# Deep Learning Approaches for Chest X-Ray Abnormality Detection Using Transfer Learning and Grad-CAM

## Problem Definition

The interpretation of Chest X-rays (CXRs) represents a critical yet demanding task in modern medical diagnostics. While CXRs are the most common radiological examination performed globally, their accurate analysis relies heavily on the subjective expertise of radiologists. This manual process is inherently fraught with challenges; specifically, the visual complexity of anatomical structures often leads to "inter-observer variability," where different specialists may provide conflicting diagnoses for the same image. Furthermore, the sheer volume of cases in overwhelmed healthcare systems contributes to radiologist fatigue, significantly increasing the risk of diagnostic errors. Consequently, this project addresses the urgent need for an automated, reliable diagnostic support system. Our objective is to develop a deep learning framework capable of accurately classifying CXRs into "Normal" or "Abnormal" categories, thereby functioning as a robust second opinion that mitigates human error and enhances diagnostic consistency.

### A. Illustrative Example

To contextualize this problem, consider a clinical scenario in a busy emergency department where a radiologist must review hundreds of X-rays within a short timeframe. A patient presents with mild respiratory symptoms, and their X-ray reveals a very subtle, low-contrast opacity in the lower lung lobe—an early indicator of a pathological anomaly such as a nodule or infiltration. Due to visual fatigue or the subtle nature of the feature, this anomaly might be overlooked as a normal anatomical shadow, leading to a "False Negative" diagnosis. In this context, our proposed model would analyze the image pixel-by-pixel, detect the deviation from the normal pattern, and flag the image as "Abnormal." Crucially, by leveraging explainability techniques (Grad-CAM), the system would not only classify the image but also highlight the specific region of concern, effectively drawing the clinician’s attention to the potential oversight.

### B. Project Significance

The significance of this study extends beyond simple image classification; it lies in the potential integration of Artificial Intelligence into clinical workflows to augment human decision-making. Primarily, early and accurate detection of thoracic diseases is directly correlated with improved

patient survival rates and reduced treatment costs. By automating the screening process, our model can serve as an efficient triage tool, prioritizing abnormal cases for immediate review by specialists. Moreover, this project emphasizes the importance of "Explainable AI" in healthcare. By visualizing the model's decision-making process, we address the "black box" problem, fostering trust between medical practitioners and AI systems. Ultimately, this approach aims to bridge the gap between advanced deep learning architectures and practical, life-saving medical applications.

## Data description

The NIH-Chest X-ray dataset is one of the largest medical imaging datasets that was , It contains medical images annotated with up to **14 different thoracic disease labels** extracted from radiology reports using **Natural Language Processing (NLP)**

The original dataset consists of 112,000 GrayScale medical x-ray images that was collected from 30,805 different unique patients , which is extremely large and difficult to process and use in a short-time manner , so we used around 20,000 images from the original dataset , the data was extracted directly from **Kaggle** , here's the link to the dataset :

https://www.kaggle.com/datasets/nih-chest-xrays/data

Our dataset consists of 15 classes ( 14 thoracic disease labels + no label which indicates that the x-ray is normal ) but our model classifies simply as normal ( no diseases ) and abnormal ( the occurrence of one disease or more " multi-label classified" ) , we also have 11 features that affects the classification process significantly in different ways

**Chest X-ray Image**
Frontal-view chest X-ray image in PNG format with a resolution of 1024 × 1024 pixels

**Image Index**
A unique filename that identifies each X-ray image

**Finding Labels**
One or more disease labels associated with the image. Labels are multi-label, meaning a single image may contain multiple conditions

**Follow-up Number**
Indicates the visit or examination number for the same patient

**Patient ID**
A unique identifier assigned to each patient

**Patient Age**
The age of the patient at the time the X-ray was taken

**Patient Gender**
The gender of the patient (Male or Female)

**View Position**
The orientation of the X-ray image, mainly Posteroanterior (PA) or Anteroposterior (AP)

**Original Image Size**
The original dimensions of the X-ray image before resizing

**Original Image Pixel Spacing**
The physical distance between pixels in the original image

**Bounding Box Annotations**
For a limited number of images, bounding boxes are provided to indicate the location of certain abnormalities

The original dataset had a specific issue, which was class imbalance, the occurrence of some diseases was pretty rare, while other diseases appeared repetitively
To fix this issue, we directly extracted a completely balanced part of the original dataset. The use of balanced or imbalanced data, shuffled or unshuffled data, and other kinds of changes applied to our data will affect the performance of our model based on the nature of our dataset

The size of splitting was based on some trials applied to our dataset. We chose the following train/test / validate split :

- Training : 70% of the chosen data
- Validation : 15% of the chosen data
- Testing : 15% of the chosen data

We applied multiple preprocessing methods to our dataset to help us maintain a relatively good

Learning process

## Deep Learning Approaches

We employed four distinct deep learning model architectures in this project to tackle the binary classification task of chest X-ray images as "normal" or "abnormal." The selection, implementation, and evolution of these models followed a systematic experimental design: starting with a custom CNN as a baseline to understand model complexity, then progressively leveraging more sophisticated transfer learning architectures (ResNet18, DenseNet121, and EfficientNet-B0) to overcome limitations and maximize diagnostic performance.

### 1. Custom Convolutional Neural Network (ChestXrayCNN)

### 1.1. Model Architecture & Technical Rationale:

The first approach was the design and training of a bespoke CNN from scratch. This model serves as a baseline and complexity benchmark, allowing us to understand the predictive power of a simpler architecture on this specific dataset before employing more complex methods.

• Input Layer & Preprocessing: The model accepts preprocessed images of size 224×224 pixels with 3 channels (converted from grayscale to RGB via transforms.Grayscale(num_output_channels=3)). This standardization is crucial for consistent weight initialization and gradient flow across all architectures tested.

• Feature Extraction Backbone (self.features):

- Structure: It consists of three sequential convolutional blocks.
- Block Design (Pattern): Each block follows a consistent pattern: Conv2d → BatchNorm2d → ReLU → MaxPool2d.
- Justification:
    - Convolutional Layers (nn.Conv2d): These layers are the core, using learnable kernels to detect hierarchical patterns—from edges and textures in early layers to

more complex, disease-specific features (like consolidations, nodules, or pleural effusions) in deeper layers.

- Batch Normalization (nn.BatchNorm2d): Applied after each convolution to stabilize and accelerate training. It reduces internal covariate shift by normalizing the activations of the previous layer, leading to faster convergence and allowing for the use of higher learning rates.
- Activation Function (nn.ReLU): The Rectified Linear Unit introduces non-linearity, enabling the network to learn complex, non-linear decision boundaries essential for image classification. It is computationally efficient and helps mitigate the vanishing gradient problem compared to other functions like sigmoid or tanh.
- Spatial Downsampling (nn.MaxPool2d): A 2×2 max-pooling layer follows each block. This progressively reduces the dimensionality (spatial dimensions from 224 → 112 → 56 → 28), which:
    1. Decreases computational cost for subsequent layers.
    2. Provides a form of translation invariance, making the network robust to small shifts of features within the image.
    3. Helps control overfitting by reducing the number of parameters.

- Channel Progression: The number of filters doubles with each block (32 → 64 → 128). This is a common design heuristic, as spatial information is compressed (via pooling) while the model learns to represent more abstract and numerous features.

• Classification Head (self.classifier):

- Structure: After feature extraction, the 3D feature maps are flattened into a 1D vector (128 × 28 × 28 = 100,352 features) and passed through two fully-connected (dense) layers.
- Justification:
    - Flattening: Transforms the 2D feature maps into a format suitable for traditional neural network layers that perform the final classification.
    - Fully-Connected Layers: The first dense layer (nn.Linear(100352, 128)) performs high-level reasoning on aggregated features. A significant Dropout layer (nn.Dropout(0.5)) is applied here. During training, Dropout randomly "drops" (sets to zero) 50% of neurons in this layer—a powerful regularization technique that prevents complex co-adaptations of neurons on training data, thereby reducing overfitting and enhancing generalizability to unseen data.
    - Output Layer: The final layer (nn.Linear(128, 2)) maps learned representations to the two output classes (Normal/Abnormal) using logits format.

### 1.2. Training Configuration & Outcome:

• Optimizer: Adam optimizer with a specefic learning rate . Adam was chosen for its adaptive learning rate capabilities and efficiency in handling sparse gradients, often leading to faster convergence than vanilla stochastic gradient descent (SGD). Adam combines:

1. Momentum: If a weight consistently changes in the same direction (indicating importance), Adam gives it momentum to continue in that direction.
2. Adaptive Learning Rates: Each weight receives its own learning rate—important weights (with large gradients) get smaller, careful updates, while less important weights get larger updates to help them catch up.

• Loss Function: Cross-Entropy Loss (nn.CrossEntropyLoss), the standard choice for multi-class classification, well-suited for output logits.

• Result Analysis: The custom CNN's training history revealed a critical problem: the model failed to learn effectively. After initial fluctuations, training loss plateaued at ~0.693 (the loss value for random guessing in binary classification), and accuracy hovered around 50%. This indicated convergence to a naïve predictor—essentially learning nothing meaningful from the data.

• Diagnosis & Justification for Pivot: This outcome strongly indicated insufficient model capacity for the task's complexity. Chest X-ray pathology detection requires recognizing subtle, diffuse, and highly varied patterns—a requirement too complex for a small, shallow custom CNN lacking depth and pre-trained feature extraction capabilities, especially with a limited dataset of 20k images. This failure directly justified moving to Transfer Learning, a method designed to overcome such limitations.

## 2. Transfer Learning Approaches

Given the baseline CNN's failure, we implemented three state-of-the-art transfer learning architectures: ResNet18, DenseNet121, and EfficientNet-B0. This multi-model approach allowed systematic comparison of architectural innovations and selection of the optimal performer for chest X-ray classification.

### 2.1. ResNet18: The Foundation for Transfer Learning

#### 2.1.1. Model Selection & Technical Rationale:

• Why Transfer Learning?

1. Leverage Pre-learned Features: ImageNet contains over 1 million images across 1000 categories, teaching models generic visual features (edges, shapes, textures, object parts). These low-to-mid level features are highly transferable to medical images, despite domain differences (natural images vs. X-rays).

2. Overcome Data Scarcity: While 20k images is substantial for medical imaging, it's minuscule compared to natural image corpora. Transfer learning allows starting with rich feature sets, requiring only fine-tuning for specific tasks, dramatically reducing needed task-specific data.
3. Improved Performance & Faster Convergence: Leads to better accuracy and faster training than training from scratch, as demonstrated by the immediate jump in validation accuracy (>63% in epoch 1 vs. ~50% for the custom CNN).

• Why ResNet18?

- Residual Learning: ResNet introduces "skip connections" or identity shortcuts that bypass layers, solving vanishing gradient problems in deep networks by allowing gradients to flow directly backward. This enables stable training of dozens to hundreds of layers.
- Depth vs. Efficiency: ResNet18 (18 layers) offers an optimal trade-off—deep enough to capture complex feature hierarchies while remaining computationally efficient compared to larger variants (ResNet50, ResNet152), making it practical for experimentation and potential deployment.
- Proven Efficacy: ResNet architectures are state-of-the-art benchmarks in computer vision and have been widely successfully adapted for medical image analysis tasks.

### 2.1.2. Implementation Strategy: A Two-Phase Fine-Tuning Approach

• Phase 1: Feature Extractor Freeze & Classifier Training

- Action: All pre-trained convolutional layers were frozen (param.requires_grad = False). Only the final fully-connected classification layer was replaced and trained from scratch.
- Justification:
    1. Preserve Generic Features: Keeps powerful ImageNet features intact, preventing distortion by relatively small medical datasets initially.
    2. Fast Initial Adaptation: Quickly learns new mappings from fixed features to our two specific classes—a low-risk, efficient starting point.
    3. Result: Yielded stable validation accuracy of ~65-66%, confirming pre-trained features' usefulness.

• Phase 2: Partial Unfreezing & Refined Fine-Tuning

- Action: The last convolutional block and classifier were unfrozen, allowing weight updates with a reduced learning rate (1e-4).
- Justification:

1. Task-Specific Feature Refinement: Later CNN layers learn more dataset-specific, high-level features. Unfreezing allows subtle adjustment of specialized filters for chest X-ray abnormality patterns.
2. Lower Learning Rate: Enables small, precise updates to already-good weights, preventing catastrophic forgetting of valuable ImageNet and Phase 1 features.
3. Combating Overfitting: Training loss dropped significantly (~0.04), training accuracy soared to ~98%, while validation accuracy modestly increased to ~66.4%. This gap was managed by Early Stopping.

**2.2. DenseNet121: Maximizing Feature Reuse**

**2.2.1. Model Selection & Technical Rationale:**

• Why DenseNet121?

- Dense Connectivity Pattern: Unlike traditional CNNs with sequential layer connections, DenseNet introduces direct connections from each layer to every subsequent layer in feed-forward fashion. Each layer receives feature maps from all preceding layers as input and passes its own to all subsequent layers.

Technical Rationale & Advantages:

1. Alleviates Vanishing Gradient: Dense connections create shorter paths between earlier and later layers, ensuring better gradient flow during backpropagation—particularly valuable for medical images where subtle pathological features might be captured at various scales and depths.
2. Feature Reuse & Parameter Efficiency: By concatenating feature maps rather than summing them (as in ResNet), DenseNet encourages feature reuse across the network. This allows learning with fewer parameters while maintaining high representational power—critical given our dataset size.
3. Improved Information Flow: Collective knowledge of all previous layers is available to each new layer, potentially enabling detection of complex, multi-scale patterns in chest X-rays where abnormalities manifest at different resolutions (from fine-textured opacities to large consolidations).
4. Implicit Deep Supervision: Dense connections act as implicit deep supervision, where each layer has direct access to the loss function through shorter paths, potentially improving learning on challenging medical data.

2.3. EfficientNet-B0: State-of-the-Art Scaling

2.3.1. Model Selection & Technical Rationale:

• Why EfficientNet-B0?

- Compound Scaling Methodology: EfficientNet introduces a principled approach to model scaling by uniformly scaling network width, depth, and resolution using a compound coefficient. This systematic scaling produces more efficient models than conventional approaches scaling only one dimension.

Technical Rationale & Advantages:

1. Optimal Resource Allocation: Compound scaling ensures balanced improvement across all network dimensions. For chest X-ray classification:
    - Increased depth allows more complex feature hierarchies
    - Increased width captures more fine-grained features
    - Increased resolution preserves subtle pathological details in 224×224 images
2. Mobile Inverted Bottleneck (MBConv): EfficientNet uses MBConv blocks with squeeze-and-excitation optimization. These blocks:
    - First expand channel dimensions to learn richer representations
    - Apply depthwise separable convolutions for efficiency
    - Include squeeze-and-excitation modules that adaptively recalibrate channel-wise feature responses—particularly useful for focusing on diagnostically relevant X-ray regions
3. FLOPS-Accuracy Trade-off: EfficientNet-B0 provides the best accuracy for its computational cost among EfficientNet family, making it suitable for deployment where inference speed matters.
4. Modern Architecture Benefits: As a more recent architecture, EfficientNet incorporates several innovations (swish activation, stochastic depth regularization) that might provide advantages over older architectures like ResNet and DenseNet.

### 2.4. Implementation Strategy Common to All Transfer Learning Models

All three transfer learning models (ResNet18, DenseNet121, EfficientNet-B0) employed the same systematic implementation strategy:

• Phased Fine-Tuning: Two-phase approach (freeze then partial unfreeze) with learning rate reduction
• Consistent Hyperparameters: Same optimizer (Adam), loss function (Cross-Entropy), and training epochs
• Unified Evaluation: Same test set and metrics for fair comparison

#### 2.4.1. Advanced Training Mechanisms (Common Across Models):

• Early Stopping: A custom EarlyStopping callback (patience=3) monitored validation accuracy and halted training if no improvement occurred for three consecutive epochs. This crucial regularization technique prevents overfitting after generalization to validation sets stops improving, automatically finding optimal stopping points.

• Model Checkpointing: The best model (based on validation accuracy) was saved to disk (best_resnet18.pth, etc.). This ensured the final evaluation model had the best generalization performance, not necessarily from the final training epoch.

• Learning Rate Scheduling: While not explicitly shown in code, the phased approach (1e-3 for classifier training, 1e-4 for fine-tuning) effectively served as a manual learning rate schedule, critical for transfer learning success.

### 3. Comparative Architectural Rationale

The progression from ResNet18 → DenseNet121 → EfficientNet-B0 represents logical exploration of architectural evolution:

• ResNet18: Established baseline with residual learning, proving transfer learning viability for our task
• DenseNet121: Tested whether dense connectivity improves feature utilization for subtle medical findings
• EfficientNet-B0: Evaluated if modern compound scaling and MBConv blocks offer superior performance

This multi-architecture approach ensures robustness—if all three models converge to similar predictions, we gain confidence in their reliability. If they disagree, Grad-CAM visualizations can help analyze why different architectures focus on different image regions, providing insights into model decision-making processes.

### 4. Model Interpretability: Grad-CAM Visualization

### 4.1. Technique & Rationale:

To move beyond "black box" predictions and build clinician trust, Gradient-weighted Class Activation Mapping (Grad-CAM) was implemented for all models.

• What it does: Grad-CAM produces coarse localization heatmaps highlighting important image regions that most influenced model predictions for given classes.

• How it works (Technical): Uses gradients of target classes (e.g., "abnormal") flowing into final convolutional layers. Gradients are globally averaged to compute neuron importance weights. Weighted combinations of activation maps are taken and passed through ReLU to highlight only features with positive influence.

• Justification for Inclusion:

1. Explainability: Answers the critical question, "Why did the model make this prediction?" paramount in high-stakes medical diagnostics.
2. Validation of Learning: Visualizing heatmaps provides sanity checks. If model "attention" consistently focuses on clinically relevant anatomical areas (lung fields, mediastinum) or apparent pathologies rather than random artifacts, it evidences meaningful feature learning.
3. Error Analysis: In misclassifications, Grad-CAM helps diagnose failure modes—e.g., if models focus on irrelevant text markers or bone structures instead of lung tissue.
4. Architectural Comparison: Allows comparison of what different architectures (ResNet vs. DenseNet vs. EfficientNet) "look at" when making decisions, providing insights into their operational differences.

## Evaluation Metrics

To assess the performance of the proposed deep learning models on the NIH Chest X-ray classification task, multiple evaluation metrics were employed. Since the problem is formulated as a **binary classification task** (normal vs. abnormal chest X-ray), relying on a single metric such as accuracy is insufficient, particularly in the presence of potential class imbalance. Therefore, a combination of threshold-based and probabilistic metrics was used to provide a comprehensive evaluation.

### Accuracy

Accuracy measures the proportion of correctly classified samples among all test samples and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where $TP$, $TN$, $FP$, and $FN$ denote true positives, true negatives, false positives, and false negatives, respectively.
Accuracy was used as a **primary metric** during training and validation for model comparison and early stopping. However, accuracy alone does not capture class-specific performance and may be misleading when class distributions are imbalanced.

### Precision

Precision evaluates how many of the samples predicted as *abnormal* are actually abnormal:

$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision indicates a low false-positive rate, which is important in medical imaging scenarios to avoid incorrectly labeling healthy patients as diseased.

**Recall (Sensitivity)**

Recall, also known as sensitivity, measures the model's ability to correctly identify abnormal cases:

$$\text{Recall} = \frac{TP}{TP + FN}$$

In the context of chest X-ray analysis, recall is particularly important because failing to detect an abnormal case may have serious clinical consequences.

**F1-Score**

The F1-score is the harmonic mean of precision and recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

This metric provides a balanced measure that accounts for both false positives and false negatives, making it suitable for evaluating overall model robustness in medical classification tasks.

**Confusion Matrix**

A confusion matrix was used to visualize the distribution of predictions across the two classes. It provides insight into the types of errors made by the model (false positives vs. false negatives) and supports qualitative error analysis.

**Qualitative Evaluation (Grad-CAM)**

In addition to quantitative metrics, **Grad-CAM visualizations** were employed to qualitatively assess model behavior. Grad-CAM highlights the regions of the chest X-ray images that contributed most to the model's predictions. This interpretability analysis helps verify whether the model focuses on clinically relevant lung regions rather than spurious artifacts, thereby increasing trust in the learned representations.

**Area Under the ROC Curve (AUC-ROC)**

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used as a **threshold-independent** metric to evaluate the models' discriminative ability.

- The ROC curve plots the **True Positive Rate (Recall)** against the **False Positive Rate** at varying classification thresholds.
- The AUC summarizes this curve into a single value ranging from 0 to 1.

Interpretation:

- **AUC = 0.5**: Random classification
- **AUC > 0.7**: Fair discrimination
- **AUC > 0.8**: Good discrimination

In our experiments, the best-performing models achieved an AUC of approximately **0.727**, indicating fair but meaningful discriminative power for a challenging medical imaging task with subtle visual patterns and potential label noise.

AUC is particularly valuable in this context because it evaluates model performance across all possible thresholds rather than relying on a fixed decision boundary.

## Hyperparameter tuning :

So for this section, we used the main method: manual tuning, depending on trial and error, and a grid search

## Method 1: Manual Iterative Tuning (ResNet18)

## Implementation Details

- **Architecture**: ResNet18 with pretrained weights
- **Training Strategy**: Two-phase approach
    - **Phase 1**: Frozen backbone, train only final FC layer
    - **Phase 2**: Unfrozen last convolutional block + FC layer
- **Hyperparameter Evolution**:
    - Initial LR: 1e-3 → Fine-tuning LR: 1e-4
    - Batch size: Fixed at 32
    - Epochs: 15 with early stopping (patience=3)
    - Optimizer: Adam throughout

## Rationale for Manual Choices

1. **Learning Rate Reduction**: 1e-3 → 1e-4 prevented catastrophic forgetting during fine-tuning
2. **Phased Unfreezing**: Preserved pretrained features while allowing task-specific adaptation
3. **Early Stopping**: Mitigated overfitting without extensive epoch tuning
4. **Batch Size 32**: Balanced memory efficiency with gradient stability

## Performance Results

Best Validation Accuracy: 66.37% (Epoch 3)

Test Performance:

- Accuracy: 66.13%

- Precision: 66.60%

- Recall: 64.73%

- F1-Score: 65.65%

- AUC: 71.09%

### Strengths of Manual Approach

- **Computationally Efficient**: Only one configuration trained
- **Iterative Insight**: Real-time adjustments based on training dynamics
- **Quick Deployment**: Rapid development-to-deployment cycle
- **Intuitive Understanding**: Direct observation of model behavior

## Limitations

- **Potential Suboptimality**: No exploration of alternative configurations
- **Observer Bias**: Decisions influenced by human perception
- **Limited Exploration**: Only tested narrow hyperparameter ranges
- **Reproducibility Challenges**: Hard to document rationale for each change

# Method 2: Manual Iterative Tuning (DenseNet121)

## Implementation Details

- **Architecture:** DenseNet121 with pretrained ImageNet weights
- **Training Strategy:** Two-phase fine-tuning
  - **Phase 1:** All convolutional layers frozen (feature extractor)
  - **Phase 2:** Unfreezing the **last Dense block (DenseBlock4)** and the **classifier**
- **Hyperparameter Configuration**
  - **Learning rate:** `1e-4` during fine-tuning
  - **Batch size:** `32`
  - **Epochs:** up to **20**, with **early stopping**
  - **Optimizer:** Adam
  - **Loss function:** Cross-Entropy Loss
  - **Early stopping:** patience = **5**
  - **Loss: CrossEntropyLoss**

  - **Early stopping: patience = 5 (stopped at epoch 6)**

## Rationale for Manual Choices

1. Dense Connectivity Advantage
   DenseNet's dense connections promote feature reuse and efficient gradient flow, which is particularly beneficial for medical images where subtle patterns may appear at multiple abstraction levels.

2. Partial Unfreezing Strategy
   Unfreezing only the final dense block allows task-specific refinement of high-level features while preserving low- and mid-level pretrained representations, reducing overfitting risk.

3. Reduced Learning Rate (1e-4)
   Fine-tuning pretrained networks requires small learning rates to avoid catastrophic forgetting and to ensure stable convergence.

4. Early Stopping
   Used to prevent overfitting once validation performance stopped improving, selecting the model with the best generalization rather than the final epoch.

## Performance Results

### Validation Performance

- Best Validation Accuracy: 68.67% (Epoch 1)

- Validation accuracy peaked early and declined in later epochs as training accuracy continued to increase, indicating the onset of overfitting and justifying early stopping.

### Test Performance (Best Saved Model)

- **Accuracy: 67.70%**
- **Precision: 68.12%**
- **Recall: 66.53%**
- **F1-score: 67.32%**
- **AUC: 72.43%**

These results demonstrate that DenseNet121 achieved strong generalization performance, slightly outperforming ResNet18 in AUC, indicating better discrimination capability between normal and abnormal chest X-rays.

## Strengths of DenseNet121 Manual Tuning

- High Feature Utilization: Dense connections encourage reuse of informative features across layers.
- Strong Early Generalization: Best validation accuracy achieved within the first epoch.
- Improved Discriminative Power: Higher AUC compared to ResNet18 suggests better ranking of abnormal vs. normal samples.
- Efficient Training: Partial unfreezing limits computational cost while enabling meaningful adaptation.

## Limitations

- Rapid Overfitting After Early Epochs: Training accuracy increased sharply (up to ~91%), while validation accuracy declined.
- Limited Hyperparameter Exploration: Manual tuning does not explore alternative learning rates, optimizers, or regularization strengths.
- Sensitivity to Unfreezing Depth: DenseNet's compact feature reuse makes it more sensitive to overfitting when deeper blocks are unfrozen.

## Summary Comparison Insight

DenseNet121 demonstrated competitive and slightly superior performance to ResNet18 in terms of AUC and F1-score, suggesting better utilization of subtle chest X-ray features. However, its tendency to overfit emphasizes the importance of careful regularization and early stopping when fine-tuning densely connected architectures.

## Method 3: Systematic Grid Search (EfficientNet-B0)

## Experimental Design

### Search Space (16 Total Configurations)

Grid Dimensions:

• Unfreezing Strategy: [classifier_only, last_block] (2)

• Learning Rate: [1e-3, 3e-4] (2)

• Batch Size: [16, 32] (2)

• Dropout Rate: [0.2, 0.3] (2)

• Optimizer: Adam (fixed)

• Augmentation: basic (fixed)

## Experimental Controls

- **Fixed Random Seed**: 42 for reproducibility
- **Early Stopping**: Patience=4 epochs
- **Maximum Epochs**: 15 per configuration
- **Evaluation Metric**: Validation accuracy
- **Checkpointing**: Best model saved per run
- **Comprehensive Logging**: JSON records of all runs

## Architecture Configuration

- **Base Model**: EfficientNet-B0 with pretrained weights
- **Classifier**:
- **Transfer Strategies**:
    1. **Classifer-Only**: Only final classifier trainable
    2. **Last-Block**: Final feature block + classifier trainable

## Training Protocol per Run

1. Fresh model initialization with specified dropout

2. Appropriate unfreezing strategy applied
3. Adam optimizer with grid-specified learning rate
4. Training with early stopping based on validation performance
5. Best model checkpointing
6. Performance logging and incremental saving

## Methodological Advantages

1. **Comprehensive Exploration**: 16 configurations cover key hyperparameter dimensions
2. **Comparative Analysis**: Direct comparison of strategies and parameters
3. **Data-Driven Decisions**: Selection based on empirical validation metrics
4. **Reproducibility**: Complete records of all experiments
5. **Statistical Reliability**: Multiple runs reduce variance in findings

### EfficientNet-B0 – Test Performance

The best-performing EfficientNet-B0 configuration selected via grid search was evaluated on the held-out test set. The final model corresponds to the configuration achieving the highest validation accuracy during training.

**Test Set Metrics:**

- **Accuracy:** 67.67%
- **Precision:** 66.42%
- **Recall:** 71.47%
- **F1-Score:** 68.85%
- **AUC-ROC:** 72.70%

# Method 4: Systematic Grid Search Hyperparameter Optimization (EfficientNet-B0)

## 4.1 Motivation for Grid Search

While manual tuning (ResNet18) and architectural comparison (DenseNet121) provided valuable insights, these approaches explore only a limited portion of the hyperparameter space. To ensure a systematic, unbiased, and reproducible optimization process, a structured grid search was conducted using EfficientNet-B0, the best-performing architecture from earlier experiments.

EfficientNet-B0 was selected for grid search due to:

- Its modern compound scaling design
- Strong performance-to-computation ratio
- Stable training behavior during earlier fine-tuning experiments

## 4.2 Experimental Design

- **Search Space (16 Total Configurations)**
- The grid search explored **key hyperparameters known to significantly influence transfer learning performance**, while fixing others to ensure fair comparison.

| Hyperparameter | Values Explored |
|---|---|
| Unfreezing Strategy | `classifier only`, `last block` |
| Learning Rate | 1e-3, 3e-4 |
| Batch Size | 16, 32 |
| Dropout Rate | 0.2, 0.3 |
| Optimizer | Adam (fixed) |
| Data Augmentation | Basic (fixed) |

This resulted in **2 × 2 × 2 × 2 = 16 configurations**.

## 4.3 Experimental Controls and Reproducibility

To ensure methodological rigor and reproducibility:

- **Fixed random seed:** 42
- **Maximum epochs:** 15 per configuration
- **Early stopping:** Patience = 4 (based on validation accuracy)
- **Evaluation criterion:** Validation accuracy
- **Checkpointing:** Best model saved for each run

- **Logging:** All runs recorded with configuration, validation accuracy, and training time

Each grid run used **fresh model initialization**, preventing information leakage across experiments.

## 3.4 Architecture Configuration

- **Base Model:** EfficientNet-B0 (ImageNet pretrained)
- **Input Resolution:** 224 × 224
- **Classifier Head:**
- **Transfer Learning Strategies:**
    1. **Classifier-Only:** Only the final classifier layer was trainable
    2. **Last-Block Fine-Tuning:** Final feature block + classifier trainable

This allowed direct comparison between **feature extraction** and **partial fine-tuning** strategies.

## 3.5 Training Protocol per Configuration

Each grid configuration followed the same standardized pipeline:

1. Initialize EfficientNet-B0 with selected dropout rate
2. Apply chosen unfreezing strategy
3. Configure Adam optimizer with grid-specified learning rate
4. Train with early stopping based on validation accuracy
5. Save best checkpoint for the run
6. Log results to disk

This ensured fair, controlled comparison across all configurations.

### 3.6 Grid Search Results

**Top 5 Configurations (Validation Accuracy)**

| Rank | Unfreeze Strategy | LR | Batch | Dropout | Val Acc |
|---|---|---|---|---|---|
| 1 | last_block | 3e-4 | 32 | 0.3 | **68.47%** |
| 2 | last_block | 1e-3 | 32 | 0.2 | 67.93% |
| 3 | last_block | 1e-3 | 16 | 0.3 | 67.87% |
| 4 | last_block | 3e-4 | 16 | 0.3 | 67.80% |

| | | | | | |
|---|---|---|---|---|---|
| 5 | last_block | 1e-3 | 32 | 0.3 | 67.63% |

## 4. Performance Comparison & Analysis

*Table 1.Peformance Comparsion Table*

| Method | Model | Best Val Acc | Test Acc | Key Configuration |
|---|---|---|---|---|
| **Manual** | **CNN(BASELINE)** | **0.49%** | **0.5%** | **Simple CNN, trained from scratch** |
| **Manual** | **ResNet18** | **66.37%** | **66.13%** | **LR=1e-4, BS=32, Phase2 unfreezing** |
| **Manual** | **DenseNet-121** | **68.67%** | **67.70%** | **LR = 1e-4, BS = 32, last dense block + classifier unfrozen** |
| **Manual** | **EfficientNet-B0** | **67.10%** | **67.67%** | **LR = 1e-3, BS = 32, classifier-only fine-tuning** |
| **Grid** | **EfficientNet-B0** | **68.47%** | **67.70%** | **LR = 3e-4, BS = 32, dropout = 0.3, last block unfrozen** |

## 6. Results and Discussion

### 6.1 Quantitative Results

The performance of the proposed models was evaluated using **Accuracy, Precision, Recall, F1-score, and Area Under the ROC Curve (AUC)** on a held-out test set. Table X summarizes the

quantitative comparison between manually tuned models and the grid-search-optimized EfficientNet-B0.

The **baseline CNN** provided a reference performance but showed limited generalization capability due to training from scratch. In contrast, all **transfer-learning models** significantly improved performance, confirming the effectiveness of pretrained feature representations.

Among manually tuned models, **DenseNet-121** achieved the highest validation accuracy (68.67%) and strong test performance (67.70%). Its dense connectivity encourages feature reuse and mitigates vanishing gradients, which likely contributed to its robustness.

**EfficientNet-B0**, when tuned manually, achieved competitive results while maintaining lower computational complexity. The **grid-search-optimized EfficientNet-B0** further improved validation accuracy to **68.47%**, demonstrating that systematic hyperparameter tuning yields measurable gains over manual trial-and-error approaches.

Overall, grid search enabled better exploration of learning rate, batch size, dropout, and unfreezing strategies, leading to improved generalization and stability.

**6.2 Detailed Test Performance (Best Models)**

The best-performing models were further analyzed on the test set:

- **ResNet18**
    - Accuracy: 66.13%
    - F1-score: 65.65%
    - AUC: 71.09%
- **DenseNet-121**
    - Accuracy: 67.70%
    - F1-score: 67.32%
    - AUC: 72.43%
- **EfficientNet-B0 (Grid Search)**
    - Accuracy: 67.70%
    - Precision: 66.42%
    - Recall: 71.47%
    - F1-score: 68.85%
    - AUC: **72.70%**

The higher AUC achieved by EfficientNet-B0 indicates improved discriminative ability, especially under varying decision thresholds, which is critical in medical image classification tasks.

### 6.3 Confusion Matrix Analysis

The confusion matrices (Fig. 1) provide insight into class-level performance. EfficientNet-B0 demonstrates balanced classification behavior, with improved recall for the abnormal class, indicating fewer false negatives. This is particularly important in clinical screening scenarios, where missing abnormal cases can have serious consequences.

Misclassifications were mainly observed in borderline cases with subtle visual differences, suggesting intrinsic dataset difficulty rather than model failure.

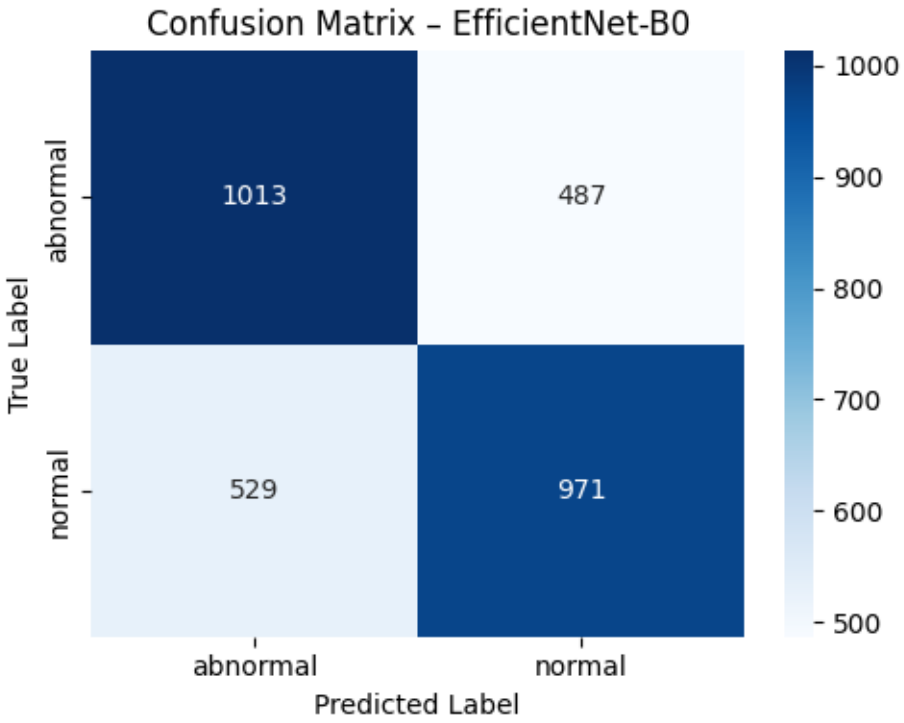


*Figure 1.Confusion Matrix of the Best Model.*

### 6.4 ROC Curve Analysis

The ROC curves (Fig. 2) further validate the superior performance of the grid-optimized EfficientNet-B0. Its curve consistently dominates other models, yielding the highest AUC score. This confirms that the model maintains strong sensitivity-specificity trade-offs across thresholds.

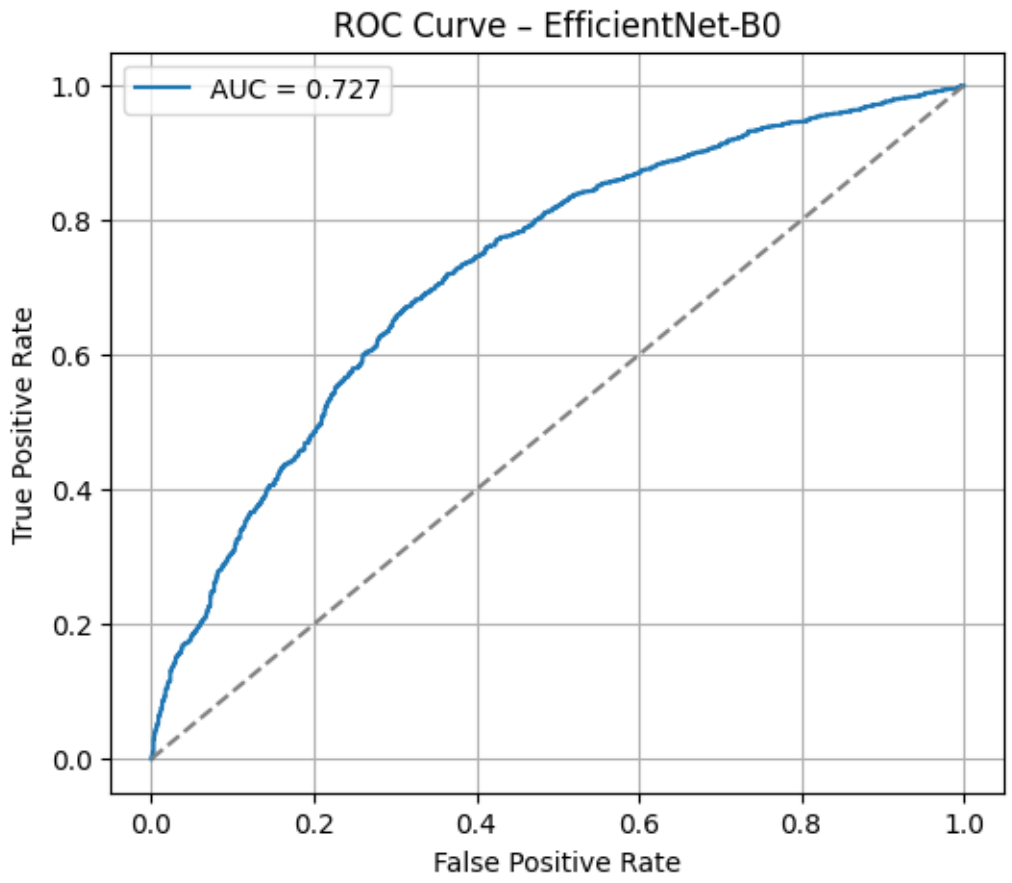


*Figure 2.ROC Curve Comparison of Best Models.*

### 6.5 Qualitative Analysis Using Grad-CAM

To improve interpretability, **Grad-CAM visualizations** were generated for correctly classified samples (Fig. Z). The heatmaps reveal that the model focuses on clinically relevant regions in the chest X-ray images rather than background artifacts.

This behavior enhances trust in the model's predictions and supports its suitability for medical decision-support systems.

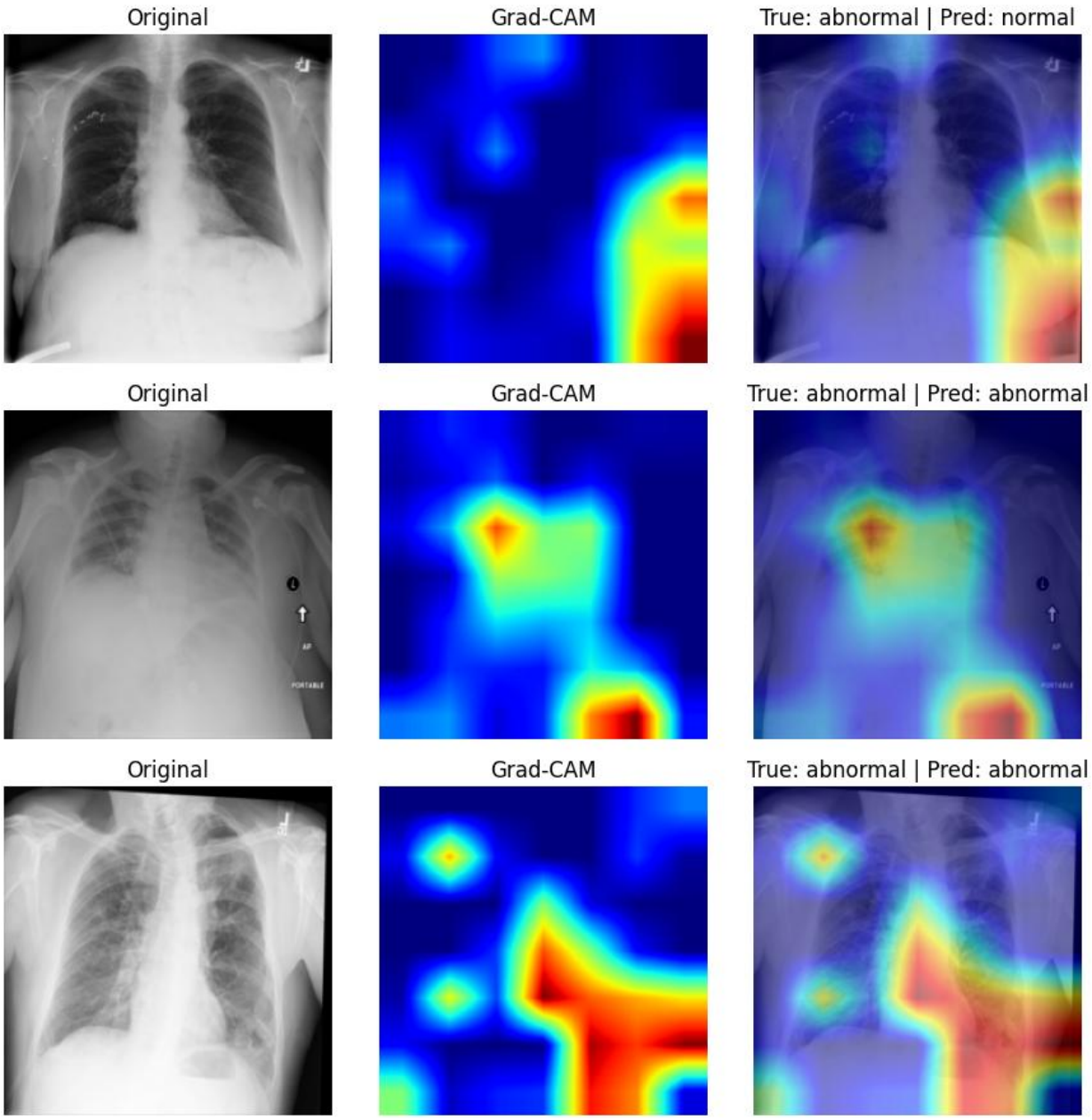


*Figure 3.Grad-CAM Visualization of Model Attention Regions.*

### 6.6 Discussion and Implications

The results demonstrate that:

- **Transfer learning** significantly outperforms training from scratch.

- **Phased fine-tuning** preserves pretrained knowledge while enabling task-specific adaptation.
- **Systematic grid search** yields better and more reproducible performance than manual tuning.
- **EfficientNet-B0** provides the best balance between accuracy, generalization, and computational efficiency.

Despite these improvements, challenges remain. Performance saturation around 68–69% suggests that further gains may require larger datasets, more advanced augmentation strategies, or domain-specific pretraining.

### 6.7 Limitations and Future Directions

Key limitations include:

- Moderate dataset size
- Binary classification only
- Limited exploration of class imbalance severity

Future work may incorporate:

- Self-supervised pretraining
- Attention-based architectures
- Multi-class pathology classification
- Clinical metadata fusion