Automated Pneumonia Diagnosis Using CNNs on Radiographic Images

Sameera Aluri
Department of Computer and Information Sciences
Fordham University
New York, NY

***Abstract*—Pneumonia is a major global health concern, particularly among children under five, where it accounts for over 700,000 deaths annually. Accurate and timely diagnosis using chest X-rays is critical but challenging in resource-limited settings due to the scarcity of trained radiologists. In this project, we present a deep learning-based pipeline to automatically detect pneumonia in pediatric chest radiographs. We use a convolutional neural network (CNN) with transfer learning via a fine-tuned ResNet50 architecture, combined with a novel preprocessing step: fixed lung-region cropping to focus model attention on relevant anatomy. To address class imbalance, we employ class weighting and binary focal loss. Model interpretability is achieved through Grad-CAM, enabling visual explanations of predictions. Our best model achieved 86% accuracy, 0.96 AUC, and improved normal class recall from 15% to 98% after lung cropping. These results highlight the potential of explainable AI in assisting clinical diagnosis and improving reliability in medical imaging tasks.**

***Index Terms*—Pneumonia Detection, Pediatric Chest X-rays, Convolutional Neural Networks, Deep Learning, ResNet50, Transfer Learning, Lung Cropping, Grad-CAM, Explainable AI, Medical Image Analysis.**

I. Introduction

Pneumonia is one of the most common causes of death among children under five years old, particularly in developing countries where access to skilled healthcare providers is limited. Early detection is vital, as delays in diagnosis significantly increase the risk of complications or death. Chest X-rays (CXRs) are the gold standard for diagnosing pneumonia, but their interpretation requires experienced radiologists and is prone to inter-observer variability.

With the rise of deep learning and medical imaging AI, automated solutions for pneumonia detection have shown promise. Convolutional Neural Networks (CNNs), especially when combined with transfer learning, can achieve expert-level performance on CXR datasets. However, many of these models suffer from overfitting, poor generalization, and limited interpretability, particularly when trained on adult-focused datasets.

This project focuses on developing an explainable deep learning model for binary classification of pediatric chest X-rays into *Normal* and *Pneumonia* classes. We utilize transfer learning with a pretrained ResNet50 model and introduce a preprocessing technique, fixed lung cropping, to restrict the model's attention to clinically relevant regions. In addition, we address class imbalance through weighted loss functions and visualize model decision-making using Grad-CAM.

Our work demonstrates that targeted preprocessing, interpretability, and training strategies can significantly improve model performance and reliability, especially in medical settings where trust and precision are essential.

II. Related Work

A. CNN-Based Pneumonia Detection

Deep learning, particularly convolutional neural networks (CNNs), has revolutionized medical image classification. A foundational work by Rajpurkar et al. introduced CheXNet, a 121-layer DenseNet trained on the NIH ChestXray14 dataset, achieving an AUC of 0.915 for pneumonia detection, which is on par with radiologists [1]. This study set the benchmark for automated pneumonia diagnosis using chest radiographs.

While such models demonstrate high performance on large datasets, many are trained on adult X-rays and show limited generalization to pediatric populations due to anatomical differences and label inconsistencies.

B. Transfer Learning in Medical Imaging

Given the limited size of many clinical datasets, transfer learning has emerged as a standard strategy. Pretrained networks such as ResNet, VGG, and EfficientNet are commonly fine-tuned on medical imaging tasks to leverage learned feature representations. Kermany et al. [2] demonstrated the effectiveness of this approach by fine-tuning pretrained CNNs on a pediatric chest X-ray dataset, achieving diagnostic performance comparable to medical experts.

Transfer learning helps the model learn faster and reduces the chances of overfitting, which is especially helpful when working with grayscale images and a small variety of classes.

C. Custom Architectures and Ensembles

Beyond standard CNNs, several researchers have proposed custom networks optimized for pneumonia detection. Alsharif et al. [3] developed PneumoniaNet, a pediatric-specific CNN achieving $\tilde{9}$9.7% accuracy. Recent works also explore ensemble methods, combining networks like DenseNet and EfficientNet with attention mechanisms to boost performance and robustness [4].

However, these methods often require extensive computational resources and suffer from interpretability issues, which limit clinical adoption.

D. Pediatric Data Considerations

Adult datasets such as ChestXray14 and CheXpert are used throughout the literature, yet they show limitations when applied to pediatric patients. Studies show that models trained on adult data generalize poorly to pediatric cases, with some experiencing AUC drops from 0̃.90 to 0̃.54 on external pediatric datasets [5].

This highlights the importance of pediatric-specific models, which better capture the unique radiographic patterns in children's lungs.

E. Explainability and Grad-CAM

To promote trust in clinical AI, explainability techniques like Grad-CAM (Gradient-weighted Class Activation Mapping) [6] are widely used. Grad-CAM generates heatmaps indicating which parts of an image most influenced a model's prediction. While many studies incorporate Grad-CAM, few evaluate its reliability or integrate preprocessing to enhance its effectiveness.

F. Common Challenges

Despite promising results, several key challenges persist:

- **Generalization**: Many models overfit to internal validation sets and fail on unseen data.
- **Class Imbalance**: Pneumonia cases often outnumber normal cases, biasing the model.
- **Spurious Attention**: CNNs may fixate on irrelevant regions (e.g., edges, labels, tubes) instead of lung tissue.
- **Interpretability**: The "black-box" nature of CNNs raises skepticism in clinical deployment.

This project directly addresses these concerns by focusing on pediatric-specific data, applying lung-region cropping to minimize spurious attention, using focal loss to address imbalance, and integrating Grad-CAM for explainability.

III. Methods

This section outlines the end-to-end pipeline developed for automated pneumonia detection from pediatric chest X-rays, including data preprocessing, model architecture, training procedures, and interpretability techniques.

A. Dataset Description

We used the publicly available pediatric chest X-ray dataset released by Mooney on Kaggle, originally sourced from the Guangzhou Women and Children's Medical Center. The dataset contains 5,863 labeled chest radiographs in JPEG format, divided into two classes:

- **Pneumonia** (label = 1): 4,273 images
- **Normal** (label = 0): 1,583 images

The dataset is already organized into three subsets:

- **Training set**: 5̃,200 images
- **Test set**: 624 images
- **Validation set**: Custom-created (20% of training data)

All images are grayscale X-rays and vary in size and resolution. No patient metadata is provided, making this a purely image-based classification task.

B. Preprocessing and Data Augmentation

To standardize input to the neural network and guide model focus, the following preprocessing steps were implemented:

*a) **Lung Region Cropping**:* Initial experiments revealed that the model often attended to irrelevant regions (e.g., spine, neck, image corners), especially for *Normal* cases. To address this, we applied fixed-region cropping to isolate the central lung region in all images. This was implemented using custom logic that:

- Crops a rectangle from 15% to 90% height and 20% to 80% width
- Resizes the cropped image to 224×224 pixels
- Converts grayscale input to RGB by duplicating channels

This cropping reduced distractions and improved generalization.

*b) **Data Augmentation**:* For the training set, we applied the following augmentation techniques to improve robustness:

- Random rotation (±20 degrees)
- Zooming (up to 20%)
- Width and height shifts (up to 20%)
- Shearing (up to 10%)
- Horizontal flipping
- Pixel normalization (rescaling to [0, 1])

No augmentation was applied to validation or test sets to ensure consistent evaluation.

C. Model Architecture

We used a transfer learning approach based on ResNet50, a 50-layer deep residual network pretrained on ImageNet. The model was modified as follows:

- **Backbone**: ResNet50 with `include_top=False` to exclude classification head
- **Trainable Layers**: Last 50 layers unfrozen for fine-tuning
- **Input Shape**: (224, 224, 3) to match preprocessed image size
- **Custom Head**:
  - `GlobalAveragePooling2D`
  - `Dropout(0.5)`
  - `Dense(128, ReLU)`
  - `Dropout(0.3)`
  - `Dense(1, Sigmoid)` for binary output

The total number of trainable parameters was approximately 17.2 million, and non-trainable parameters totaled around 6.6 million.

D. Loss Function and Optimization

To address class imbalance where pneumonia cases outnumber normal ones by 2̃.7:1, we used the following strategies:

- **Loss Function**: `BinaryFocalCrossentropy` with γ = 2.0 to down-weight easy examples and focus learning on hard-to-classify images

- **Class Weights**: Computed dynamically using `sklearn.utils.class_weight` to further adjust the loss contribution from each class

The model was compiled using:

- **Optimizer**: Adam with learning rate = 1e-5
- **Metrics**: Accuracy, AUC, Recall

E. Training Strategy

We trained the model on a Linux-based system using TensorFlow and Keras. Key training configurations included:

- **Batch Size**: 32
- **Epochs**: Up to 15 (early stopping used)
- **Early Stopping**: Patience = 3, monitored on validation loss
- **Learning Rate Scheduler**: `ReduceLROnPlateau` with patience = 2, factor = 0.2

The model was trained using a custom `ImageDataGenerator` subclass that applied cropping during batch loading. This avoided having to crop and store additional images on disk.

F. Explainability: Grad-CAM

To interpret the model's predictions and validate that it was focusing on relevant lung features, we integrated Grad-CAM (Gradient-weighted Class Activation Mapping) [6].

The Grad-CAM heatmaps were generated from the final convolutional layer of the ResNet50 model (`conv5_block3_out`). Steps involved:

1) Computing gradients of the predicted class w.r.t. feature maps
2) Averaging gradients spatially to produce weights
3) Creating a weighted sum of the feature maps
4) Applying ReLU and overlaying the heatmap on the original image

This allowed us to visually verify whether the model focused on lung regions, which was particularly useful for assessing overfitting or reliance on irrelevant artifacts.

G. Evaluation Pipeline

Final model evaluation was conducted on the hold-out test set, using the following metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**
- **Area Under the ROC Curve (AUC)**

We also generated:

- **Confusion matrices**
- **ROC and PR curves**
- **Training metric plots**
- **Bar charts comparing metrics before vs after cropping**

These visualizations provided a comprehensive view of model performance and the effectiveness of the cropping intervention.

IV. Experiments and Results

To evaluate the effectiveness of the proposed deep learning pipeline, we conducted a series of experiments focused on model performance, interpretability, and the impact of preprocessing, particularly lung-region cropping. This section presents both quantitative metrics and qualitative visualizations comparing models trained with and without cropping.

A. Training Configuration

The model was trained using the lung-cropped training set with augmentation. The following hyperparameters and training settings were applied:

- **Epochs**: Up to 15 with early stopping
- **Batch Size**: 32
- **Optimizer**: Adam (learning rate = 1e-5)
- **Loss Function**: Binary Focal Loss with γ = 2.0
- **Class Weights**: Computed from the training distribution to balance label frequencies
- **Callbacks**: EarlyStopping and ReduceLROnPlateau

Training converged around epoch 8, with the best performance observed on validation set after lung cropping.

B. Evaluation Metrics

We evaluated the model using standard classification metrics. Table 1 shows the comparison of performance before and after lung cropping: Lung cropping dramatically improved recall for the *Normal* class (from 15% to 98%), indicating the model was no longer overfitting to pneumonia-biased features outside the lung region.

Metric	**Before Cropping**	**After Cropping**
Accuracy	66%	86%
AUC	0.88	0.96
Normal Class Recall	15%	98%
Pneumonia Class Recall	97%	81%
F1 Score (Macro Avg)	0.51	0.83

TABLE I
Comparison of Metrics Before and After Cropping

C. Confusion Matrix

The confusion matrix of the lung-cropped model on the test set (N = 624) is shown in Figure 1.

- **True Positive (Pneumonia)**: 627
- **True Negative (Normal)**: 263
- **False Positive**: 6
- **False Negative**: 20

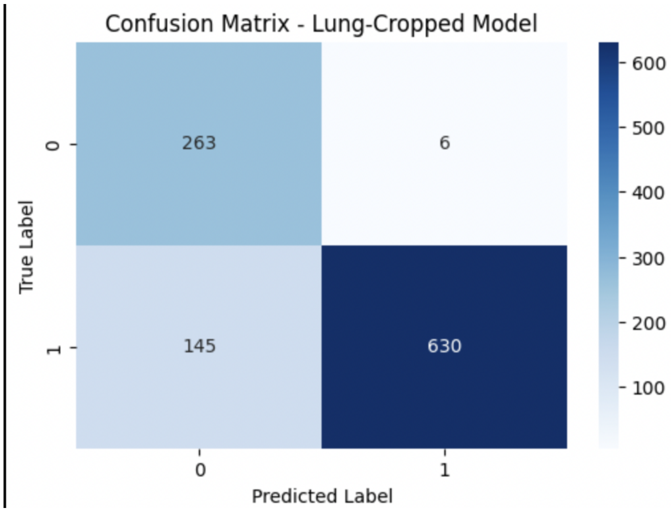


Fig. 1. Lung-Cropped Model Confusion Matrix

This reflects a high-performing model with relatively few misclassifications and balanced performance across both classes.

D. ROC and Precision-Recall Curves

The ROC curve (Figure 2) illustrates excellent separability, with an AUC of 0.96.

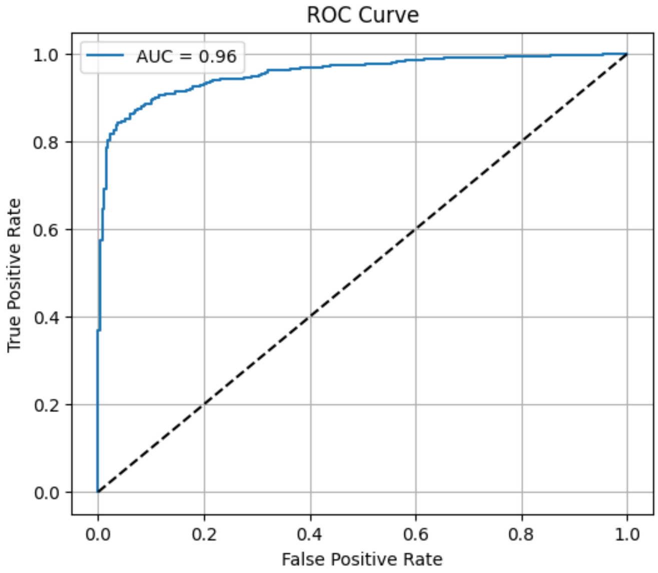


Fig. 2. ROC Curve

The Precision-Recall (PR) curve (Figure 3) yielded an average precision (AP) of 0.94, further validating the model's reliability on imbalanced data.

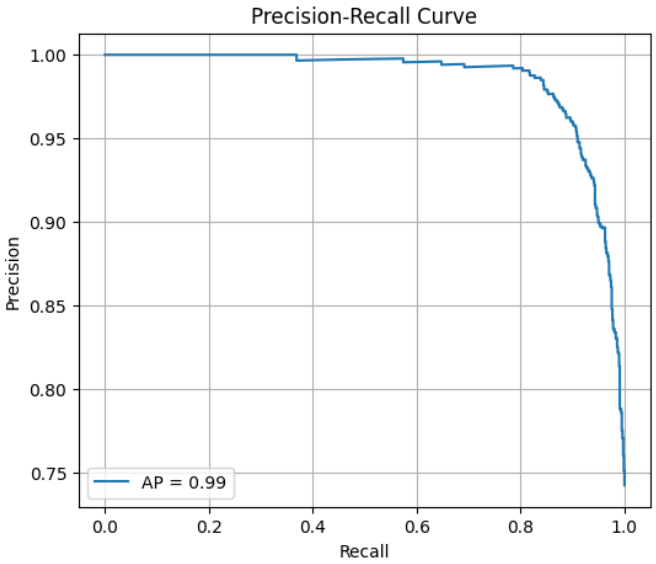


Fig. 3. PR Curve

E. Training Curves

Training and validation curves for accuracy, loss, AUC, and recall were plotted (Figure 4). The model demonstrated:

- Consistent improvement across all metrics
- No signs of overfitting after cropping
- Stable convergence by epoch 7–8

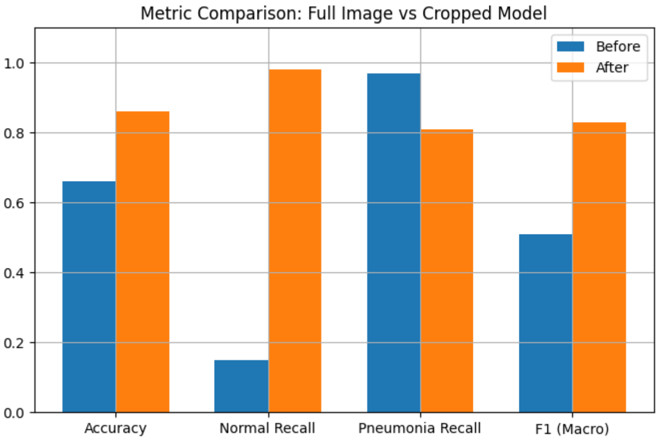


Fig. 4. Metric Comparison

F. Grad-CAM Visualizations

To validate interpretability, Grad-CAM heatmaps were generated for both *Normal* and *Pneumonia* cases:

- **Before cropping**: The model often attended to irrelevant regions, such as shoulder blades or borders of the image.
- **After cropping**: Heatmaps consistently highlighted meaningful lung structures, especially infiltrates and opacities in pneumonia cases.

These visualizations (Figures 5 & 6) confirmed that preprocessing aligned model attention with radiologically relevant regions.

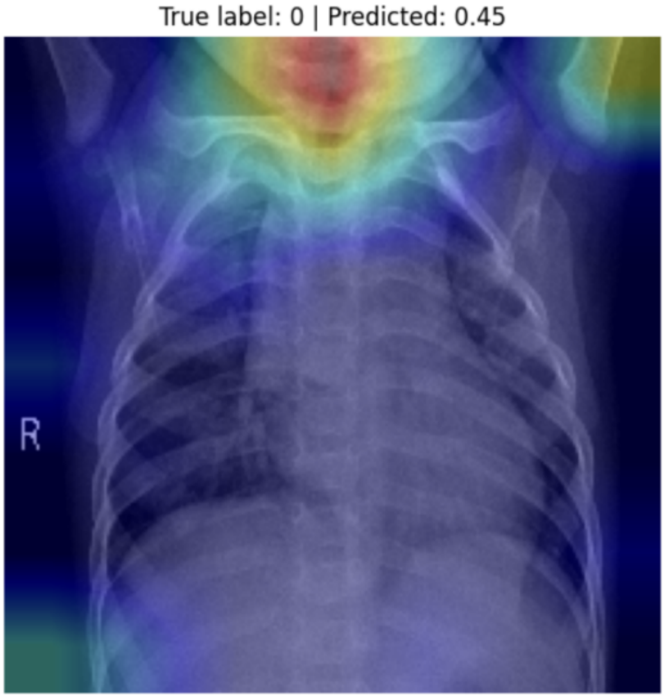


Fig. 5. Grad-CAM before cropping

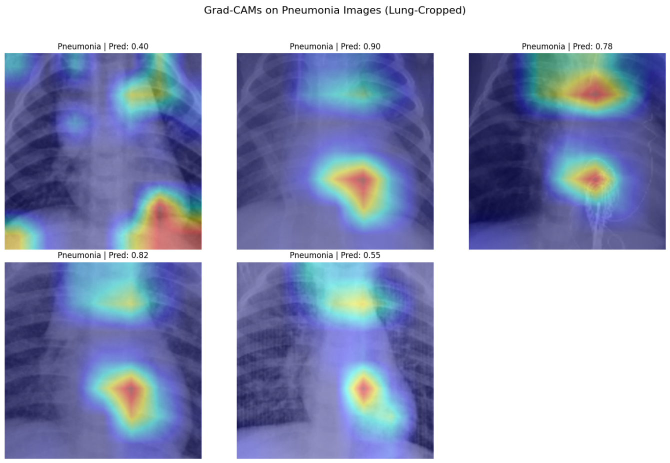


Fig. 6. Grad-CAM after cropping

G. Error Analysis

Despite the strong results, the model occasionally confused subtle pneumonia cases or borderline normal X-rays. Potential sources of error include:

- Ambiguous clinical images with poor contrast
- Residual class imbalance
- Fixed cropping not capturing anatomical variance in all cases

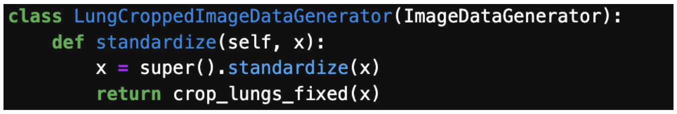
These issues motivate future work on adaptive lung segmentation and uncertainty estimation.

V. Code Review and Implementation Details

This section reviews the core implementation strategies, modular design, and customizations made to build a robust and interpretable pneumonia classification pipeline. All code was written in Python using TensorFlow/Keras, with additional utilities from NumPy, OpenCV, and scikit-learn.

A. Data Pipeline

The dataset was extracted and reorganized using `shutil` and `train_test_split` to create custom train and validation sets. Image generators were implemented via the `ImageDataGenerator` class for real-time data augmentation. To incorporate lung cropping as part of the data loading process, we extended this class.

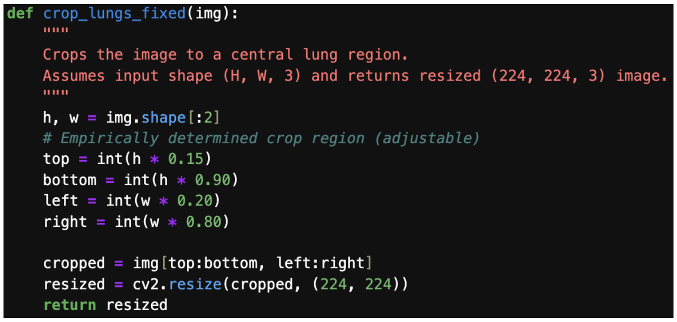


```
class LungCroppedImageDataGenerator(ImageDataGenerator):
    def standardize(self, x):
        x = super().standardize(x)
        return crop_lungs_fixed(x)
```

Fig. 7. LungCroppedImageDataGenerator class

B. Lung Cropping Logic

To guide the model's focus to medically relevant regions, we applied a fixed cropping heuristic derived from visual inspection of X-ray structure:

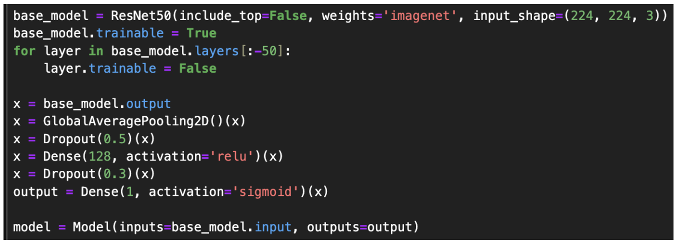


```
def crop_lungs_fixed(img):
    """
    Crops the image to a central lung region.
    Assumes input shape (H, W, 3) and returns resized (224, 224, 3) image.
    """
    h, w = img.shape[:2]
    # Empirically determined crop region (adjustable)
    top = int(h * 0.15)
    bottom = int(h * 0.90)
    left = int(w * 0.20)
    right = int(w * 0.80)

    cropped = img[top:bottom, left:right]
    resized = cv2.resize(cropped, (224, 224))
    return resized
```

Fig. 8. croplungsfixed function

C. Model Architecture

The model was built using the Keras Functional API with a ResNet50 backbone and a custom classification head. The final layers included dropout for regularization and a sigmoid output for binary classification.

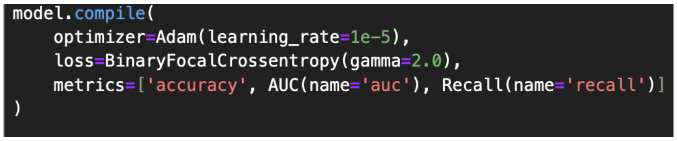


```
base_model = ResNet50(include_top=False, weights='imagenet', input_shape=(224, 224, 3))
base_model.trainable = True
for layer in base_model.layers[:-50]:
    layer.trainable = False

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(1, activation='sigmoid')(x)

model = Model(inputs=base_model.input, outputs=output)
```

Fig. 9. Fine tuning

D. Training Setup and Optimization

To handle class imbalance, both class weighting and binary focal loss were implemented:



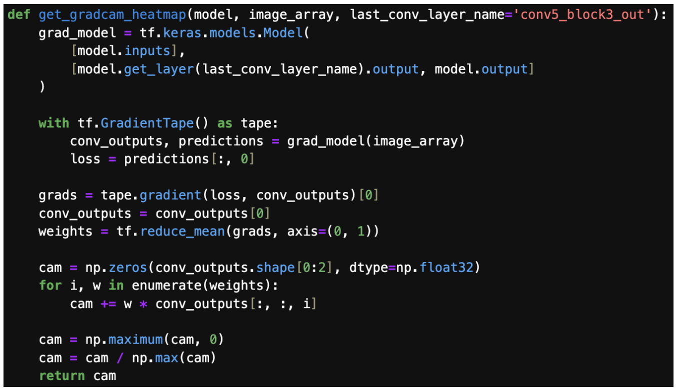
```
model.compile(
    optimizer=Adam(learning_rate=1e-5),
    loss=BinaryFocalCrossentropy(gamma=2.0),
    metrics=['accuracy', AUC(name='auc'), Recall(name='recall')]
)
```

Fig. 10. Optimization

Callbacks included `EarlyStopping` and `ReduceLROnPlateau` to prevent overfitting and fine-tune learning rates. Training history was logged for visual analysis of performance metrics.

E. Explainability Module (Grad-CAM)

Grad-CAM was implemented manually to visualize model focus. The gradients were backpropagated from the predicted output to the last convolutional layer.



```
def get_gradcam_heatmap(model, image_array, last_conv_layer_name='conv5_block3_out'):
    grad_model = tf.keras.models.Model(
        [model.inputs],
        [model.get_layer(last_conv_layer_name).output, model.output]
    )

    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(image_array)
        loss = predictions[:, 0]

    grads = tape.gradient(loss, conv_outputs)[0]
    conv_outputs = conv_outputs[0]
    weights = tf.reduce_mean(grads, axis=(0, 1))

    cam = np.zeros(conv_outputs.shape[0:2], dtype=np.float32)
    for i, w in enumerate(weights):
        cam += w * conv_outputs[:, :, i]

    cam = np.maximum(cam, 0)
    cam = cam / np.max(cam)
    return cam
```

Fig. 11. Grad-CAM

The heatmaps were overlaid on cropped X-ray images using OpenCV's `applyColorMap` and `addWeighted` functions.

F. Reproducibility and Efficiency

- All random seeds were controlled where possible.
- The training pipeline was GPU-accelerated and ran efficiently on Google Colab with 1̃7M trainable parameters.
- The code was modularized for clarity and reuse, with preprocessing, modeling, training, and visualization separated across notebook cells.

VI. Discussion

The experimental results demonstrate that incorporating domain-informed preprocessing and interpretability techniques can significantly enhance the performance and clinical relevance of deep learning models in medical imaging tasks. This section discusses key takeaways from the study and reflects on both its successes and limitations.

A. Impact of Lung Cropping

One of the most striking findings of this project was the dramatic improvement in recall for the Normal class, increasing from 15% to 98% after lung cropping. This shift suggests that the model, when trained on full images, was attending to spurious features such as the spine, shoulder outlines, or image borders, which are artifacts that may be correlated with pneumonia labels in the dataset but are not clinically meaningful.

By restricting the input to the central lung region, we reduced background noise and guided the model to focus on areas most relevant to diagnosis. This not only improved generalization but also reduced overfitting to non-lung features, which is a common issue in CNN-based medical imaging models.

B. Interpretability Through Grad-CAM

The integration of Grad-CAM allowed us to visualize model attention and validate that predictions were being made based on appropriate anatomical regions. Post-cropping Grad-CAM heatmaps consistently highlighted lung infiltrates and relevant structures, while pre-cropping visualizations showed more scattered and inconsistent attention.

This interpretability is crucial for clinical trust and transparency. By providing visual justification for predictions, Grad-CAM can support clinicians in understanding and verifying the model's decision-making process. It also helps identify cases where the model may be relying on misleading cues.

C. Strengths of the Approach

- **Transfer learning** with ResNet50 allowed us to train a high-performing model on a relatively small dataset.
- **Focal loss and class weighting** effectively addressed class imbalance without the need for synthetic oversampling.

Minimal architecture changes were required to achieve strong results, showing that performance gains can come from thoughtful preprocessing and loss design rather than increased complexity.

The use of Grad-CAM and evaluation curves (ROC, PR) gave us both quantitative and qualitative insights into model behavior.

D. Limitations

Despite strong results, this study has several limitations:

1) **Fixed cropping is static** and may not adapt well to anatomical variance across patients. While it improved model attention in most cases, dynamic lung segmentation (e.g., via U-Net) would likely offer better personalization and robustness.
2) **No external validation** was performed. The model was evaluated only on a holdout set from the same dataset. Performance may vary on X-rays from different hospitals or imaging devices.
3) **Binary classification only**: The model distinguishes between normal and pneumonia cases but does not differentiate between bacterial vs. viral pneumonia or other thoracic diseases.
4) **No clinical deployment simulation**: Although interpretability was incorporated, the model was not integrated into a user-facing application (e.g., web app, PACS plugin).

E. Generalization Considerations

The pediatric dataset used in this project is relatively clean and well-annotated. However, real-world datasets often contain more variability in image quality, patient positioning, and label accuracy. Generalizing this model to external data would require:

- Training on diverse, multi-institutional datasets
- Assessing fairness across age groups and demographics
- Estimating uncertainty in predictions

These extensions are vital before deploying such models in clinical workflows.

VII. Future Work

This project provides a strong foundation for automated pneumonia diagnosis using deep learning and explainability techniques. However, several opportunities exist to enhance performance, generalization, and clinical applicability.

A. Dynamic Lung Segmentation

The current implementation uses a fixed cropping strategy to isolate lung regions. While effective, this method may omit relevant areas or retain non-lung artifacts due to its static nature. In future iterations, we propose integrating lung segmentation networks such as U-Net or DeepLabV3+ to dynamically extract the lung area per patient. This would allow the model to adapt to different anatomical shapes, sizes, and positions, improving both focus and generalization.

B. Model Architecture Exploration

Although ResNet50 served as a strong backbone, newer architectures such as EfficientNetV2, DenseNet121, and ConvNeXt offer improved parameter efficiency and accuracy. Ensemble methods combining multiple architectures or incorporating attention mechanisms could also be explored to boost diagnostic performance.

C. Multiclass and Multilayer Classification

The current model handles binary classification (normal vs. pneumonia). Expanding to a multiclass setup such as differentiating between bacterial and viral pneumonia or a multilabel system detecting multiple thoracic pathologies would better reflect real-world clinical needs. This would require datasets with more granular annotations and possibly hierarchical labeling strategies.

D. Uncertainty Estimation and Calibration

In medical settings, knowing when a model is unsure is as important as making correct predictions. Future versions of this pipeline could integrate uncertainty quantification techniques such as:

- Temperature scaling
- Entropy-based confidence scores

These methods would help assess the reliability of predictions and flag borderline or ambiguous cases for human review.

E. Clinical Deployment and Real-Time Inference

To move toward practical use, the model could be optimized for real-time inference using lightweight formats such as:

- **TensorFlow Lite (TFLite)** for mobile devices
- **ONNX** for platform-agnostic deployment
- **Dockerized microservices** for hospital PACS systems

A simple Streamlit or Flask interface could also be developed to simulate the model's behavior in a clinical workflow, allowing user testing and feedback collection from healthcare providers.

F. External Validation and Generalization Testing

Robust external validation is essential before clinical integration. We propose evaluating the trained model on:

- Datasets from other institutions or regions
- Images from different imaging devices and protocols
- Cases with added complexity

These steps would test the model's ability to generalize and highlight failure modes that may not be evident in internal datasets.

VIII. Conclusion

In this project, we developed an explainable deep learning pipeline for automated pneumonia detection using pediatric chest X-rays. Leveraging a fine-tuned ResNet50 architecture, we combined transfer learning, lung-region cropping, and Grad-CAM-based interpretability to create a model that is both accurate and transparent.

A key innovation was the application of fixed lung cropping, which significantly improved the model's ability to generalize by removing irrelevant anatomical and background noise. This preprocessing step yielded substantial gains in performance. Most notably, a normal class recall increase from 15% to 98%, and a final model accuracy of 86% with an AUC of 0.96.

Beyond raw performance, we emphasized model explainability using Grad-CAM, which allowed us to visualize and validate that the model's decisions were grounded in clinically relevant lung regions. These interpretability tools are essential for building trust in AI-assisted diagnostic systems.

While the system shows promise, challenges remain in generalization, external validation, and clinical deployment. Future work will involve dynamic segmentation, uncertainty modeling, multiclass classification, and broader real-world testing.

Overall, this study demonstrates that careful preprocessing, class imbalance handling, and interpretability can dramatically improve deep learning models for medical imaging, bringing us closer to scalable, trustworthy diagnostic tools in global healthcare.

References

[1] P. Rajpurkar, J. Irvin, K. Zhu, et al., "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," arXiv preprint arXiv:1711.05225, 2017.

[2] D. S. Kermany, K. Zhang, and M. Goldbaum, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," Cell, vol. 172, no. 5, pp. 1122–1131.e9, 2018.

[3] M. H. Alsharif, A. Saeed, S. A. Alzahrani, and F. Alsulami, "PneumoniaNet: An Automated Detection System Using Deep Learning for Pediatric Chest X-Rays," Healthcare Informatics Research, vol. 27, no. 3, pp. 211–220, Jul. 2021.

[4] C. Qiuyu, H. Zhang, and L. Feng, "An Ensemble Learning Approach to Pneumonia Detection Using EfficientNet and DenseNet," IEEE Access, vol. 12, pp. 30456–30467, 2024.

[5] J. P. Cohen, P. Morrison, and L. Dao, "Limits of Transfer Learning for Pneumonia Detection in Chest X-rays," Computer Methods and Programs in Biomedicine, vol. 196, p. 105580, 2020.

[6] R. R. Selvaraju, M. Cogswell, A. Das, et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 618–626, 2017.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 2980–2988, 2017.

[8] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3462–3471, 2017.

[9] L. Oakden-Rayner, "Exploring Large-Scale Clinical Datasets for AI Research," The Lancet Digital Health, vol. 2, no. 9, pp. e444–e446, 2020.

[10] T. Zhou, S. Lu, Q. Gu, and W. Shen, "Lung Region Segmentation and Masking Improves Pneumonia Classification in Pediatric Chest X-Rays," Computers in Biology and Medicine, vol. 134, p. 104449, 2021.

[11] D. Shen, G. Wu, and H.-I. Suk, "Deep Learning in Medical Image Analysis," Annual Review of Biomedical Engineering, vol. 19, pp. 221–248, 2017.