MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

FPT UNIVERSITY

Capstone Project Document

Block-Wise Attention-Driven Soft Segmentation for Imbalanced Multi-Label Chest X-Ray Classification

<Group 4>	
Group Members	<Nguyen Huu Duy> <RollNo> <SE183995> <Hoang Khuong Duy> <RollNo> <SE184883> <Tran Khanh Nguyen> <RollNo> <SE183486> <Nguyen Khac Vuong> <RollNo> <SE183769>
Supervisor	Dr. Huynh Cong Viet Ngu
Ext Supervisor	
Capstone Project code	DSP391m

- HoChiMinh, July/2025 -

Table of Contents

Acknowledgement

We would like to thank our supervisor, Mr. Huynh Cong Viet Ngu, for his continuous guidance and feedback throughout this project. We also express our gratitude to FPT University for the opportunity and resources provided. Special thanks to our friends and family for their encouragement and support.

Definition and Acronyms

Acronym	Definition
AI	Artificial Intelligent
DL	Deep Learning
CNN	Convolutional Neural Network
AWB	Attention-Wise Block
LKA	Large Kernel Attention
AUC	Area Under the Curve
BCE	Binary Cross Entropy
CXR	Chest X-Ray

List of tables

List of figures

I. Project Introduction

1. Overview

1.1 Project Information

- **Project Title**: Block-Wise Attention-Driven Soft Segmentation for Imbalanced Multi-Label Chest X-Ray Classification
- **Team**: Group 4
- **Supervisor**: Dr. Huynh Cong Viet Ngu

1.2 Project Overview

This project investigates a novel attention-enhanced CNN architecture for the task of multi-label classification on chest X-ray images. We integrate Attention-Wise Blocks (AWBs) based on Large Kernel Attention (LKA) into DenseNet121 and VGG16, and adopt a two-stage training strategy to address class imbalance.

2. Project Background

Chest X-rays play a vital role in modern medicine as a non-invasive and widely accessible imaging modality that enables physicians to clearly observe the internal anatomy of the lungs and heart. This imaging technique provides crucial information for identifying a wide range of thoracic conditions, including pneumonia, tuberculosis, pleural effusion, and cardiomegaly, thereby serving as an essential diagnostic tool in both routine checkups and emergency care settings.

In recent years, the integration of artificial intelligence (AI) into medical imaging has opened new opportunities for improving diagnostic accuracy and efficiency. AI models, particularly those based on deep learning, have demonstrated significant potential in automating the detection of abnormalities in chest X-rays, offering decision support to clinicians by enhancing the consistency and speed of interpretation across large volumes of data.

Despite these advancements, several challenges remain in effectively applying AI to chest X-ray analysis. Datasets such as ChestXray14 are characterized by issues such as class

imbalance and the presence of multiple co-occurring pathologies within single images, which complicate the learning process for conventional convolutional neural networks (CNNs). Moreover, traditional CNNs often struggle to capture long-range spatial dependencies, limiting their ability to fully understand the global context of thoracic abnormalities. These limitations underscore the need for more sophisticated and context-aware mechanisms, such as attention-based models, to enable more robust and interpretable AI-driven diagnostic tools in clinical practice.

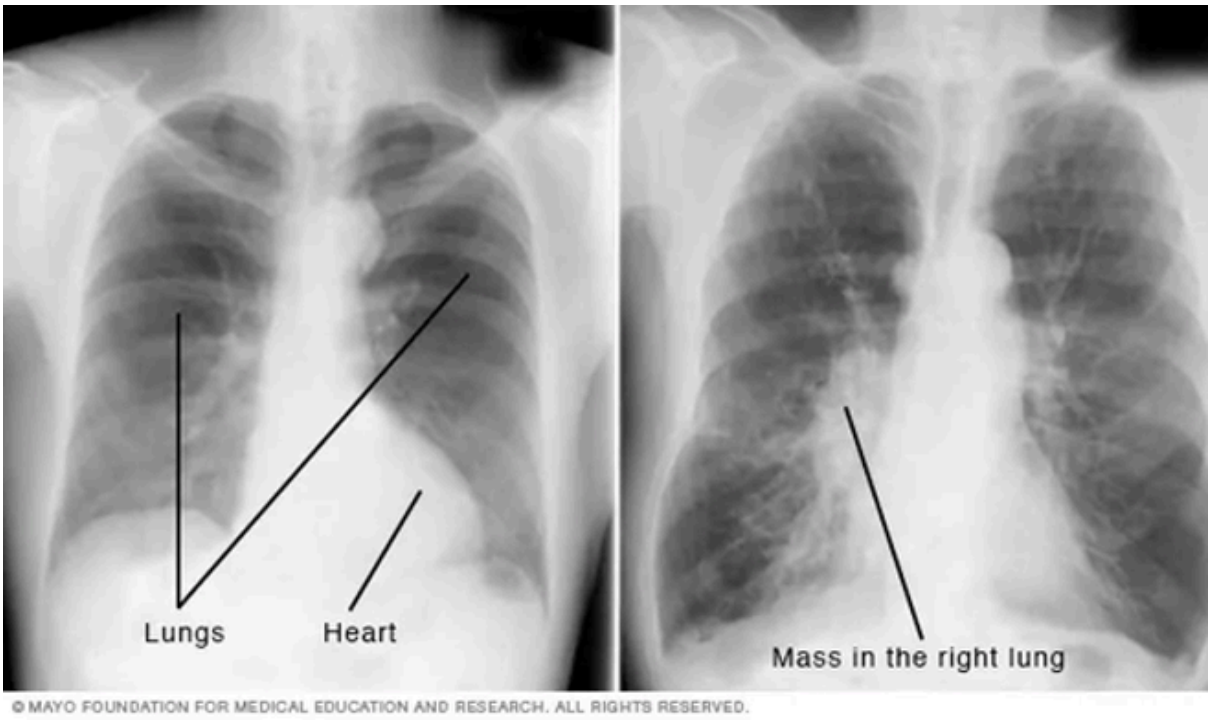


Fig 1: Examples of ChestXray Images

3. Project Objective

This project aims to enhance the diagnostic accuracy of multi-label thoracic disease classification by integrating lightweight attention modules into convolutional neural network (CNN) backbones to extract more specialized and context-aware features with minimal resource consumption. Additionally, it seeks to address the challenges posed by severe class imbalance in medical imaging datasets through the adoption of advanced loss functions such as Focal Loss, thereby ensuring more robust and reliable model performance across both common and rare disease categories.

4. Problem Statement

The task of multi-label thoracic disease classification faces several critical challenges that limit current model performance (Fig. 2). First, **severe data imbalance** across disease classes leads to biased learning, where rare conditions are often underrepresented and poorly predicted. Second, the **lack of pixel-level annotations** in large-scale chest X-ray datasets restricts the ability to localize and understand disease-specific regions, making it difficult to train models that can attend to fine-grained pathological features. Third, many

existing approaches either rely on **computationally intensive architectures**, such as Transformers, or fail to effectively capture discriminative features necessary for distinguishing between multiple **co-occurring thoracic abnormalities**, particularly when attention mechanisms lack sufficient spatial and semantic granularity.

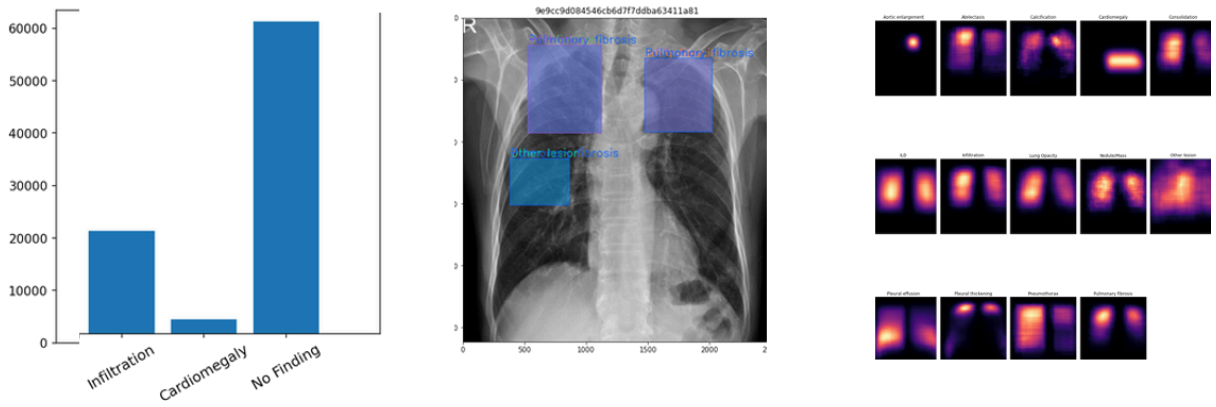


Fig 2: Common Problems in ChestXray Images Classification

5. Significance of the Project

This project demonstrates that combining lightweight attention modules with a carefully designed training strategy can substantially improve the performance of multi-label thoracic disease classification, particularly in challenging scenarios involving rare pathologies and imbalanced datasets. By achieving state-of-the-art results on the ChestXray14 benchmark—especially for underrepresented classes such as Hernia and Nodule—the proposed method highlights the effectiveness of strategically placed attention mechanisms in enhancing feature discrimination without incurring high computational costs. These findings underscore the potential of resource-efficient, attention-enhanced CNN architectures as practical and scalable solutions for improving diagnostic accuracy in real-world medical imaging applications.

6. Project Scope & Limitations

- **Scope**: Focus on multi-label thoracic disease classification using ChestXray14; implement and evaluate AWBs in CNN backbones.
- **Limitations**: Evaluation limited to one dataset; pixel-level annotations are not used; clinical deployment not covered.

II. Project Management Plan

1. Team Work

1.1 Team Structure and Roles

- Nguyen Huu Duy – Model development, Evaluation
- Hoang Khuong Duy – Experimental design, Evaluation
- Tran Khanh Nguyen – Model development, Data preprocessing
- Nguyen Khac Vuong – Experimental design, Demo system
- Dr. Huynh Cong Viet Ngu – Supervisor, technical advisor

1.2 Communication Plan

Weekly in-person meetings and asynchronous updates via email and Messenger. Github is used for document sharing and source control.

III. Existing Systems

1. Overview of the Field

Automated chest X-ray (CXR) analysis has become a critical area in medical image processing due to the high demand for scalable diagnostic tools. Chest X-rays are one of the most commonly performed radiological procedures, but their interpretation remains time-consuming and error-prone, requiring specialized radiological expertise. Deep learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool for automating disease classification from CXR images. Recent advancements also explore the use of attention mechanisms and Transformer-based architectures to further improve model focus, interpretability, and diagnostic accuracy.

2. Historical Context

- Initial efforts in CXR analysis relied heavily on handcrafted features and traditional machine learning classifiers, such as Support Vector Machines and Random Forests. These approaches required significant domain knowledge and often struggled with generalization across datasets.
- The breakthrough came with the advent of CNNs, particularly after the release of large annotated datasets like **ChestXray14** by Wang et al. (2017), which enabled end-to-end training of deep neural networks. **CheXNet** (Rajpurkar et al., 2017) was one of the first high-profile models to reach radiologist-level performance on pneumonia detection, using a DenseNet121 architecture.

3. Key Studies and Theories

3.1 Learning Technique

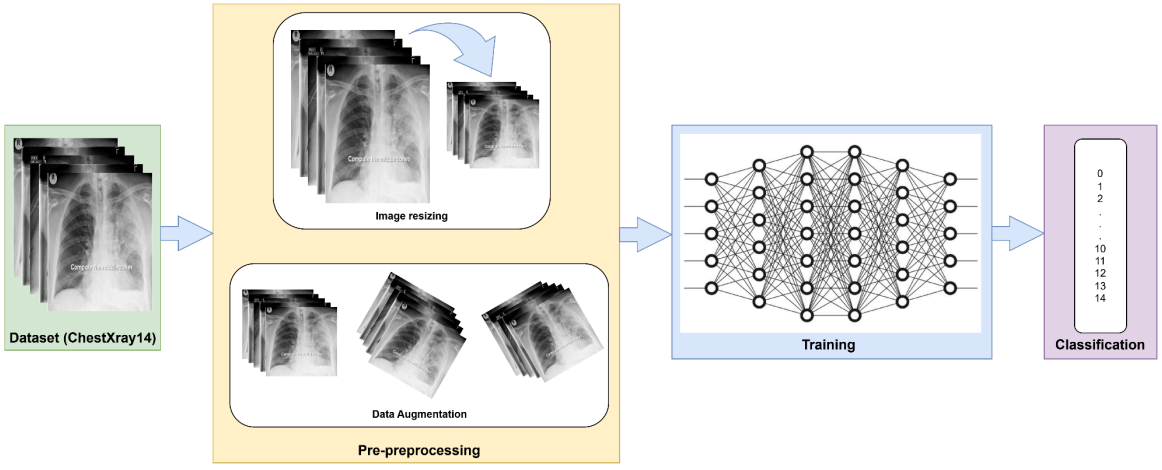


Fig 3: Workflow for ChestXray14 Classification

The common workflow for multi-label thoracic disease classification begins with **data pre-processing**, where raw chest X-ray images are cleaned and normalized to ensure consistency across samples, and then transformed into a format suitable for model input. Following this, the focus shifts to **learning the representation space**, in which the model encodes the input data into meaningful embeddings that capture essential visual patterns and disease-related features. These learned representations serve as the foundation for the next stage: **learning the classifier**, where the model maps the embeddings to corresponding disease labels and iteratively minimizes prediction errors through supervised learning.

This **three-stage pipeline** ensures that the model not only receives **high-quality** input but also builds a **robust understanding** of the underlying structures needed for accurate multi-label classification.

3.2 Backbone Research

DenseNet121 Extractor [18]

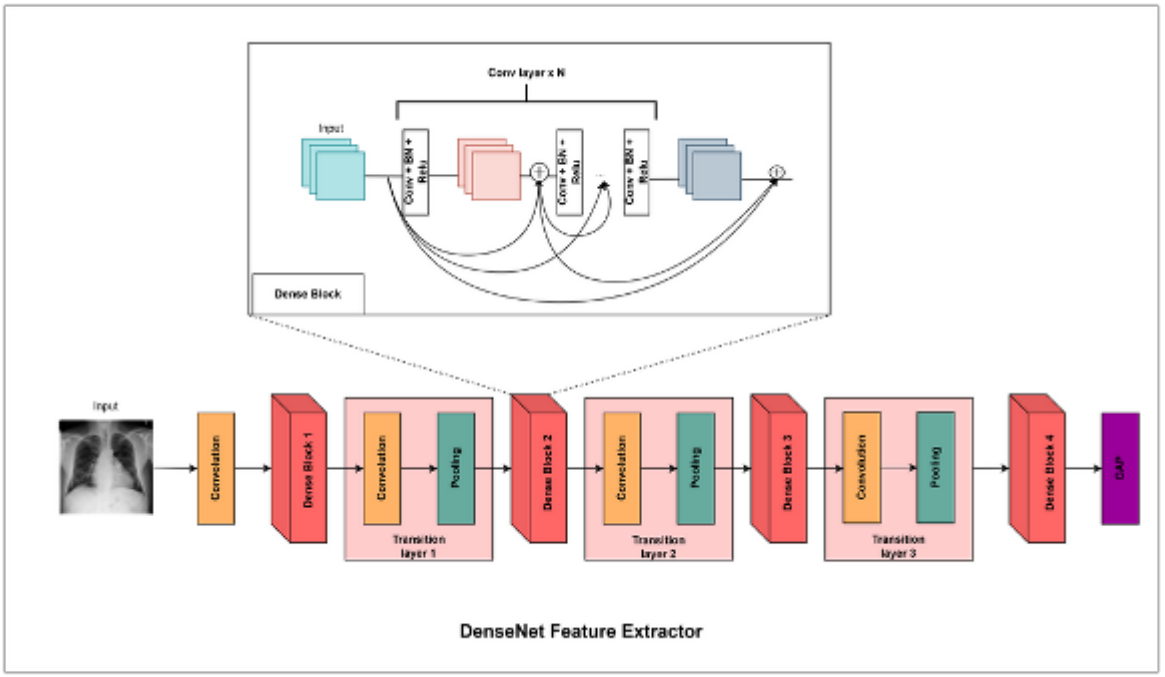


Fig 4: DenseNet121's Architecture

The core architecture of the proposed model is built upon **dense blocks**, which consist of convolutional layers followed by batch normalization and ReLU activation functions. These dense blocks are interconnected by **transition layers**, which serve two primary purposes: reducing the number of feature maps to control model complexity and performing spatial downsampling to decrease the resolution of the feature maps.

This architectural design offers several key advantages, including **mitigation of the vanishing gradient problem**, efficient **feature reuse** across layers, and a **reduction in the number of parameters** compared to traditional CNNs. However, despite these benefits, the model also faces certain limitations such as high memory consumption, increased computational demands, and added complexity in implementation due to the densely connected structure.

VGG16 Extractor [19]

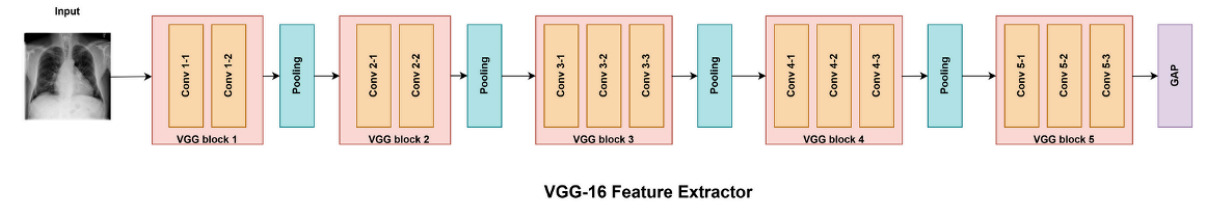


Fig 5: VGG16's Architecture

The core architecture consists of a total of **16 weight layers**, including **13 convolutional layers** and **3 fully connected layers**, following a straightforward yet effective design. It employs small 3×3 convolution filters with a stride of 1 and padding of 1 to preserve spatial resolution, and integrates 5 max pooling layers (2×2 with stride 2) to progressively reduce the spatial dimensions of the feature maps. Each convolutional layer is followed by a ReLU activation function to introduce non-linearity and improve learning capacity.

This architecture is well-regarded for its simplicity, **strong capability as a feature extractor**, and suitability for pretraining in various **transfer learning tasks**. However, it also comes with notable limitations, including a large number of parameters, which increases the risk of overfitting, especially in the absence of modern techniques such as skip connections that help stabilize deeper networks.

3.3 AG-CNN Architecture [16]

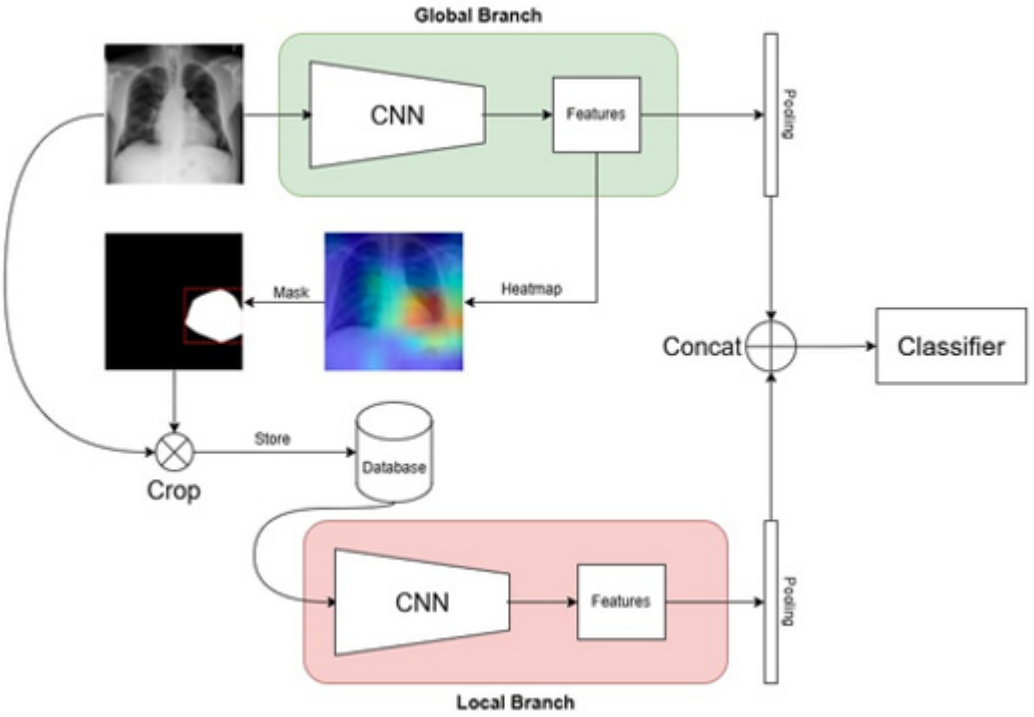


Fig 6: AG-CNN's Architecture

The proposed architecture introduces a **multi-branch** design to enhance diagnostic performance in chest X-ray classification.

- The **global branch** processes the entire X-ray image to capture broad contextual information,
- While the **local branch** leverages an attention mechanism to focus on specific pathological regions, effectively filtering out irrelevant background and emphasizing diagnostically significant areas.
- These two perspectives are then integrated in the **fusion branch**, which combines global context with fine-grained local features to produce a more comprehensive and accurate representation for classification.

This multi-branch approach significantly improves the model's ability to detect subtle abnormalities by learning both **high-level** and **detailed patterns**. However, the inclusion of three separate branches increases the overall **computational complexity**, which can pose challenges for **real-world deployment**, particularly in resource-constrained clinical environments.

3.4 Large Kernel Attention (LKA) [14]

Large Kernel Attention (LKA) enhances the spatial reasoning capabilities of convolutional neural networks by incorporating **depthwise dilated convolutions** with **large receptive fields**, allowing the model to capture long-range dependencies across the image—an ability traditionally associated with Transformer-based architectures. Additionally, LKA performs a form of **soft segmentation** by applying **element-wise multiplication** between the input feature map and a generated highlight mask, which emphasizes important regions while suppressing irrelevant background. This mechanism enables the network to focus more effectively on pathological areas within medical images. By combining **Transformer-like expressiveness** with the **computational efficiency of CNNs**, LKA offers a powerful yet lightweight attention mechanism well-suited for high-resolution and resource-constrained medical imaging tasks.

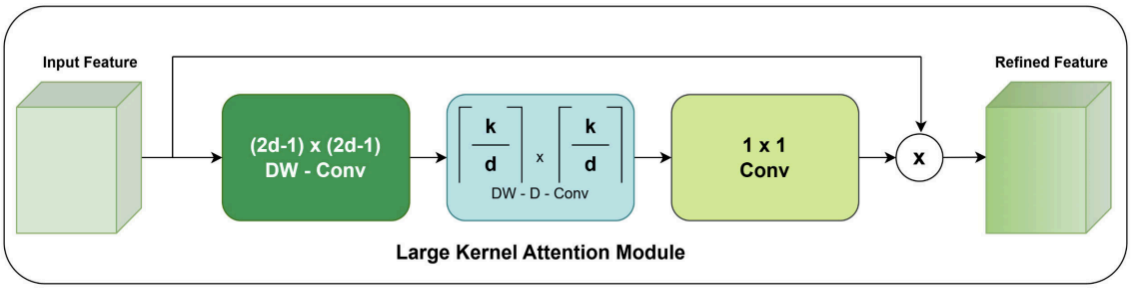


Fig 7: The Overview of LKA

Receptive field calculation:

$$R = k_2 + (k_2 - 1)(d - 1) - \text{padding} = 7 + (7 - 1)(3 - 1) - 6 = 7 + 12 - 6 = 13$$

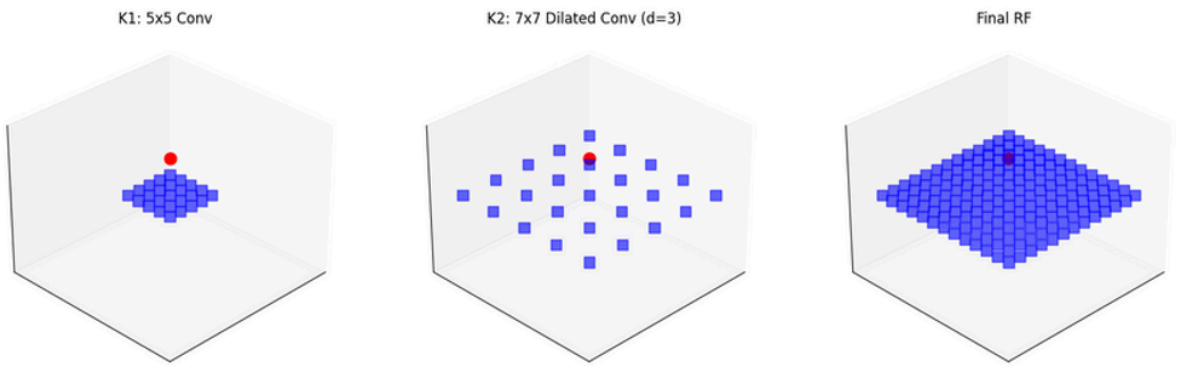


Fig 8: LKA Receptive Field Illustration

3.5 Other Works

- **CheXNet (Rajpurkar et al., 2017) [5]**: DenseNet121-based model that achieved high performance on pneumonia classification using binary cross-entropy loss. However, it struggled with class imbalance and multi-label generalization.

- **CBAM (Woo et al., 2018) [17]**: Introduced the Convolutional Block Attention Module, which enhances CNNs by applying sequential channel and spatial attention. While CBAM improved model focus on relevant areas, it was not specifically optimized for medical imaging challenges such as label imbalance.

- **SynthEnsemble (Ashraf et al., 2023) [7]**: Proposed an ensemble approach combining CNNs and Vision Transformers, showing strong performance but requiring significant computational resources.

- **SwinCheX [8] and MedViT [10]**: Transformer-based architectures that incorporate self-attention for multi-label classification. These models leverage long-range dependencies but have high memory and data requirements, limiting their practicality in low-resource settings.

4. Technological Advancements

Recent advances in deep learning for medical imaging have emphasized the integration of attention mechanisms and domain-specific architectures:

- **Hybrid Attention-CNNs**: Combining CNN backbones with lightweight attention modules has shown promising results in medical image analysis by improving feature localization and model interpretability.

- **Vision Transformers (ViT)**: Although powerful in natural image classification, ViTs generally underperform on smaller datasets like ChestXray14 without heavy pretraining or data augmentation.

- **Loss Functions for Imbalance**: Traditional models used binary cross-entropy, which is biased toward majority classes. The introduction of **Focal Loss [15]** helps address this by down-weighting well-classified examples and focusing on hard samples—particularly beneficial for rare disease detection.

5. Comparison of Existing Systems

Model/System	Backbone	Method Architecture	Imbalance Handling	Performance (AUC)
CheXNet	DenseNet121	Finetune	BCE Loss	0.841
AG-CNN	Resnet50 DenseNet121	3-Branch Attention based CNN	BCE Loss	0.868 0.871
SynthEnsemble	CNN + ViT	Ensemble Method	BCE Loss With 2 stage training	0.854

Table 1: Comparison of Existing Systems

6. Gaps in the Literature/Technology

Despite these advances, several limitations persist in the existing body of work:

- **Severe Class Imbalance**: Many large-scale Chest-XRay datasets such as ChestXray14 [1], MIMIC-CXR [2], and CheXpert [3] suffers from severe class imbalance where frequent classes dominate training, leading to biased model learning and degraded performance on underrepresented diseases. However, the issues remain insufficiently addressed in many models.We address the issue by introducing a two-stage training pipeline. The first stage pretrains only on abnormal samples, allowing the model to specialize in pathology detection. The second stage applies Focal Loss [15] to suppress the overwhelming dominance of the "No-Finding" class.
- **Fragmented Use of Attention:** Existing attention designs often emphasize either spatial or channel features, limiting their capacity to capture complex interactions between features. We proposed Attention-Wise Blocks (AWBs), based on the LKA

[14] mechanism, which can be strategically placed into a CNN backbone. These blocks jointly leverage spatial and channel attention while remaining lightweight.

- **High Computational Cost**: Recent works have explored Vision Transformers (ViTs) for chest X-ray interpretation due to their strength in modeling long-range dependencies [8] [9] [10] [11]. However, ViTs typically require massive annotated datasets and computational resources. We propose a lightweight alternative by embedding LKA-based attention within a CNN framework, which is computationally efficient while maintaining the ability to focus on critical regions.

7. Justification for the Project

The task of multi-label classification in chest X-ray analysis remains fundamentally challenged by one persistent issue: severe class imbalance. While deep learning models like CheXNet [5] have achieved strong performance on frequent disease categories, they continue to struggle with underrepresented classes such as Hernia, Nodule, and Pneumonia, which are often clinically significant.

This project distinguishes itself by addressing class imbalance in a unified, lightweight framework. The two-stage training strategy proposed in this paper is crucial in overcoming the class imbalance challenge: first, by isolating abnormal cases during pretraining to focus on pathology-specific representations; and second, by applying Focal Loss [15] in the full training phase to down-weight dominant ‘No Finding’ samples and emphasize rare diseases.

Moreover, while previous attention mechanisms enhance either spatial or channel features independently, they fall short in jointly capturing multi-dimensional dependencies critical for subtle pathology localization. The proposed method introduces Attention-Wise Blocks (AWBs), which incorporate Large Kernel Attention (LKA) modules [14] into specific high-level convolutional blocks, enabling hierarchical soft segmentation without pixel-level annotations. This combination allows the model to attend to both local and global patterns effectively.

Another key distinction is computational efficiency. Transformer-based models like SwinCheX [8] and MedViT [10] perform well but demand large-scale annotated data and high compute budgets, limiting clinical scalability. In contrast, the proposed AWB-integrated CNN retains the lightweight nature of VGG16 [19] while significantly improving its representational power via attention integration.

In summary, the proposed approach is unique in that it:

- Explicitly targets the limitations of existing systems—poor rare disease detection and isolated attention usage—by introducing a two-stage training strategy tailored for class imbalance and jointly leveraging the strength of both channel and spatial attention.

- A strategic module placement scheme that identifies optimal insertion points for AWBs, exploiting the strengths of the CNN backbone in extracting low-level visual features while enhancing high-level semantic representations to improve decision-making in pathology detection.

IV. Methodology

1. Research Questions and Objectives

This project is centered around the following research questions:

- **RQ1**: How to improve the performance of multi-label thoracic disease classification by **extracting specialized features** with low resource consumption?
- **RQ2**: Are current traditional evaluation methods suitable for the classification problem for **imbalance dataset?**

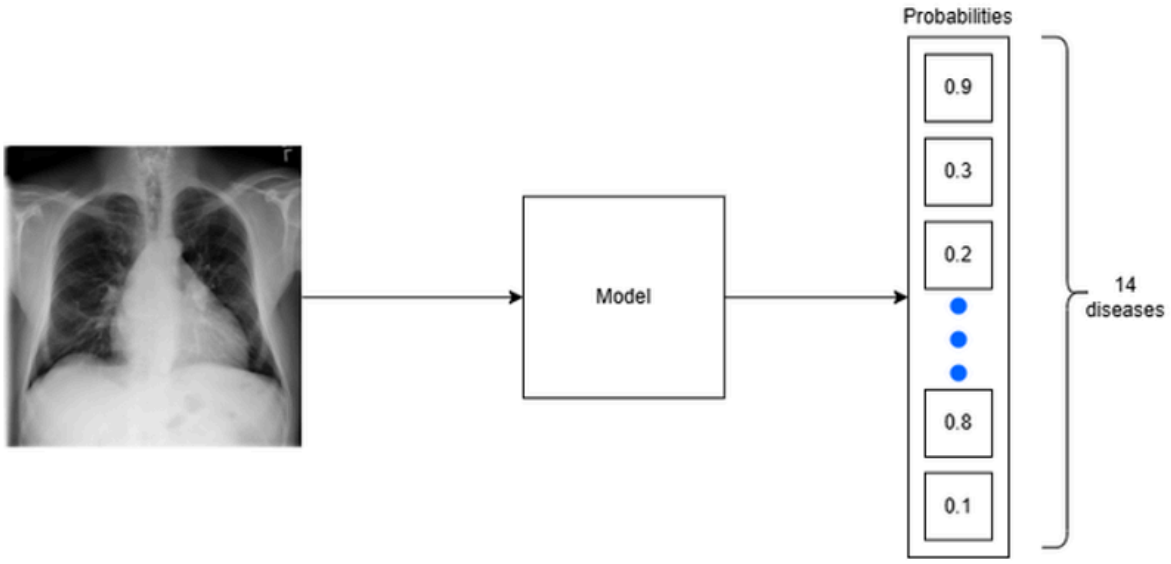


Fig 9: Common ChestXray's Classification Pipeline

To answer these, our objectives are:

- To implement and integrate Attention-Wise Blocks (AWBs) inspired by Large Kernel Attention (LKA) into the DenseNet121 and VGG16 backbone.
- To design a training pipeline that separates feature learning and class rebalancing by combining Binary Cross-Entropy and Focal Loss in sequence.
- To evaluate the model on ChestXray14 using appropriate metrics and benchmarks.

2. Data Collection and Preprocessing

2.1 Data Overview

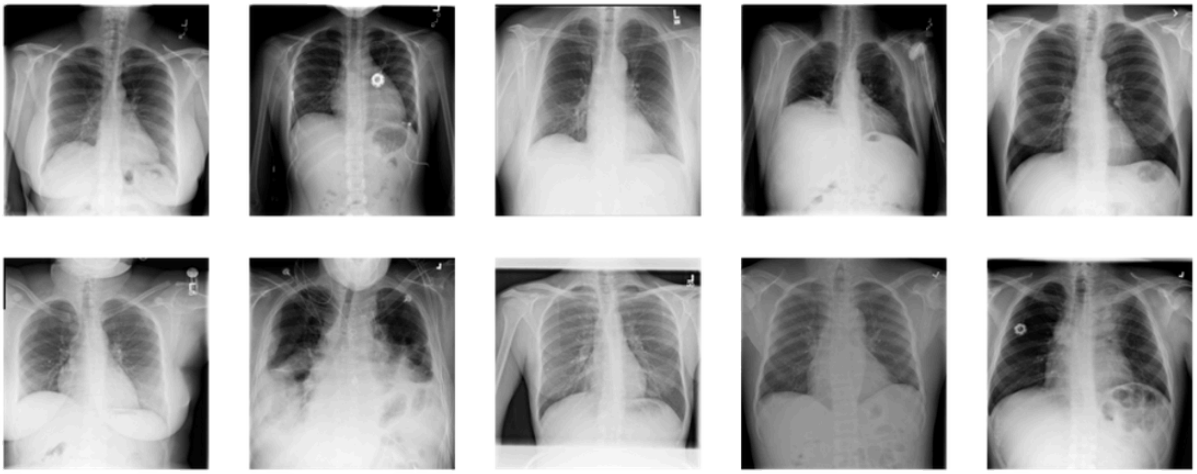


Fig 10: ChestXray14's Example Images

We use the **ChestXray14** [1] dataset, a public benchmark from the NIH Clinical Center consisting of:

- 112,120 frontal-view chest X-ray images.
- Image-level annotations for 14 thoracic diseases, including: Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis, Pleural Thickening, Hernia.
- Labels are extracted from radiology reports (which is not included) using NLP algorithms, therefore are expected to be >90% accurate and suitable for weakly-supervised learning.

2.2 Data Analysis

2.2.1 Data Imbalance

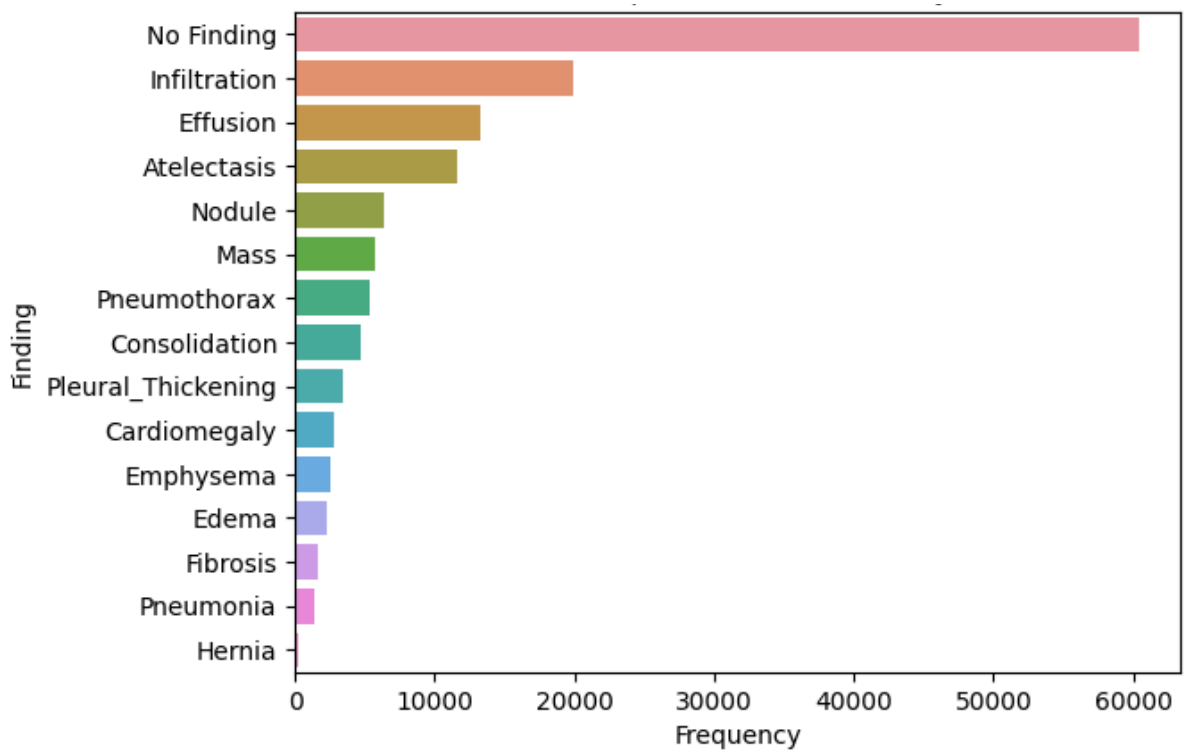


Fig 11: Label Frequencies in ChestXray14

2.2.1.1 Extreme Imbalance Between 'No Finding' and Abnormal Cases

- The ChestXray14 dataset exhibits a severe class imbalance. As illustrated in Fig 11, over 60,000 images (more than half the dataset) are labeled as 'No Finding', which significantly outweighs the presence of any individual disease label. This disproportion skews the training process, making it challenging for models to learn and detect rarer pathologies effectively.

2.2.1.2 Disparity Among Disease Classes

- Beyond the dominance of 'No Finding', the distribution of abnormal findings is also highly uneven. Common conditions such as Infiltration, Effusion, and Atelectasis occur far more frequently than rarer conditions like Hernia, Fibrosis, and Pneumonia (Fig 11). This imbalance can lead to biased models that overfit to common diseases and underperform on clinically critical but rare conditions.

2.2.2 Label Ambiguity and Multi-label Complexity

- ChestXray14 follows a multi-label classification format, where a single chest X-ray may be annotated with multiple disease labels simultaneously (Fig 12). While this reflects real-world clinical scenarios, it adds ambiguity to the learning signal.

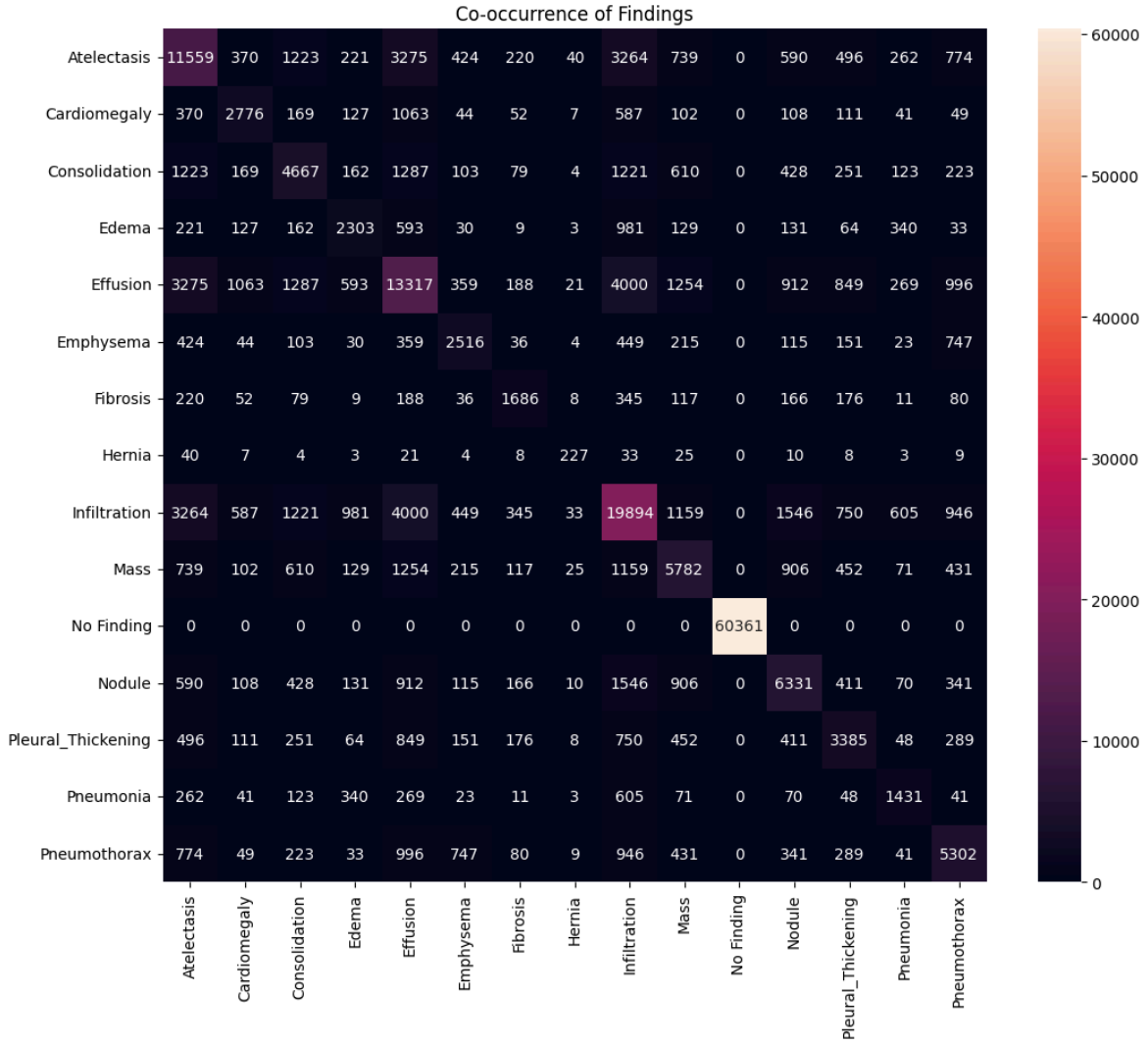


Fig 12: Co-Occurence of Labels in ChestXray14

- The label 'No Finding' does not imply that the image is entirely normal. It simply indicates the absence of the 14 predefined pathologies, meaning the image may still exhibit abnormalities that are either outside the scope of the dataset or are uncertain findings within those categories. This undermines the reliability of the ground-truth labels and may introduce noise into model training.

2.2.3 Scarcity of Localized Annotations

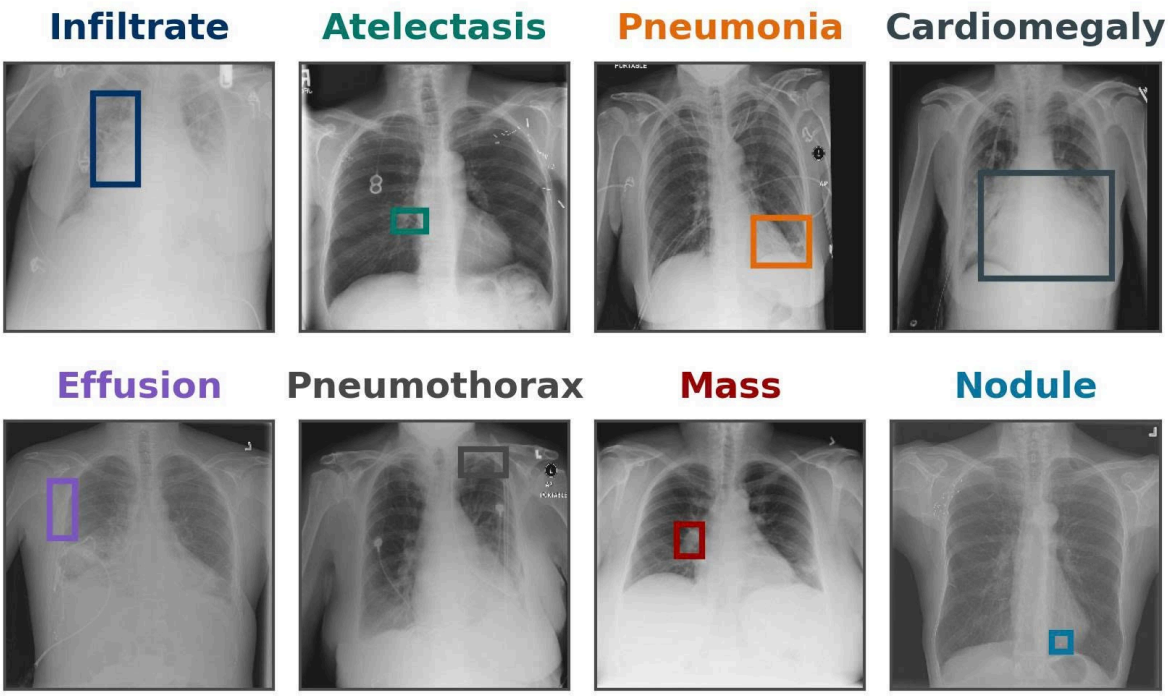


Fig 13: Bounding Box of Diseases Regions in ChestXray14

- A significant limitation of the dataset is the lack of annotated bounding boxes or segmentation masks for most disease categories. Only a small subset of samples includes spatial annotations, which severely limits the use of supervised object localization methods. As a result, models trained on this dataset are often restricted to weakly supervised or attention-based approaches for interpretability and localization.

2.3 Preprocessing Steps

We follow the common preprocessing pipeline used in prior ChestXray14 studies [1], [5], [6], [7], [12], [16], which includes:

- **Image resizing**: All images resized to 224×224 pixels.
- **Normalization**: Normalized using ImageNet statistics.
- **Data Augmentation**: Applied during training to improve generalization:
    - Random rotation (±10°)
    - Horizontal flipping
    - Random cropping and zooming

- **Patient-wise splitting**: Ensured no patient appears in both training and test sets to avoid data leakage, consistent with prior benchmarks [7].

3. Feature Extraction

Our model adopts an **end-to-end feature extraction strategy** using convolutional neural network (CNN) layers enhanced with attention-based weighting blocks (AWBs), allowing the network to automatically learn and refine feature representations from raw chest X-ray images.

We base our approach on the **assumption** that **earlier layers** in the CNN primarily focus on capturing **low-level visual features**, such as edges, textures, and simple shapes, which are common across different regions of the image. In contrast, **deeper layers** are designed to extract more **semantic and class-specific features**, such as the shapes and locations of lesions or patterns indicative of specific thoracic diseases [21]. **By integrating LKA at strategic depths within the network**, we aim to guide the model in **emphasizing diagnostically relevant features** while suppressing irrelevant or redundant information, thereby improving both feature discrimination and overall classification performance. This hierarchical learning process ensures that the model captures both fine-grained local details and high-level semantic patterns critical for accurate multi-label disease prediction.

3.1 Attention-Wise Blocks (AWBs) in DenseNet121 Backbone

In the DenseNet121 backbone, Attention-Wise Blocks (AWBs) are integrated into **Blocks 3 and 4**, which correspond to the deeper stages of the network where higher-level semantic features are learned. These stages are critical for detecting complex pathological patterns in chest X-rays.

By embedding **Large Kernel Attention (LKA)** modules into these blocks, the model gains the ability to capture long-range spatial dependencies through **depthwise dilated convolutions**, while also generating **soft spatial attention masks** that highlight disease-relevant regions. This hierarchical soft segmentation allows the network to suppress irrelevant background and focus on informative regions, acting as an implicit form of feature selection. The placement of LKA in later blocks ensures that attention is applied where high-level semantic understanding is most critical for multi-label classification.

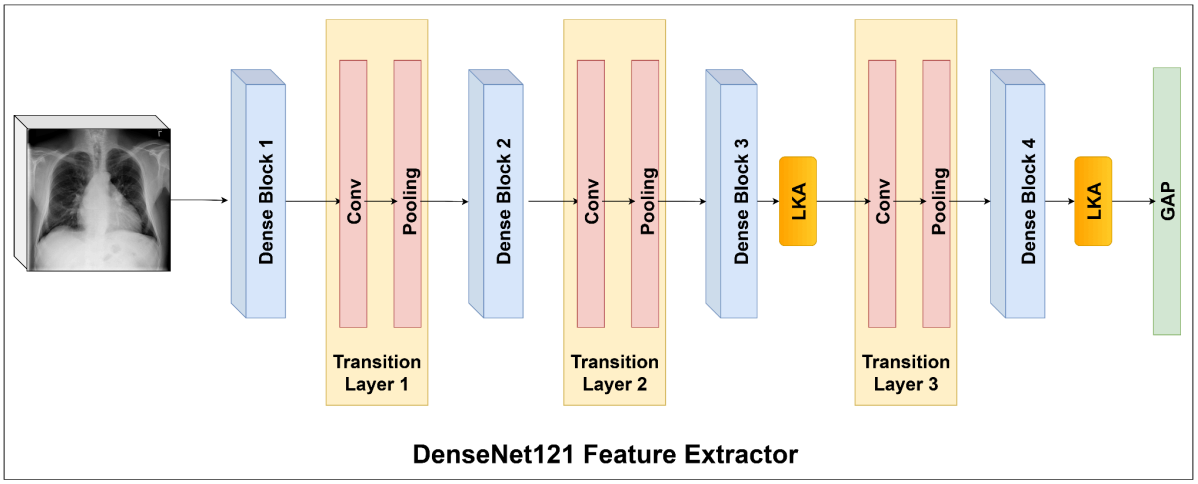


Fig 14: Architecture with AWBs in DenseNet121

3.2 Attention-Wise Blocks (AWBs) in VGG16 Backbone

For the VGG16 backbone, AWBs are inserted into **Blocks 3, 4, and 5**, progressively increasing the network's capacity to refine spatial attention across multiple feature scales. Unlike DenseNet, VGG16 lacks dense connections, making it more reliant on explicit mechanisms like AWBs to improve feature representation. By integrating LKA modules in these mid-to-deep layers, the network enhances its ability to distinguish subtle and co-occurring thoracic abnormalities. The **soft attention masks** generated in each block emphasize diagnostically significant patterns while filtering out noise and redundant signals. This design strengthens the model's discriminative power, especially in complex and imbalanced multi-label scenarios, and leverages the hierarchical nature of CNNs to apply attention at progressively higher semantic levels.

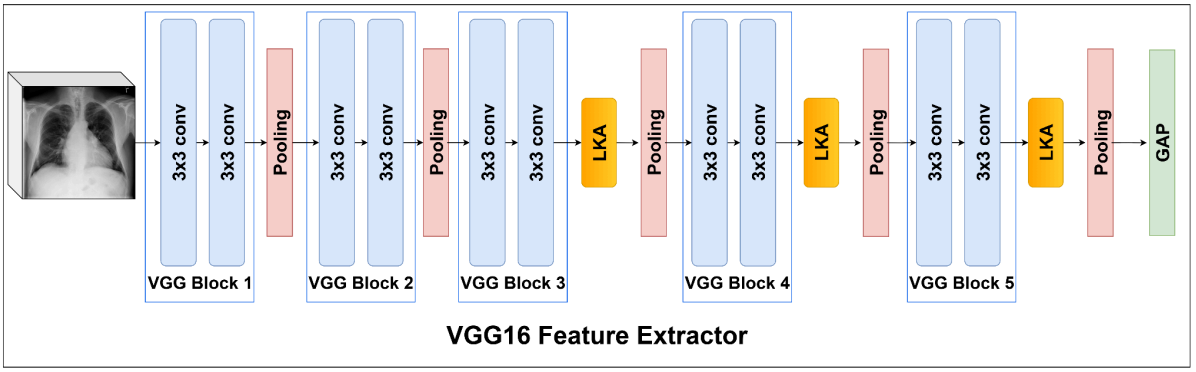


Fig 15: Architecture with AWBs in VGG16

4. Model Training and Validation

4.1 Model Architecture

- **Base Network**: DenseNet121[18], VGG16 [19] with batch normalization
- **Modifications**:
    - Inserted AWBs into blocks 3, and 4 for DenseNet121, blocks 3, 4, and 5 for VGG16
    - Replaced the final classification layer with a sigmoid-activated 14-unit dense layer

4.2 Training Pipeline

We adopted a two-stage training process inspired by [7] to effectively adapt the model to the medical imaging domain while addressing dataset imbalance challenges. In the **first stage**, we trained the model **only on abnormal samples**, excluding "No Finding" cases. This serves as a **warm-up phase**, allowing the model to gradually shift from generic features learned during pretraining on ImageNet-1K to more disease-specific representations. This stage also helps to **prevent early bias** toward the dominant "No Finding" class, which constitutes a large portion of the dataset and can hinder the model's ability to learn meaningful features for rare or subtle pathologies.

In the **second stage**, we fine-tuned the model on the **entire dataset**, including both normal and abnormal samples, to allow it to learn a balanced decision boundary across all classes. This staged approach improves feature discrimination, enhances generalization to real-world clinical data, and ensures that rare disease classes receive sufficient attention during early optimization.

- **Stage 1: Abnormal-Only Training**
    - Training data filtered to exclude "No Finding" images.
    - Binary Cross-Entropy (BCE) loss used to learn features purely from pathological samples.

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{N_c}\left[y_{i,c}\log(\sigma(z_{i,c})) + (1-y_{i,c})\log(1-\sigma(z_{i,c}))\right]$$

    - Goal: help the network learn disease-relevant patterns without class imbalance interference.

- **Stage 2: Full Dataset Fine-Tuning**
    - Reintroduced full dataset (including ‘No Finding' cases).
    - Loss switched to Focal Loss [15] to mitigate class imbalance by down-weighting easy examples and emphasizing hard-to-classify samples.

$$L_{FL} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{N_c}\alpha_c(1-p_{i,c})^{\gamma}y_{i,c}\log(p_{i,c})$$

- **Suppress easy examples**: When a model correctly predicts a majority class with high confidence (i.e., $p_{i,c}$ is close to 1), the term $(1-p_{i,c})^{\gamma}$ becomes small, reducing the loss contribution from those examples.
- **Emphasizes hard/misclassified examples**: When a minority class is misclassified (i.e., $p_{i,c}$ is low), this term is larger, increasing the loss and hence the learning signal.
- **Class-level reweighting** with α_c allows additional manual balancing for rare disease labels.

4.3 Train and Test Strategy

We perform a patient-wise split to prevent information leakage, following the protocol of prior works [7]. The dataset is divided as follows:

- 70% for training
- 10% for validation
- 20% for testing

Stage	Training Set	Validation Set	Test Set
Stage 1	36,506 images	5,215 images	-
Stage 2	78,544 images	11,220 images	22,356 images

Table 2: Splitting Dataset

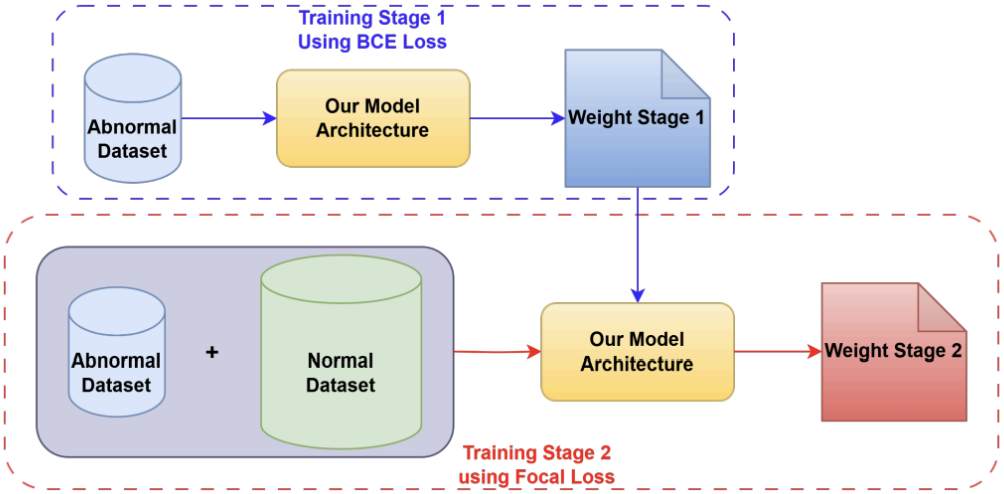


Fig 16: Two-Stage Training Strategy

5. Evaluation Metrics

To comprehensively evaluate performance on multi-label classification with class imbalance, we used the following metrics:

- **Area Under the Curve (AUC)**: Calculated per class and averaged, used in [1], [5], [7], [16].
- **ROC Curve**: To visualize sensitivity/specificity trade-offs, used in [1], [7], [16].
- **Grad-CAM Visualizations [20]**: For interpretability of attention maps.

6. Implementation Plan

6.1 Development Tools

- **Language**: Python 3.x
- **Frameworks**: PyTorch, NumPy, OpenCV, Scikit-learn
- **Visualization**: Matplotlib, Seaborn
- **Version Control**: GitHub
- **Platform**: Kaggle (NVIDIA P100 GPU)

6.2 Implementation Steps

1. Load and preprocess ChestXray14 dataset
2. Implement AWB modules using LKA design
3. Modify DenseNet121/ VGG16 to include AWBs at strategic locations
4. Train model in two phases using BCE and Focal Loss
5. Evaluate and analyze performance
6. Visualize attention using Grad-CAM
7. Compare with baselines (CheXNet, SynthEnsemble, AG-CNN)

Code repo: *https://github.com/NNNguyenDuyyy/DSP391m_GROUP4*

7. Ethical Considerations

- **Data Privacy**: ChestXray14 is a publicly available, de-identified dataset with no personally identifiable information.
- **Bias and Fairness**: Efforts were made to reduce dataset bias by using balanced training and fair evaluation. However, demographic attributes (e.g., age, sex) were not used in the analysis.
- **Clinical Use**: This model is designed for research purposes only. It is not FDA-approved and should not be deployed for real-world diagnosis without further validation.
- **Transparency**: All code and model checkpoints are publicly available to encourage reproducibility and transparency.

V. System Design and Implementation

1. AI Model Integration

AWBs are embedded into CNN at architectural level. Input X-ray → CNN + AWB → Multi-label Output

2. Data Flow and Processing

Image → Preprocess → CNN Backbone → AWBs → Prediction (14 disease labels)

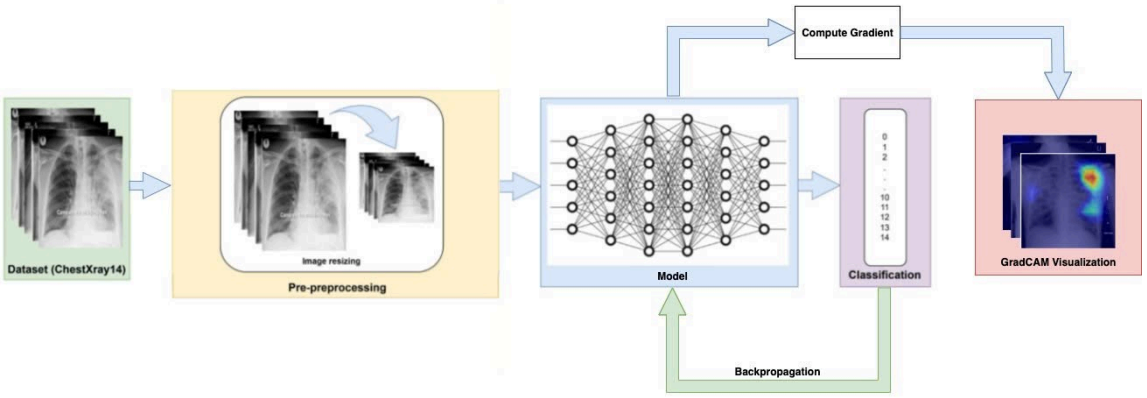


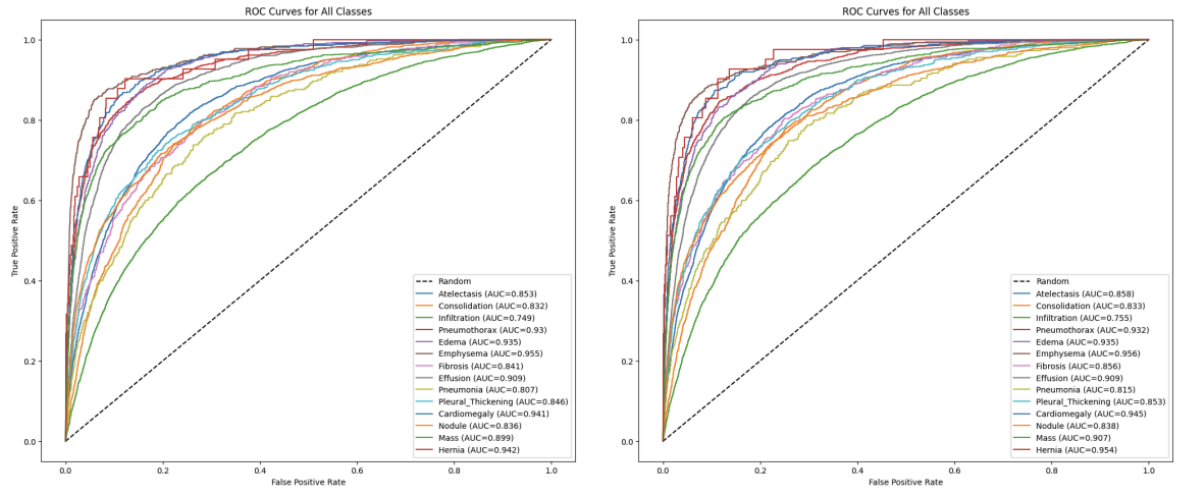
Fig 17: System Workflow

VI. Results and Discussion

1. Results and Analysis

We evaluated our proposed Attention-Wise Block (AWB)-based VGG16 architecture on the **ChestXray14** dataset, which contains over 100,000 frontal-view X-ray images labeled with 14 thoracic disease categories. We adopted the standard patient-wise data split to avoid label leakage and ensure clinically relevant evaluation.

The **main evaluation metric** used was the Area Under the Receiver Operating Characteristic Curve (**AUC**), which is widely accepted in medical image analysis for imbalanced, multi-label classification.



(a) Ours (BCELoss 2 Stage). (b) Ours (BCE + FocalLoss).

Fig 18: ROC AUC

1.1 Overall Performance

Pathology	Wang et al.[1]	Rajpurkar et al.[5]	Ashraf et al.[7]	Guan et al.(1)[16]	Guan et al.(2)[16]	Ours(1)	Ours(2)
Atelectasis	0.716	0.8094	0.83390	0.844	0.853	0.853	**0.858**
Cardiomegaly	0.807	0.9248	0.91954	0.937	0.939	0.941	**0.945**
Effusion	0.784	0.8638	0.88977	0.904	0.903	**0.909**	**0.909**
Infiltration	0.609	0.7345	0.74102	0.753	0.754	0.749	**0.755**
Mass	0.706	0.8676	0.87315	0.893	0.902	0.899	**0.907**
Nodule	0.671	0.7802	0.80611	0.827	0.828	0.836	**0.838**
Pneumonia	0.633	0.7680	0.77648	0.776	0.774	0.807	**0.815**
Pneumothorax	0.806	0.8887	0.90164	0.919	0.921	0.930	**0.932**
Consolidation	0.708	0.7901	0.81575	**0.842**	**0.842**	0.832	0.833
Edema	0.835	0.8878	0.91034	0.919	0.924	**0.935**	**0.935**
Emphysema	0.815	0.9371	0.92946	0.941	0.932	0.955	**0.956**
Fibrosis	0.769	0.8047	0.83347	0.857	**0.864**	0.841	0.856
Pleural Thickening	0.708	0.8062	0.81270	0.836	0.837	0.846	**0.853**
Hernia	0.767	0.9164	0.91723	0.903	0.921	0.942	**0.954**
Mean AUC	0.738	0.841	0.85433	0.868	0.871	0.8768	**0.8818**

Table 3: Our Results Compared to Previous Studies

Ours				
Model variant	**Training stage 1**	**Training stage 2**	**Total**	**GFLOPS**
Densenet121+LKA full	6563.2s	8588.8s	15152s	3.64
Densenet121+LKA 2 block	6375.6s	8239s	14614.6	3.14
Densenet121+LKA after	5611.4s	6957.3s	12568.7s	2.92
VGG16+LKA full	14225.2s	10206.1s	24431.3s	16.70
VGG16+LKA 3 block	8628.6s	5682.1s	14310.7s	15.93
VGG16+LKA after	4851.2s	5492.6s	10343.8s	15.39

Ours		
Model variant	**Trainable params**	**Total params**
Densenet121+lka full	9609870	9609870
Densenet121+lka 2 block	9223054	9223054
Densenet121+lka after	8095630	8096530
VGG16+lka full	15453966	15453966
VGG16+lka 3 block	15418702	15418702
VGG16+lka after	15031886	15031886

Synth Ensemble				
Model	**Trainable Params**	**Total Params**	**FLOPS (GFLOPS)**	**Training time**
CoAtNet	1128320	73904264	14.55	Stage 1: 21045.6s Stage 2: 6811.2s
DenseNet121	1144512	8014720	2.87	Stage 1: 40958.1s
MaxViTV2	1122944	116128376	23.88	Stage 1: 13791.6s
SwinV2	842240	49725140	9.08	Stage 1: 31071.3s
VOLO D2	319872	57923696	14.24	Stage 1: 32398.5s
convnextv2	1647488	198007040	34.4	Stage 1: 35538.9 Stage 2: 11597.6s
Total	6205376	503703236	99.02	~193212.8s

Table 4: Our Parameters and Training Time Compared to Synth Ensemble

Our model **achieved a new state-of-the-art mean AUC of 0.8818**, outperforming both earlier CNN-based baselines (e.g., CheXNet, DenseNet121) and hybrid transformer models.

1.2 Per-Class AUC Improvements

We observed substantial improvements on underrepresented diseases, which are typically difficult to detect in imbalanced datasets:

- **Hernia**: 0.954 (previous best: 0.921)
- **Edema**: 0.935 (previous best: 0.924)
- **Pneumonia**: 0.815 (previous best: 0.776)

These gains demonstrate the ability of our architecture to **enhance feature learning on rare pathologies** via attention-guided modules and training rebalancing.

1.3 Ablation Study Results

Model Variant	One-Stage BCE	One-Stage Focal	Two-Stage
DenseNet121	0.8227	0.8044	0.8402
DenseNet121 + AWBs	0.8340	0.8369	**0.8498**
VGG16	0.8275	0.8094	0.8401
VGG16 + AWBs	0.8238	0.8247	**0.8818**

Table 5: Impact of Training Strategy on Mean AUC

Model	Position	Mean AUC
DenseNet121 + LKA	After	0.8475
DenseNet121 + LKA	Full Block	0.8416
DenseNet121 + LKA	Last 2 Block	**0.8498**
VGG16 + LKA	After	0.8470
VGG16 + LKA	Full Block	0.8666
VGG16 + LKA	Last 3 Block	**0.8818**

Table 6: Impact of AWBs placement on Mean AUC

These results confirm that **combining a two-stage training pipeline with AWB placement in deeper layers of the CNN significantly improves performance**, especially compared to one-stage.

2. Discussion

2.1 The experimental results validate our two key hypotheses:

1. **Integrating attention mechanisms enhances feature discriminability**, especially for subtle and spatially diffuse abnormalities in CXR images.

2. **Addressing class imbalance through a two-stage training strategy**—first training on abnormal samples and later rebalancing with Focal Loss—enables better generalization, particularly for rare classes.

2.2 Why Our Method Performs Better

- **AWBs improve both spatial and channel-wise attention**, which helps the model identify not just what features are important but also where they occur.

- The **hierarchical placement** of AWBs in blocks 3–5 aligns with semantic depth in CNNs, boosting high-level decision-making.

- **Focal Loss** in the second training stage helps the model avoid being overwhelmed by the "No Finding" majority class, which often dominates learning in medical datasets.

2.3 Comparison with Prior Work

- **CheXNet** used DenseNet121 but lacked attention mechanisms and imbalance handling—limiting sensitivity to rarer conditions.

- **SynthEnsemble** introduced hybrid ensembles but did not offer as lightweight or interpretable a model.

- **Transformer-based methods** such as SwinChex and MedViT require more data and computation, whereas our approach achieves competitive (and better) performance using just VGG16 + attention modules.

2.4 Visualization Insights

Grad-CAM visualizations show that AWBs lead to **sharper and more anatomically consistent activations**. Our LKA-inspired attention maps localize disease-relevant areas more precisely, acting as an implicit soft segmentation mechanism.

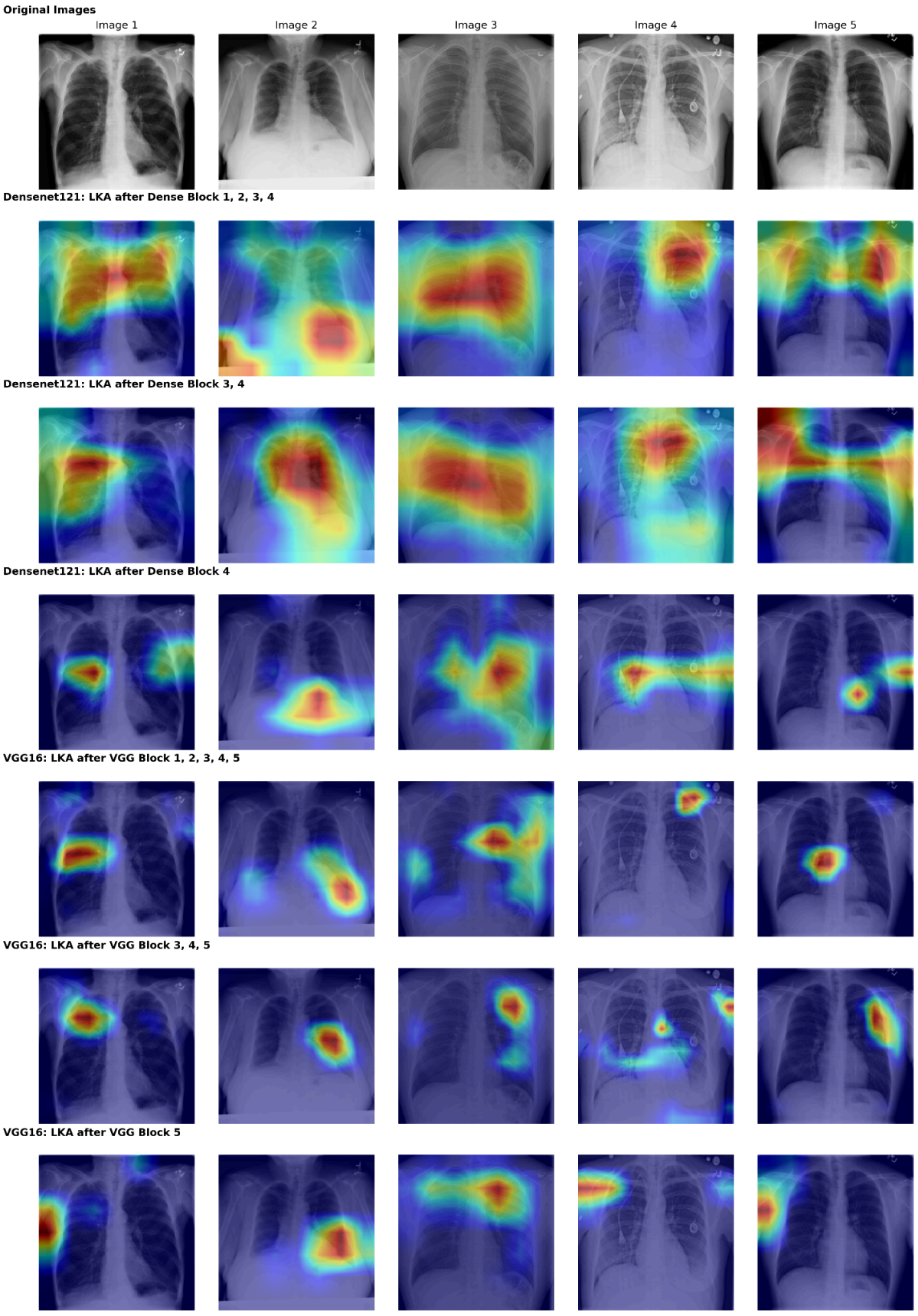


Fig 19: Grad-CAM Visualizations

3. Recommendations

Based on our results, we recommend the following for future AI-based medical imaging systems:

- Combine **architecture-level attention** with **loss-level imbalance handling** for better generalization in multi-label tasks.
- Prioritize **lightweight attention modules** like LKA for resource-constrained environments, e.g., in developing countries or rural clinics.
- Use **two-stage or curriculum-based training strategies** for datasets with extreme label imbalance.

VII. Conclusion

1. Summary of Findings

This project set out to address two significant challenges in chest X-ray diagnosis using deep learning: the difficulty of detecting co-occurring thoracic diseases and the severe class imbalance present in large medical datasets such as ChestXray14.

To tackle these issues, we proposed a novel framework that integrates **Attention-Wise Blocks (AWBs)**—based on the Large Kernel Attention (LKA) mechanism—into a **DenseNet121** and **VGG16 backbone**, combined with a **two-stage training pipeline** that isolates abnormal cases in early training and uses **Focal Loss** for fine-tuning.

Key findings include:

- Our method achieved a **mean AUC of 0.8818**, surpassing state-of-the-art models including CheXNet, SynthEnsemble, and DenseNet121.
- It performed especially well on **rare pathologies** like Hernia (AUC: 0.954), Pneumonia (AUC: 0.815), and Nodule (AUC: 0.838), validating the effectiveness of our approach on underrepresented classes.
- Grad-CAM visualizations confirmed that AWBs enabled the model to focus more precisely on disease-relevant anatomical regions.
- Ablation studies showed that both the **strategic module placement** and **two-phase curriculum training** contributed significantly to overall performance improvements.

These results strongly support the hypothesis that integrating attention mechanisms and tailored training strategies can dramatically enhance performance on complex, real-world medical image classification tasks.

2. Contributions and Reflections on the Project

2.1 Contributions

This capstone project contributes to the field in several meaningful ways:

- **Architectural Contribution**: Introduced AWBs as lightweight, interpretable modules that can be easily plugged into existing CNNs to enhance their spatial reasoning capabilities.

- **Training Strategy**: Designed a novel two-stage training procedure that balances representation learning and class imbalance handling, leading to significantly improved detection of rare conditions.

- **Open Access**: All source code and model weights are released publicly, promoting transparency, reproducibility, and potential clinical collaboration in the future.

- **Evaluation and Insight**: Conducted rigorous experiments, ablation studies, and visualizations to demonstrate the effectiveness and practicality of the proposed approach.

2.2 Reflections

Throughout this project, we learned to approach deep learning not only as a performance optimization problem but also as a **practical, interpretable, and ethical tool** in high-stakes domains like healthcare. We recognized that minor architectural innovations, when paired with thoughtful training design, can have a significant impact—even when using legacy architectures like VGG16.

We also encountered and overcame challenges related to computational constraints, data imbalance, and model generalization. These experiences enriched our understanding of deep learning workflows in applied AI projects and strengthened our collaborative and research skills.

3. Limitations and Future Work

3.1 Limitations

While our project achieved strong results, some limitations remain:

- **Dataset Boundaries**: Evaluation was conducted on ChestXray14 only. The model has not yet been tested on other datasets like CheXpert or MIMIC-CXR, which may vary in resolution, label quality, and demographic composition.

- **Lack of Clinical Validation**: Although Grad-CAM helped improve interpretability, we did not conduct formal clinical validation with radiologists to assess diagnostic trustworthiness.

- **Fixed Backbone**: Our experiments focused on VGG16. It is unclear how AWBs will generalize across modern architectures like EfficientNet or ResNet variants.

- **No Pathology Dependency Modeling**: The model does not explicitly consider inter-label dependencies (e.g., Infiltration and Effusion frequently co-occur).

3.2 Future Work

To build upon this project, we propose the following directions:

- **Cross-Dataset Evaluation**: Test the AWB-enhanced models on other public datasets to evaluate generalization and robustness.

- **Multimodal Learning**: Incorporate radiology reports using vision-language models for richer representations.

- **Graph-based Label Modeling**: Integrate label co-occurrence graphs or multi-task frameworks to improve understanding of pathological dependencies.

- **Clinical Collaboration**: Partner with hospitals to deploy the model in a low-risk diagnostic assistant setting and gather clinician feedback.

- **AutoML Exploration**: Use neural architecture search (NAS) to automate and optimize AWB placement across different CNN backbones.

VIII. References

1. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2097–2106 (2017).

2. Johnson, A.E., Pollard, T.J., Berkowitz, S.J., Greenbaum, N.R., Lungren, M.P., Deng, C.y., Mark, R.G., Horng, S.: Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. Scientific data 6(1), 317 (2019).

3. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al.: Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 590–597 (2019).

4. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., S´anchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis 42, 60–88 (2017).

5. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 (2017).

6. Yao, L., Poblenz, E., Dagunts, D., Covington, B., Bernard, D., Lyman, K.: Learning to diagnose from scratch by exploiting dependencies among labels. arXiv preprint arXiv:1710.10501 (2017).

7. Ashraf, S.N., Mamun, M.A., Abdullah, H.M., Alam, M.G.R.: Synthensemble: a fusion of cnn, vision transformer, and hybrid models for multi-label chest x-ray classification. In: 2023 26th International Conference on Computer and Information Technology (ICCIT). pp. 1–6. IEEE (2023).

8. Taslimi, S., Taslimi, S., Fathi, N., Salehi, M., Rohban, M.H.: Swinchex: Multilabel classification on chest x-ray images with transformers. arXiv preprint arXiv:2206.04246 (2022).

9. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021).

10. Manzari, O.N., Ahmadabadi, H., Kashiani, H., Shokouhi, S.B., Ayatollahi, A.: Medvit: a robust vision transformer for generalized medical image classification. Computers in biology and medicine 157, 106791 (2023).

11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).

12. Yan, C., Yao, J., Li, R., Xu, Z., Huang, J.: Weakly supervised deep learning for thoracic disease classification and localization on chest x-rays. In: Proceedings of the 2018 ACM

international conference on bioinformatics, computational biology, and health informatics. pp. 103–110 (2018).

13. Yan, Y., Kawahara, J., Hamarneh, G.: Melanoma recognition via visual attention. In: Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26. pp. 793–804. Springer (2019).

14. Guo, M.H., Lu, C.Z., Liu, Z.N., Cheng, M.M., Hu, S.M.: Visual attention network. Computational visual media 9(4), 733–752 (2023).

15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Doll´ar, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017).

16. Guan, Q., Huang, Y., Zhong, Z., Zheng, Z., Zheng, L., Yang, Y.: Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. arXiv preprint arXiv:1801.09927 (2018).

17. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018).

18. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

20. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Gradcam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017).

21. Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Cham: Springer International Publishing.

IX. Appendices

1. LKA in code

```python
class LKA(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.dw_conv1 = nn.Conv2d(dim, dim, kernel_size=5, padding=2, groups=dim)
        self.dw_conv2 = nn.Conv2d(dim, dim, kernel_size=7, padding=9, dilation=3, groups=dim)
        self.pw_conv = nn.Conv2d(dim, dim, kernel_size=1)

    def forward(self, x):
        u = x
        x = self.dw_conv1(x)
        x = self.dw_conv2(x)
        x = self.pw_conv(x)
        return x * u  # Attention applied element-wise
```

2. How we integrate LKA into CNN backbones in code

2.1 DenseNet121 + LKA Full Block

```python
class DenseNet121_fullLKA(nn.Module):
    def __init__(self, num_classes=1000, dropout=0.5):
        super(DenseNet121_fullLKA, self).__init__()
        densenet = models.densenet121(pretrained=True)
        self.features = densenet.features

        # Apply attention only in deeper blocks to save FLOPs
        self.att_block1 = LKA(256)
        self.att_block2 = LKA(512)
        self.att_block3 = LKA(1024)
        self.att_block4 = LKA(1024)

        self.avgpool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout)
        self.classifier = nn.Linear(1024, num_classes)
```

```python
def forward(self, x):
    x = self.features.conv0(x)
    x = self.features.norm0(x)
    x = self.features.relu0(x)
    x = self.features.pool0(x)

    x = self.features.denseblock1(x)
    x = self.att_block1(x)
    x = self.features.transition1(x)

    x = self.features.denseblock2(x)
    x = self.att_block2(x)
    x = self.features.transition2(x)

    x = self.features.denseblock3(x)
    x = self.att_block3(x)
    x = self.features.transition3(x)

    x = self.features.denseblock4(x)
    x = self.att_block4(x)

    x = self.features.norm5(x)
    x = self.avgpool(x)
    x = torch.flatten(x, 1)
    x = self.dropout(x)
    x = self.classifier(x)
    return x
```

2.2 DenseNet121 + LKA 2 Block

```python
class DenseNet121_fullLKA(nn.Module):
    def __init__(self, num_classes=1000, dropout=0.5):
        super(DenseNet121_fullLKA, self).__init__()
        densenet = models.densenet121(pretrained=True)
        self.features = densenet.features

        # Apply attention only in deeper blocks to save FLOPs
        #self.att_block1 = LKA(256)
        #self.att_block2 = LKA(512)
        self.att_block3 = LKA(1024)
        self.att_block4 = LKA(1024)

        self.avgpool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout)
        self.classifier = nn.Linear(1024, num_classes)
```

```python
def forward(self, x):
    x = self.features.conv0(x)
    x = self.features.norm0(x)
    x = self.features.relu0(x)
    x = self.features.pool0(x)

    x = self.features.denseblock1(x)
    #x = self.att_block1(x)
    x = self.features.transition1(x)

    x = self.features.denseblock2(x)
    #x = self.att_block2(x)
    x = self.features.transition2(x)

    x = self.features.denseblock3(x)
    x = self.att_block3(x)
    x = self.features.transition3(x)

    x = self.features.denseblock4(x)
    x = self.att_block4(x)

    x = self.features.norm5(x)
    x = self.avgpool(x)
    x = torch.flatten(x, 1)
    x = self.dropout(x)
    x = self.classifier(x)
    return x
```

2.3 DenseNet121 + LKA After

```python
class densenet_LKA_after(nn.Module):
    def __init__(self, num_classes, dropout=0.5):
        super(densenet_LKA_after, self).__init__()
        base = models.densenet121(pretrained=True)

        # Full backbone (keep as-is)
        self.features = base.features

        # Add LKA after norm5
        self.lka = LKA(dim=1024)

        # Classification head
        self.pool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout) if dropout else nn.Identity()
        self.classifier = nn.Linear(1024, num_classes)

    def forward(self, x):
        x = self.features(x)      # DenseNet121 full features (includes norm5)
        x = self.lka(x)           # LKA-enhanced final feature map
        x = self.pool(x).flatten(1)
        x = self.dropout(x)
        x = self.classifier(x)
        return x
```

2.4 VGG16 + LKA Full Block

```python
class VGG16_LKA(nn.Module):
    def __init__(self, num_classes, dropout=None):
        super().__init__()
        base = models.vgg16_bn(pretrained=True)

        self.block1 = nn.Sequential(*base.features[0:6])    # 64
        self.block2 = nn.Sequential(*base.features[7:13])   # 128
        self.block3 = nn.Sequential(*base.features[14:23])  # 256
        self.block4 = nn.Sequential(*base.features[24:33])  # 512
        self.block5 = nn.Sequential(*base.features[34:43])  # 512

        # Insert LKA modules after block3, block4, block5
        self.lka1 = LKA(dim=64)
        self.lka2 = LKA(dim=128)
        self.lka3 = LKA(dim=256)
        self.lka4 = LKA(dim=512)
        self.lka5 = LKA(dim=512)

        self.pool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout) if dropout else nn.Identity()
        self.classifier = nn.Linear(512, num_classes)
```

```python
def forward(self, x):
    x = self.block1(x)
    x = self.lka1(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block2(x)
    x = self.lka2(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block3(x)
    x = self.lka3(x)  # Attention-enhanced features
    x = F.max_pool2d(x, 2, 2)

    x = self.block4(x)
    x = self.lka4(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block5(x)
    x = self.lka5(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.pool(x).flatten(1)
    x = self.dropout(x)
    x = self.classifier(x)
    return x
```

2.5 VGG16 + LKA 2 Block

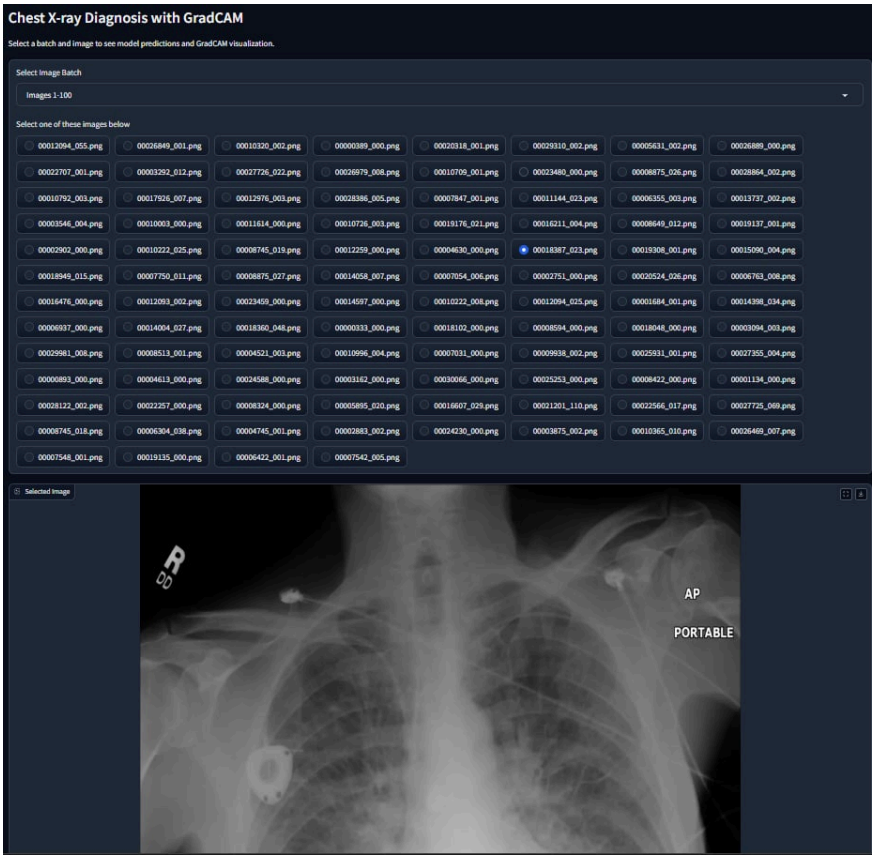
```python
class VGG16_LKA(nn.Module):
    def __init__(self, num_classes, dropout=None):
        super().__init__()
        base = models.vgg16_bn(pretrained=True)

        self.block1 = nn.Sequential(*base.features[0:6])    # 64
        self.block2 = nn.Sequential(*base.features[7:13])   # 128
        self.block3 = nn.Sequential(*base.features[14:23])  # 256
        self.block4 = nn.Sequential(*base.features[24:33])  # 512
        self.block5 = nn.Sequential(*base.features[34:43])  # 512

        # Insert LKA modules after block3, block4, block5
        self.lka3 = LKA(dim=256)
        self.lka4 = LKA(dim=512)
        self.lka5 = LKA(dim=512)

        self.pool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout) if dropout else nn.Identity()
        self.classifier = nn.Linear(512, num_classes)
```

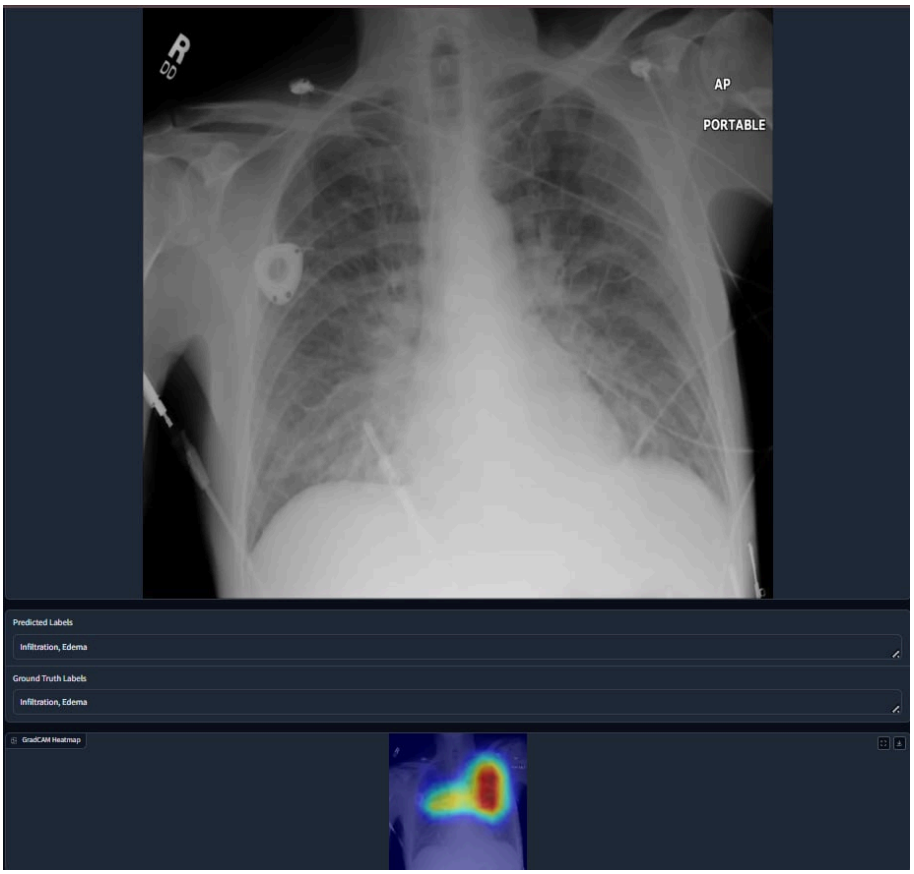
```python
def forward(self, x):
    x = self.block1(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block2(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block3(x)
    x = self.lka3(x)  # Attention-enhanced features
    x = F.max_pool2d(x, 2, 2)

    x = self.block4(x)
    x = self.lka4(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.block5(x)
    x = self.lka5(x)
    x = F.max_pool2d(x, 2, 2)

    x = self.pool(x).flatten(1)
    x = self.dropout(x)
    x = self.classifier(x)
    return x
```

2.6 VGG16 + LKA After

```python
class VGG_LKA_after(nn.Module):
    def __init__(self, num_classes, dropout=0.5):
        super(VGG_LKA_after, self).__init__()
        base = models.vgg16_bn(pretrained=True)

        # Full backbone (keep as-is)
        self.features = base.features

        # Add LKA after norm5
        self.lka = LKA(dim=512)

        # Classification head
        self.pool = nn.AdaptiveAvgPool2d((1, 1))
        self.dropout = nn.Dropout(dropout) if dropout else nn.Identity()
        self.classifier = nn.Linear(512, num_classes)

    def forward(self, x):
        x = self.features(x)      # DenseNet121 full features (includes norm5)
        x = self.lka(x)           # LKA-enhanced final feature map
        x = self.pool(x).flatten(1)
        x = self.dropout(x)
        x = self.classifier(x)
        return x
```

2.7 Demo UI




R
AP
PORTABLE
Predicted Labels
Infiltration, Edema
Ground Truth Labels
Infiltration, Edema
GradCAM Heatmap

- THE END -