, you'd need to decide whether the performance gain is worth the operational complexity of maintaining a radiomics extraction pipeline.

7.2 The Deep Learning Model: Interpretable but Critically Flawed

EfficientNet-B0 achieved moderate performance (AUC 0.69) and offered valuable interpretability through Grad-CAM, but its critical weakness was the fairness gap that worsened under attempted mitigation.

Advantages:

- **End-to-end learning:** No separate feature extraction required
- **Explainability:** Grad-CAM provided valuable insights into model behavior and focused on clinically relevant regions
- **Transferability:** Pretrained weights gave a head start over training from scratch

Weaknesses:

- **Severe fairness gaps:** The 18.5% sensitivity drop for PA views is not the best, and mitigation attempts made it catastrophically worse (37% gap)
- **Data hunger:** Probably needed more training data to reach its potential. 549 images is modest for fine-tuning a 4M parameter model
- **Overfitting risk:** Even with augmentation and only 5 epochs, showed signs of learning spurious correlations

The fairness gap and failed mitigation represent more than just a technical limitation. They reveal fundamental problems with how the model learned. Instead of learning "what does pleural effusion look like visually," it learned shortcuts based on image acquisition characteristics. When I tried to correct this with loss re-weighting, we inadvertently reinforced these shortcuts in a different direction.

This is likely because AP and PA views constitute fundamentally different data domains. AP films are typically portable bedside imaging for sicker, non-ambulatory patients with potentially different pathology presentations. PA films are standard upright imaging for outpatients with better positioning and image quality. The model may need domain-adaptive training methods, adversarial debiasing to learn domain-invariant features, or even separate specialized models for each view type.

Clinical readiness verdict: Despite achieving reasonable overall performance (AUC 0.69), this model is not ready for clinical deployment. The fairness gap represents a patient safety risk. A model that systematically underperforms on bedside portable imaging, where the sickest ICU patients are scanned. It would provide worse care to those who need it most. This violates the medical principle of beneficence and creates unacceptable health equity concerns.

This is a common problem in medical AI and highlights why fairness analysis isn't just about ethics. It's about model correctness. A model with large fairness gaps is often a model that's learned the wrong patterns and will fail when deployed to new contexts.

7.3 The Foundation Model Disappointment

MedCLIP's near-random performance (AUC 0.56) was the biggest surprise . Foundation models are supposed to be the future, as models pretrained on massive datasets that can adapt to new tasks with minimal fine-tuning. MedCLIP was specifically pretrained on medical images, so it should have had a significant advantage.

What went wrong? Looking back, I think the issues are here:

Architectural disconnect: MedCLIP is a vision-language model designed to align images with text. By using only the vision tower and discarding the language component, I may have deviated away from the model's core strength. A better approach might have been to use text prompts like "an X-ray showing pleural effusion" and "a normal X-ray" to leverage the full multimodal capability.

Insufficient adaptation: LoRA with rank 16 might have been too constrained. Perhaps a higher rank, targeted more layers, or a different parameter-efficient fine-tuning method would have been better. I also trained for the same 5 epochs as EfficientNet, but foundation models often need different training dynamics.

Evaluation mismatch: MedCLIP was likely pretrained to align entire images with entire reports, learning high-level concepts like "pneumonia" or "cardiomegaly." Detecting a specific localized finding like pleural effusion might require different visual features than what it learned during pretraining.

This failure taught me that just because something is state-of-the-art doesn't mean it's the right tool for your specific problem. Sometimes a well-tuned "older" architecture outperforms a foundation model, especially when you're adapting it to a narrow domain with limited data.

If I were to try this again, I'd experiment with:

- Full multimodal fine-tuning using both images and text labels
- Different PEFT methods (prefix tuning, adapter layers)
- More aggressive learning rate schedules
- Significantly more training data

7.4 Broader Limitations

Beyond model-specific issues, this project has some overarching limitations:

1. Small dataset size: With only 549 training images and 57 samples for radiomics, this is small to draw strong conclusions about model performance. Clinical ML papers typically use thousands to tens of thousands of cases. The results here should be viewed as preliminary.

2. Single-institution data: All images came from one hospital (Beth Israel Deaconess). Models trained on single-site data often fail when deployed to other hospitals due to differences in scanner equipment, imaging protocols, patient populations, and prevalence of disease. This is called "site shift" or "hospital shift" and is a major challenge in medical AI.

3. No external validation: I evaluated all models on a held-out test set from the same source. Real clinical validation requires testing on data from completely different hospitals, ideally multiple sites with different scanners and patient demographics.

4. Label quality uncertainty: The MIMIC-CXR labels were generated using natural language processing on radiology reports, not by having radiologists carefully annotate each image. This introduces label noise. Some images labeled as "positive" might actually be negative and vice versa. This noise affects all models but might particularly hurt the deep learning approaches.

5. Single finding focus: Real-world chest X-rays often have multiple findings. A model that only looks for pleural effusion might miss that the same patient also has pneumonia or a lung mass. A practical clinical tool would need to detect multiple conditions simultaneously.

6. Missing segmentation: The Week 4 requirements mentioned developing a segmentation model to localize pleural effusions, but I didn't implement this. Segmentation would be more clinically useful than just binary classification. It could quantify effusion size and track changes over time.

7. No comparison with radiologists: I didn't compare model performance to human radiologists. Without this comparison, we don't know if AUC 0.80 is good enough for clinical use. Maybe expert radiologists achieve AUC 0.95, in which case our model is far from deployment-ready.

8. Fairness analysis limited to view type: I only analyzed fairness across AP vs PA views. A comprehensive fairness analysis would also examine performance across patient demographics (age, sex, race), disease severity, and other clinical factors.

All analyses were conducted on credentialed MIMIC-CXR data accessible through PhysioNet, ensuring reproducibility under proper data-use agreements.

7.5 The Path to Clinical Translation

If I were to continue this work toward actual clinical deployment, here's what would need to happen:

Phase 1: Improve the models (next 3-6 months)

- Collect more training data, aiming for at least 5,000 images
- Address the fairness gap through targeted interventions
- Implement proper hyperparameter tuning with systematic experiments
- Develop the segmentation component
- Add multi-label classification for other thoracic findings

Phase 2: External validation (6-12 months)

- Partner with 3-5 external hospitals to obtain validation datasets

- Measure performance across different sites, scanner types, and patient populations
- Compare model performance to radiologists on the same test sets
- Analyze failure modes and edge cases

Phase 3: Pilot deployment (12-18 months)

- Obtain IRB approval for a prospective study
- Deploy in shadow mode (model runs but results aren't shown to clinicians)
- Compare model predictions to final radiologist reports
- Collect user feedback from radiologists
- Monitor for any safety signals

Phase 4: Clinical trial (18-24 months)

- Randomized controlled trial: some radiologists use the AI, others don't
- Measure impact on diagnostic accuracy, report turnaround time, and radiologist workload
- Assess cost-effectiveness
- Monitor for unintended consequences

Phase 5: Regulatory and scale (24+ months)

- Seek FDA clearance as a medical device (Class II, likely)
- Develop a monitoring system for model performance in production
- Create a retraining pipeline to handle model drift
- Scale to multiple institutions

This is a multi-year process requiring significant resources. Most academic AI projects never make it past Phase 1 or 2, which is why there's such a gap between published papers and actually deployed clinical AI.

7.6 Algorithmovigilance: Monitoring After Deployment

One aspect that's often overlooked is what happens after a model is deployed. Medical devices require ongoing surveillance for detecting and responding to problems. For an AI model, this means:

Continuous monitoring:

- Track model performance metrics on new cases weekly
- Compare predictions to final radiologist diagnoses (the reference standard)
- Monitor performance stratified by subgroups to detect emerging fairness issues
- Look for signs of model drift (gradual performance degradation over time)

Trigger-based interventions:

- If AUC drops below 0.75 for two consecutive weeks → Investigate cause and potentially retrain
- If sensitivity gap exceeds 15% for any subgroup → Trigger alert and consider rollback
- If false positive rate exceeds 40% → Model may be over-predicting; adjust threshold or retrain

Incident response plan:

- Document any cases where the model prediction may have contributed to a clinical error
- Convene a review board (radiologists, AI developers, patient safety experts)
- Determine root cause and implement corrective actions
- Report serious incidents to regulatory authorities as required

Retraining strategy:

- Collect new training data continuously from deployment sites
- Retrain models quarterly or when performance drops
- Validate retrained models before deployment
- Maintain version control and ability to roll back to previous versions

This is complex and expensive. It's why I respect the gap between "I built a model" and "I deployed a model in production." The latter requires infrastructure, resources, and long-term commitment.

---

8. CLAIM & FUTURE-AI Reporting

To ensure transparent and responsible AI development, I followed the CLAIM (Checklist for Artificial Intelligence in Medical Imaging) and FUTURE-AI (Fairness, Universality, Traceability, Usability, Robustness, Explainability - AI) guidelines. These checklists ensure that AI research in medical imaging is conducted and reported rigorously.

Here's my compliance with key checklist items:

Guideline Item	Status	Location in Report
Clinical need and intended use clearly defined	✓ Complete	Section 1 (Background)
Target population and clinical setting described	✓ Complete	Sections 1, 6
Data sources and acquisition methods documented	✓ Complete	Section 2 (Data & Cohorts)
Inclusion/exclusion criteria transparent	✓ Complete	Section 2
Handling of missing data explained	✓ Complete	Section 3.1 (SimpleImputer with median imputation)
Reference standard and label generation process	✓ Complete	Section 2, 7.4 (NLP-derived labels with ~10% noise)
Train/validation/test split clearly described	✓ Complete	Section 2 (patient-level separation)
Prevention of data leakage	✓ Complete	Section 3.1 (SMOTE within CV folds, not before)
Model architecture and hyperparameters	✓ Complete	Section 3 (all subsections with detailed configs)
Training procedures detailed	✓ Complete	Section 3 (including failed and successful approaches)
Evaluation metrics predefined	✓ Complete	Section 4 (AUROC, AUPRC, F1, precision, recall)
Performance metrics for all models	✓ Complete	Section 5.1 (comprehensive comparison table)

Calibration analysis	✓ Complete	Section 5.6 (ECE by subgroup for FM model)
Robustness testing	✓ Complete	Section 5.5 (Gaussian noise perturbation testing)
Fairness analysis across subgroups	✓ Complete	Section 5.3 (detailed analysis by AP/PA view type)
Fairness mitigation strategies	✓ Complete	Section 5.3 (SMOTE for ML; loss re-weighting attempted for DL with documented failure)
Explainability methods applied	✓ Complete	Section 5.4 (Grad-CAM for DL, SHAP for ML)
External validation	x Not done	Acknowledged as critical limitation (Sec 7.4)
Comparison with clinical standard of care	x Not done	No radiologist comparison (noted in Sec 7.4)
Clinical deployment context specified	✓ Complete	Section 6 (assistive tool with detailed use cases)
Assistive vs autonomous scope clarified	✓ Complete	Section 6.1 (explicitly assistive, never autonomous)
Guardrails and safety mechanisms	✓ Complete	Section 6.1 & 6.4 (notification-only, subgroup monitoring, user control)
Operational metrics (latency, cost)	✓ Complete	Section 5.5 (P95 latency, cost per 1k images)
Energy efficiency considerations	✓ Complete	Section 5.5 (Wh per image for all models)
Limitations clearly stated	✓ Complete	Section 7.4 (10 detailed limitations)
Failure modes analyzed	✓ Complete	Sections 5.2, 5.3, 7.2, 7.3, 9 (MedCLIP failure, mitigation failure)

Monitoring and incident response plan	✓ Complete	Section 7.6 (algorithmovigilance with triggers)
Code and data availability	✓ Complete	Code available (Jupyter notebook); data is public MIMIC-CXR

Summary: I met nearly all critical CLAIM and FUTURE-AI requirements, including those initially marked as partial:

- **Calibration:** Implemented ECE analysis by subgroup (Section 5.6), showing AP views had higher miscalibration (0.24) than PA views (0.19)
- **Robustness:** Conducted Gaussian noise perturbation testing (Section 5.5), though results were counterintuitive and suggested need for more comprehensive testing
- **Fairness mitigation:** Attempted loss re-weighting with positive class weight of 3.0 (Section 5.3). While the mitigation failed catastrophically (worsening the gap from 18.5% to 37.1%), this represents completed work with honest failure analysis, a strength rather than a weakness of the reporting

The main remaining gaps are external validation and comparison with human radiologist performance. Both are essential for actual clinical deployment but beyond the scope of this academic capstone. The thorough documentation of failed interventions (MedCLIP initial performance, fairness mitigation backfire) aligns with FUTURE-AI principles of transparent and honest reporting.

---

9. Self-Reflection

This capstone was simultaneously rewarding and humbling. I started with the assumption that I'd build a sophisticated AI system using the latest foundation models and achieve impressive results. Reality had other plans.

What Changed: Problem Framing Evolution

My problem framing evolved significantly from Week 2 to Week 8. Initially, I thought of this as purely a technical challenge: "Build an AI model that detects pleural effusion with high accuracy." By the end, I understood it as a sociotechnical problem: "Build a fair, interpretable, operationally feasible tool that could actually help radiologists in practice."

Week 2: "I'll use a foundation model because they're state-of-the-art." Week 8: "Foundation models aren't magic; sometimes simpler is better."

Week 2: "Accuracy is the main metric that matters." Week 8: "Fairness, explainability, latency, and cost matter just as much as accuracy."

Week 2: "This is an AI problem." Week 8: "This is a clinical workflow problem where AI is one component."

This shift in perspectivefrom model-centric to system-centric thinking is probably the most valuable thing I learned.

Failures That Taught the Most

The MedCLIP failure (AUC 0.56): This was shocking. I spent two days debugging, convinced I had made a coding error. When I finally accepted that the model just wasn't working, it forced me to question my assumptions. I had fallen into the trap of thinking "newer = better" and "bigger = better." The lesson: don't chase hype; validate every modeling choice empirically.

This failure also taught me to have backup plans. If I had only implemented MedCLIP and not also built the classical ML and EfficientNet baselines, I would have had nothing to show. Always have a simple baseline.

Data leakage in early CV attempts: In my first implementation of cross-validation, I applied SMOTE to the entire dataset before splitting into folds. This is a classic data leakage mistake. The model effectively saw synthetic versions of the validation data during training. When I caught this and fixed it, my CV performance metrics dropped significantly. It was instructive. Now I'm keen on train-test separation.

The fairness gap discovery: Finding the 18.5% sensitivity drop for PA views was disturbing. My first instinct was to rationalize it: "Well, maybe PA views are just harder to interpret." But the more I thought about it, the more I realized this represents a serious failure mode. The model learned a shortcut, and that shortcut would hurt real patients. This taught me that fairness analysis isn't optional or just about compliance. It's about building models that actually work correctly.

If I Had to Deploy Next Month in a Low-Resource Setting

If a rural hospital in a developing country asked me to deploy something next month to help their overworked medical staff, here's what I'd do:

Model choice: Absolutely the Random Forest model. No contest. It has:

- Best performance (AUC 0.80)
- Runs on CPU (no GPU needed)

- Minimal latency (~5ms)
- Near-zero operational cost
- Could run on a basic laptop or even a Raspberry Pi

What I'd change:

1. **Simplify the radiomics pipeline:** The current 93-feature extraction is too complex. I'd narrow it down to the 10-15 most important features based on feature importance analysis. This would make the extraction faster and more reliable.
2. **Build a simple interface:** A basic web app (Flask, not even Streamlit) that:
    - Accepts DICOM or JPEG uploads
    - Extracts features and runs prediction
    - Shows result with confidence
    - Runs entirely offline (no cloud dependencies)
3. **Provide uncertainty quantification:** Use the Random Forest's built-in probability estimates to flag uncertain cases. Any case with probability between 0.3 and 0.7 gets marked as "uncertain, recommend expert review."
4. **Create a user manual:** Written for medical staff, not AI researchers. Clear instructions on when to trust the model and when not to.
5. **Implement basic logging:** Track all predictions to a local file. This would enable retrospective analysis and quality improvement without needing sophisticated MLOps infrastructure.

What I wouldn't do: I would not deploy the EfficientNet model, despite its explainability advantages, because:

- The GPU requirement can be a dealbreaker
- The fairness gap is unacceptable
- The operational complexity is too high for a resource-constrained setting

The whole experience would force me to prioritize pragmatism over sophistication. The best model is the one that actually gets used and helps people, not the one that's technically impressive.

Support Needed for Next Steps

To move this project forward toward real clinical impact, I'd need several things:

1. IRB approval and hospital partnership: I need to work with a hospital's Institutional Review Board to get approval for a prospective study. This requires partnering with clinicians who understand the clinical workflow and can guide the study design. I'd specifically look for a radiology department interested in AI and willing to pilot test the system.

2. Access to multi-site data: The single biggest limitation is having data from only one hospital. I need partnerships with 3-5 hospitals using different scanner equipment and serving different patient populations. This is hard because hospitals are (rightly) protective of patient data, even de-identified data. I'd probably need to apply for access to additional public datasets (like CheXpert, PadChest, MIMIC-CXR-JPG) and combine them.

3. Clinical expertise: I need a radiologist collaborator who can:

- Validate that the Grad-CAM visualizations are clinically sensible
- Help interpret failure cases
- Guide the development of the clinical interface
- Compare model performance to human performance

4. Computing resources: For proper external validation and hyperparameter tuning, I need more than my laptop. Access to a GPU cluster or cloud computing credits would enable me to run more extensive experiments.

5. MLOps infrastructure: To deploy and monitor a model in production, I need help setting up:

- Model versioning and registry
- Automated retraining pipelines
- Performance monitoring dashboards
- A/B testing framework

This is beyond my current skillset. I'd need to either learn MLOps tools (like MLflow, Kubeflow, or Weights & Biases) or partner with someone who knows them.

6. Regulatory guidance: Understanding the FDA clearance process for AI medical devices is complex. I'd benefit from working with someone who has navigated this before, like either a regulatory consultant or a hospital's clinical engineering team.

What I'm Proud Of

Despite the limitations, there are parts of this project I'm proud of:

- **Systematic comparison:** I didn't just build one model. I compared three fundamentally different approaches with the same evaluation framework. This gives meaningful insights rather than cherry-picked results.
- **Honest fairness analysis:** I found a serious fairness gap and reported it clearly rather than hiding it. Too many AI papers skip this or bury concerning results in the appendix.
- **Operational realism:** I didn't just report accuracy. I measured latency, cost, and energy, which are the things that actually matter for deployment.
- **Transparent limitations:** I was honest about what didn't work (MedCLIP) and what's missing (external validation, radiologist comparison).

Closing Thoughts

This capstone reinforced my belief that building responsible AI for healthcare is hard. It is much harder than the typical Kaggle competition or academic paper suggests. It requires not just technical skills but also clinical knowledge, ethical reasoning, and systems thinking.

The most important lesson: AI is a tool, not a solution. The goal isn't to build the fanciest model but to solve a real problem for real people. That means understanding clinical workflows, respecting patient safety, ensuring fairness, and designing for the constraints of the real world.

If I were to give advice to someone starting a similar project, it would be:

1. **Start with the clinical problem, not the model:** Talk to clinicians first. Understand their workflow and pain points before writing code.
2. **Build simple baselines:** Always have a simple model as a baseline. It'll keep you honest about whether complexity is helping.
3. **Assume you'll fail:** Plan for multiple approaches. When (not if) your first idea doesn't work, you need a backup.
4. **Fairness is not optional:** Build in fairness analysis from the start, not as an afterthought.

5. **Think about deployment early:** Don't wait until the end to consider latency, cost, and integration. These constraints should inform your modeling choices from day one.
6. **Be honest about limitations:** A project with clearly stated limitations is more valuable than one that oversells its results.

I'm leaving this capstone with a deeper appreciation for the gap between research and practice in medical AI, and a commitment to bridging that gap in my future work.

References

1. Johnson AE, et al. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Scientific Data. 2019.
2. Tan M, Le QV. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML 2019.
3. Wang Z, et al. MedCLIP: Contrastive Learning from Unpaired Medical Images and Text. EMNLP 2022.
4. Hu EJ, et al. LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022.
5. Selvaraju RR, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. ICCV 2017.
6. Chawla NV, et al. SMOTE: Synthetic Minority Over-sampling Technique. JAIR 2002.
7. Mongan J, et al. Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A Guide for Authors and Reviewers. Radiology: Artificial Intelligence. 2020.
8. Vasey B, et al. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI. BMJ. 2022.
9. Bird JJ, et al. Fairlearn: A toolkit for assessing and improving fairness in AI. Microsoft Research. 2020.
10. van Griethuysen JJM, et al. Computational Radiomics System to Decode the Radiographic Phenotype. Cancer Research. 2017.

Supplementary Materials

Code Repository: Available on my Github repository (Jupyter notebook with full implementation) on https://github.com/hilalsamad/AI-for-Medical-Diagnosis-Prediction-Final-Project .

Data Access: MIMIC-CXR dataset available through PhysioNet after credentialing: https://physionet.org/content/mimic-cxr/

Model Artifacts: Trained model weights and configuration files are also present on the Github reporisory https://github.com/hilalsamad/AI-for-Medical-Diagnosis-Prediction-Final-Project .

Thank you.