Computer Science Project – DLMCSPCSP01

University of Applied Science - Online

Computer Science - Master of Science (M.Sc.)

Comparative Analysis of modern Deep Learning approaches for Pneumonia Detection

Alexander Szabados

Matrikelnummer: IU14086153

Advisor: Dr. Oezdemir Cetin

Delivery date: January 27, 2025

I Abstract

Medical image classification plays a vital role in healthcare, aiding in precise and timely disease diagnosis. This project focuses on developing a flexible machine learning classification system to evaluate and compare the performance of various models on the Chest X-ray Images (Pneumonia) dataset from Kaggle. The dataset consists of 5,863 labeled chest X-ray images divided into two categories: Normal and Pneumonia.

The primary objective is to identify the best-performing model for pneumonia detection by leveraging transfer learning with pre-trained convolutional neural networks such as DenseNet131, EfficientNet, ResNet18, ResNet50, and exploring transformer-based methods like Visual Transformers (ViT). The project emphasizes fine-tuning these models to ensure optimal performance on the dataset.

To comprehensively evaluate model performance, metrics including Accuracy, F1-Score, Precision, Recall, Specificity, AUC/ROC curves, and confusion matrices are employed. Additionally, Grad-CAM is utilized for visualization and explainability, highlighting the critical lung regions influencing the models' predictions.

Furthermore, an easy-to-use Gradio interface has been developed to provide pneumonia classification, confidence ratings, and Grad-CAM visualization for uploaded X-ray images. This interface simplifies accessibility and usability, allowing users to obtain model predictions and insights quickly, demonstrating real-world applicability in assisting healthcare professionals.

The ultimate goal is to build a robust and adaptable classifier written with Pytorch Lightning, capable of reliably distinguishing between healthy lungs and pneumonia-affected lungs. This project underscores the potential of deep learning models in advancing medical image analysis and supporting radiologists in early disease diagnosis through systematic model comparison and explainability.

Contents

II List of Figures

III List of Tables

IV Abbreviations

AI Artificial Intelligence
AUC Area Under the Curve
CE Conformité Européenne
CNN Convolutional Neural Network
FDA US Food and Drug Administration
Grad-CAM Gradient-weighted Class Activation Mapping
LLM Large Language Model
ML Machine Learning
PACS Picture Archiving and Communication System
RIS Radiology Information Systems
ROC Receiver Operating Characteristic
SMOTE Synthetic Minority Oversampling Technique
TTL Truncated Transfer Learning
U-Net Universal Network
ViTs Vistion Transformers

1 Introduction

Pneumonia is a severe respiratory infection that remains a significant global health challenge, especially in children under five years old and older adults. Timely diagnosis is critical for effective treatment and can significantly reduce mortality rates. Radiological imaging, particularly chest X-rays, is a primary diagnostic tool for detecting pneumonia. However, the reliance on radiologists to interpret these images can lead to delays and variability in diagnoses due to subjective assessments and workload constraints.

The advent of deep learning has introduced a transformative approach to medical image analysis, including pneumonia detection. Convolutional Neural Networks (CNNs) have shown remarkable promise in automating the classification of chest X-ray images, offering consistent and rapid diagnosis. Despite these advances, several challenges persist. Imbalanced datasets, variability in image quality, and the need for flexible classifiers capable of adapting to different architectures and preprocessing techniques are some of the obstacles faced by researchers and practitioners.

This project aims to address these challenges by developing a flexible and robust image classification pipeline tailored for pneumonia detection. The pipeline incorporates innovative preprocessing techniques, including dataset reordering, image equalization, and lung segmentation. By systematically evaluating multiple state-of-the-art architectures—such as ResNet, DenseNet, EfficientNet, and Vision Transformers, this project aims to identify the optimal configuration for accurate and reliable classification. Transfer learning is employed to leverage pre-trained models, significantly reducing the computational requirements and time needed for training while maintaining high accuracy, especially given the limited size of the pneumonia dataset.

Furthermore, this research emphasizes the importance of interpretability and performance evaluation. Techniques such as GradCAM are employed to provide visual explanations for model predictions, enhancing trust in automated systems. Evaluation metrics, including F1-score, specificity, and precision-recall curves, are carefully analyzed to ensure a comprehensive understanding of model performance.

To bridge the gap between research and practical application, a Gradio interface has been developed to make the pneumonia classification model accessible to a wider audience, including medical professionals and researchers. This web-based tool provides an intuitive way to interact with the model by allowing users to upload chest X-ray images and receive automated classification results along with a confidence score. Additionally, the Grad-CAM visualization highlights the most relevant regions in the X-ray that influenced the model's decision, offering insights that could aid in clinical interpretation. By integrating this interactive feature, the project ensures that the developed model is not just a theoretical framework but a real-world tool that can contribute to more efficient and explainable pneumonia detection.

By systematically exploring and documenting the challenges and solutions in pneumonia classification, this project not only aims to deliver a practical tool for medical practitioners but also contributes valuable insights to the broader field of medical image analysis. The methodology and findings presented in this work have the potential to inform future research and applications, ultimately advancing the role of artificial intelligence in healthcare.

1.1 Scope

The project emphasizes:

- Comparing multiple deep learning architectures (ResNet, DenseNet, EfficientNet, and ViTs) for pneumonia detection.
- Evaluating advanced training strategies, including transfer learning, gradual unfreezing, and class balancing.
- Preprocessing techniques like lung segmentation and histogram equalization to enhance diagnostic accuracy.
- Exploring model interpretability through Grad-CAM visualizations.
- Deployment of a Real-World Application, utilizing a Gradio interface, that lets users make prediction with the resulting models. For setup see the instructions on the Github Repository.

This study focuses on the detection of pneumonia using chest X-ray images from the publicly available Chest X-ray Dataset.

Kaggle Dataset

https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

Github Repository

https://github.com/SzabadosA/pneumonia_detection.git

Full Technical Documentation

https://szabadosa.github.io/pneumonia_detection

Processed Images and Pretrained Models

https://drive.google.com/drive/folders/1MIcgEqpcMU24N4IEFqmBIhDUXR6jAzv2?usp=sharing

1.2 Limitations

While the study provides valuable insights into pneumonia detection using deep learning, several limitations are acknowledged:

- **Dataset Size:** Despite leveraging transfer learning, the relatively small dataset may limit the generalizability of the findings. Larger, more diverse datasets could improve model robustness.
- **Data Imbalance:** The dataset is imbalanced, with more pneumonia cases than normal cases. While class weights and dataset reordering mitigated this issue, residual imbalance may still influence model performance.
- **ViTs' Requirements:** Vision Transformers, although tested, did not perform optimally due to the dataset size and lack of large-scale medical pretraining tailored to chest X-rays.
- **Focus on Pneumonia Only:** The study is limited to binary classification (normal vs. pneumonia) and does not address multi-class classification or other pathologies present in chest X-rays.
- **Lung Segmentation Accuracy:** The lung masks used in preprocessing were generated using a U-Net trained on a small set of annotations. Any inaccuracies in segmentation could propagate through the pipeline, affecting results.
- **Computational Constraints:** Training and testing were conducted on limited computational resources, which constrained the ability to experiment with more complex architectures. All models were trained on a Nvidia GeForce RTX 4070 Ti card.

This research prioritizes performance comparison and explainability, offering insights for AI applications in real-world medical diagnostics while recognizing areas for future improvement.

2 Related Work

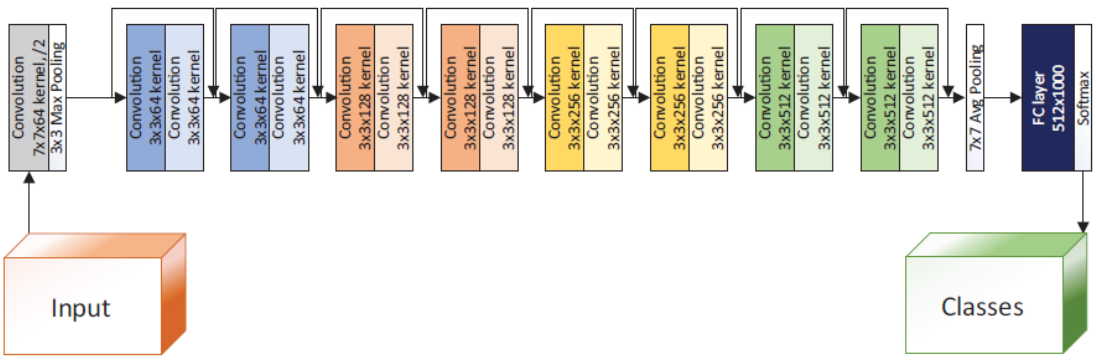
2.1 Approaches to Pneumonia Detection with Deep Learning

Recent advancements in deep learning have enabled the development of diverse architectures tailored for pneumonia detection. Numerous methods have been proposed, ranging from traditional convolutional approaches to novel transformer-based models. Each method offers unique strengths and trade-offs, making it challenging to identify a universally best-performing model. Based on a review of prior research, this study selects a representative subset of models to evaluate their effectiveness for pneumonia classification.

2.1.1 Convolutional Neural Networks (CNNs)

CNNs have long been the backbone of image classification tasks, including pneumonia detection. These networks work by applying convolutional layers to extract spatial hierarchies of features, ranging from low-level edges to high-level patterns. This hierarchical learning approach enables CNNs to identify critical pathological features in medical images, such as opacities in chest X-rays, making them particularly well-suited for diagnostic tasks. Among these, ResNet and DenseNet have been extensively utilized.

Figure 1: The architecture of ResNet18 model.



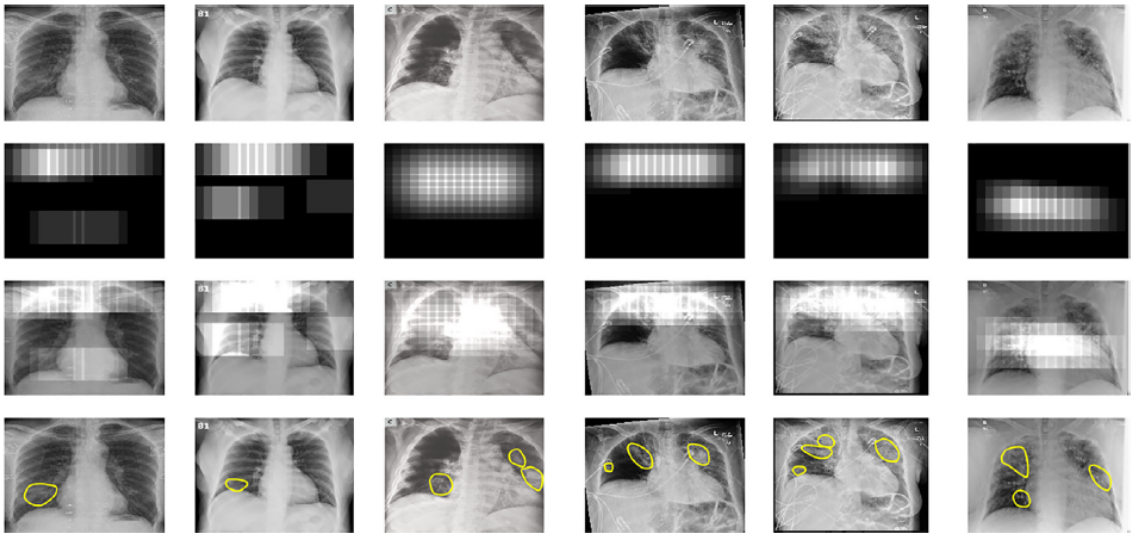
Source: Adapted from Minaee et al. (2020, p. 4).

ResNet (Residual Networks)

ResNet18 and ResNet50 are widely adopted due to their use of residual connections, which address the vanishing gradient problem and enable training deeper networks. (Minaee et al. 2020, p. 3) demonstrated that these architectures achieved high sensitivity rates, up to 98%, on chest X-ray datasets. Further evaluation by the same study revealed that ResNet18 and ResNet50, trained on the COVID-Xray-5k dataset using transfer learning, also achieved specificity rates of 90.7% and 89.6%, respectively (p. 7). The ResNet models outperformed other tested architectures, such as DenseNet121 and SqueezeNet, in terms of sensitivity while maintaining competitive specificity.

The study also highlighted the models' robustness across varying thresholds. For instance, at a threshold of 0.2, ResNet18 achieved 95% sensitivity and 92.4% specificity, demonstrating adaptability to changing classification requirements. Additionally, the use of heatmaps generated by ResNet18 identified infected regions in chest X-rays with high agreement to annotations by board-certified radiologists, further validating its application in clinical scenarios (p. 8-9) and adding explainability to the trained model.

Figure 2: COVID-19 infected regions detected by our ResNet18 model, in six chest X-ray images from the test set. Vertical sets give the Original images (top row), COVID-19 region heatmap (2nd row), heatmap overlaid on the image (3rd row), and the independent standard of radiologist-marked COVID-19 disease regions (bottom row).



Source: Adapted from Minaee et al. (2020, p. 8).

The results underscore ResNet's reliability in medical imaging tasks, particularly for binary classification problems such as pneumonia or COVID-19 detection. Its strong performance across sensitivity, specificity, and visualization of disease-affected areas makes it a leading choice for automated diagnostic systems in radiology.

DenseNet (Dense Convolutional Networks)

DenseNet employs dense connectivity across layers to enhance feature propagation and reduce redundancy. Roy et al. (2024) noted DenseNet's adaptability to subtle pathological signs in chest X-rays (p. 339), outperforming all other tested models.

Table 1: Transfer learning results of CNN models for 2-class classification.

Model	Acc	Pre	Rec	AUC	F1
DenseNet169	92.95	91.39	97.95	96.41	94.55
MobileNet-v2	84.78	86.97	88.97	89.41	87.96
Xception	75.80	72.09	100.00	73.68	83.78
ResNet-50	68.59	66.55	100.00	77.35	79.92
VGG-16	75.80	72.50	98.72	85.35	83.60
NasNet-Mobile	79.81	75.98	98.97	75.09	85.97
Inception-ResNet-v2	81.89	77.64	99.74	94.82	87.32

Source: Adapted from Roy et al. (2024, p. 342)

Further exploration showed that DenseNet121, when enhanced with a fuzzy attention mechanism (FCSSAM), achieved a binary classification accuracy of 97.15% and significantly improved recall (+1.1%) and AUC (+2.74%) compared to baseline DenseNet models (Roy et al. 2024, p. 342). This adaptability underscores DenseNet's robustness in handling class imbalances and subtle variations in pathological features.

EfficientNet

EfficientNet balances accuracy and computational efficiency by systematically scaling network depth, width, and resolution. Sharma (2024, p. 1) employed the EfficientNet model for classifying COVID-19 from CT scan images, achieving a training accuracy of 99.54% and a testing accuracy of 98.24%, with minimal training and testing losses of 0.015 and 0.145, respectively. The model's superior performance, compared to ResNet50 and VGG19, was attributed to its compound scaling technique, which allows the model to effectively adapt to varying image resolutions and complexities. By leveraging transfer learning and augmenting datasets, the EfficientNet model demonstrated state-of-the-art accuracy and generalization capabilities, proving its efficacy for medical imaging tasks (Sharma 2024, p. 3-5).

2.1.2 Vision Transformers (ViT)

Vision Transformers (ViT) have emerged as a transformative approach in medical image analysis, offering distinct advantages over traditional CNN-based methods. Unlike CNNs, which are inherently local in their feature extraction, ViTs employ self-attention mechanisms to model global dependencies within an image. Siddiqi et al. (2024, p. 4) highlight the potential of ViTs in overcoming CNNs' limitations by capturing both long-range interactions and complex structural features, which are critical for accurate pneumonia detection in chest X-rays. However, they also emphasize the challenges posed by ViTs, including their reliance on large datasets and computational complexity, which can be barriers in medical applications with limited data availability (p. 8).

Manzari et al. (2023) introduced MedViT, a specialized transformer architecture designed for medical imaging tasks. MedViT incorporates convolutional blocks alongside transformer layers to combine the local feature extraction strength of CNNs with the global modeling capabilities of transformers. This hybrid architecture demonstrates enhanced robustness to adversarial attacks and generalization across diverse datasets, including chest X-rays. The study reported that MedViT outperformed ResNet variants in both accuracy and AUC across multiple datasets, underscoring its effectiveness for medical image classification tasks (Manzari et al. 2023, p. 7). Furthermore, MedViT's hierarchical design allows for computational efficiency, making it a viable option for clinical deployment (Manzari et al. 2023, p. 10).

These findings collectively illustrate the transformative potential of CNNs and ViTs in medical imaging, showcasing that a wide variety of approaches can achieve success if carefully implemented. This underscores the importance of tailoring methods to specific tasks and datasets. This project aims to build a flexible pipeline that facilitates cross-validation between different models, enabling systematic comparisons and fostering adaptability to diverse medical imaging challenges, while also addressing the computational demands and data requirements of innovative approaches.

2.2 Transfer Learning in Medical Imaging

Transfer learning has emerged as a critical strategy in addressing the challenges of limited labeled data in medical imaging. It involves reusing a model trained on a large source dataset to improve performance on a smaller, domain-specific target dataset. Typically, this involves freezing the initial layers of the pre-trained model, which capture general features, and fine-tuning the later layers to adapt to the specific target task. By leveraging pre-trained models on large-scale datasets, transfer learning reduces computational costs and training time while maintaining high performance.

Minaee et al. (2020, p. 3) applied transfer learning using ResNet variants, achieving notable improvements in COVID-19 detection sensitivity (p. 3). Similarly, , p. 1016 combined pre-trained models with advanced preprocessing techniques, demonstrating enhanced generalization and accuracy. Additionally, Hasse et al. (2024, p. 4) emphasized the significance of transfer learning for chest X-ray classification, showcasing its capacity to reduce overfitting when applied to small datasets by leveraging extensive pre-training on large image repositories (p. 4). This finding aligns with recent efforts in optimizing transfer learning strategies for domain-specific medical tasks.

2.2.1 Selective Fine-Tuning

Peng et al. (2021, p. 1) proposed Truncated Transfer Learning (TTL), which fine-tunes the lower layers of pre-trained models while discarding redundant upper layers, optimizing efficiency and reducing overfitting. This approach highlights the adaptability of transfer learning to domain-specific tasks.

Transfer learning bridges the gap between limited medical datasets and the need for robust models, making it an essential component of modern medical AI pipelines.

2.3 Comparative Evaluation of CNNs and ViTs for Pneumonia Detection

Deep learning approaches have demonstrated their potential for enhancing pneumonia detection, with CNNs and ViTs representing two distinct paradigms.

2.3.1 Performance of CNNs

CNNs excel at capturing localized features in medical images, such as opacities and consolidations in lungs. DenseNet and ResNet, as noted by and Minaee et al. (2020, p. 1), have consistently achieved high precision and recall, particularly in structured datasets. Their hierarchical feature extraction enables them to identify pathological regions effectively, making them reliable for pneumonia classification.

2.3.2 Role of Vision Transformers

ViTs, in contrast, focus on modeling global patterns and relationships across an image. Siddiqi et al. (2024, p. 20-21) argue that ViTs' ability to extract long-range dependencies offers a significant advantage over CNNs, particularly in tasks like pneumonia detection where subtle structural features are critical. Manzari et al. (2023, p. 10) further emphasize the robustness of ViTs, noting that architectures like MedViT achieve superior accuracy and generalization across datasets while addressing computational efficiency challenges.

2.3.3 Application in Pneumonia Detection

While CNNs are adept at detecting localized features, ViTs' ability to capture global dependencies provides complementary insights. This study evaluates both CNNs and ViTs independently to compare their effectiveness in pneumonia detection without combining their architectures. Recent findings indicate that both paradigms have distinct strengths, offering valuable perspectives for future advancements.

2.4 Data Preprocessing in Pneumonia Detection

Data preprocessing plays a crucial role in medical image analysis, particularly in addressing challenges such as imbalanced datasets, noisy images, and subtle diagnostic features. Effective preprocessing ensures that the input data is optimized for model training and evaluation, enhancing generalization and reducing biases.

2.4.1 Equalization

Histogram equalization improves image quality by normalizing intensity distributions, making subtle features like opacities more detectable. Çelika et al. (2023, p. 1017) implemented histogram equalization in their preprocessing pipeline, significantly enhancing the visibility of critical diagnostic features in chest X-rays.

2.4.2 Augmentation

Data augmentation techniques, such as flipping, rotation, and brightness adjustments, address data scarcity by artificially increasing dataset diversity. Mahr et al. (2024, p. 1-5) highlighted how augmentation enhances model robustness by simulating real-world variations in X-ray imaging.

2.4.3 Segmentation

Lung segmentation isolates regions of interest, focusing model attention on diagnostically relevant areas. Çelika et al. (2023, p. 1017-1018) utilized Mask R-CNN for lung segmentation, achieving improved classification metrics by eliminating background noise. Similarly, Niu et al. (2020, p. 3786-3787) emphasized the importance of segmentation in reducing false positives during pneumonia detection.

2.4.4 Class Balancing

Addressing class imbalances ensures equitable model performance across categories. Techniques such as SMOTE (Synthetic Minority Oversampling Technique) were employed by Çelika et al. (2023, p. 1016) to mitigate biases and improve recall for minority.

These preprocessing strategies collectively enhance the reliability and accuracy of machine learning models in pneumonia detection, making them indispensable components of modern medical image analysis workflows.

3 Technical Background

This chapter provides a foundational overview of the concepts and methodologies underpinning this study's approach to pneumonia detection. While the Related Work chapter focused on reviewing prior research and results, this section delves into the theoretical and technical principles required to understand the development and evaluation of the models and techniques discussed.

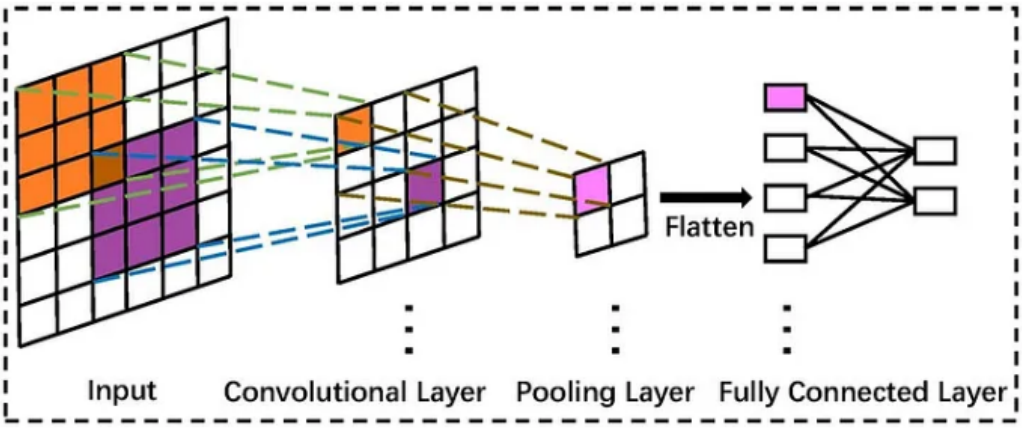
3.1 Fundamentals

3.1.1 Convolutional Neural Networks (CNNs)

CNNs are specialized for image data and use convolutional layers to extract spatial hierarchies of features:

- **Convolutional Layers:** Identify patterns such as edges and textures by applying learnable filters.
- **Pooling Layers:** Reduce spatial dimensions, preserving important features while lowering computational complexity.
- **Fully Connected Layers:** Combine extracted features for classification tasks.

Figure 3: CNN Layer Architecture.



Source: Adapted from VinLab (2025).

CNNs' hierarchical structure enables them to identify increasingly complex patterns as data passes through deeper layers, making them ideal for tasks like pneumonia detection.

3.1.2 Vision Transformers (ViTs)

ViTs represent a paradigm shift by using self-attention mechanisms rather than convolutions:

- **Self-Attention:** Models dependencies between different parts of an image by assigning attention scores.
- **Positional Encoding:** Introduces spatial information into the sequence of image patches.
- **Patches and Tokens:** In ViTs, images are divided into smaller fixed-size patches, which are then flattened and linearly embedded into tokens. These tokens are processed as sequences, similar to words in natural language models, allowing the transformer to learn global patterns across the entire image.

ViTs gained significant popularity through their foundational role in large language models (LLMs), where transformers demonstrated unparalleled capabilities in understanding sequences and capturing long-range dependencies. Building on this success, ViTs have adapted the transformer framework for visual data, making them particularly suited for capturing global context in images. However, they often require large datasets or transfer learning to achieve competitive performance.

3.2 Transfer Learning

3.2.1 Principles

Transfer learning involves reusing pre-trained models developed on large datasets, such as ImageNet, for domain-specific tasks. This reduces training time and data requirements, addressing the challenges of limited annotated medical datasets.

3.2.2 Freezing and Fine-Tuning

- **Freezing Layers:** Retains the knowledge captured in the initial layers, which learn generic features.
- **Fine-Tuning:** Adapts the later layers to the target dataset, focusing on domain-specific patterns.
- **Gradual Unfreezing:** Involves unfreezing and fine-tuning layers incrementally, starting from the last layers and moving backward. This approach helps prevent catastrophic forgetting and ensures that the model progressively adapts to the new dataset while retaining useful features from pre-training.

3.3 Data Preprocessing

Preprocessing ensures data quality and relevance, enhancing model performance and reducing biases.

3.3.1 Histogram Equalization

Equalization adjusts intensity distributions to normalize image quality, improving visibility of diagnostic features like opacities.

3.3.2 Data Augmentation

Augmentation generates diverse training samples through techniques such as:

- **Flipping:** Horizontal or vertical reflection.
- **Rotation:** Adjusting orientation.
- **Brightness Adjustment:** Modifying intensity to simulate real-world conditions

3.3.3 Lung Segmentation

Segmentation isolates regions of interest (e.g., lungs) by masking irrelevant areas, reducing noise and false positives. Methods like U-Net are commonly used for segmentation in medical imaging. U-Net is a convolutional network architecture specifically designed for biomedical image segmentation. U-Net employs encoder-decoder pathways with skip connections, allowing precise localization and boundary detection for target regions, such as lungs.

3.3.4 Class Balancing

Dataset imbalances can be addressed by assigning weights to the classes during model training. This method prevents bias toward the majority class and promotes balanced performance across all classes. By adjusting the loss function to assign higher importance to underrepresented classes, the model's ability to learn from minority class samples improves without artificially augmenting the dataset.

3.4 Evaluation Metrics

Key metrics in medical imaging include:

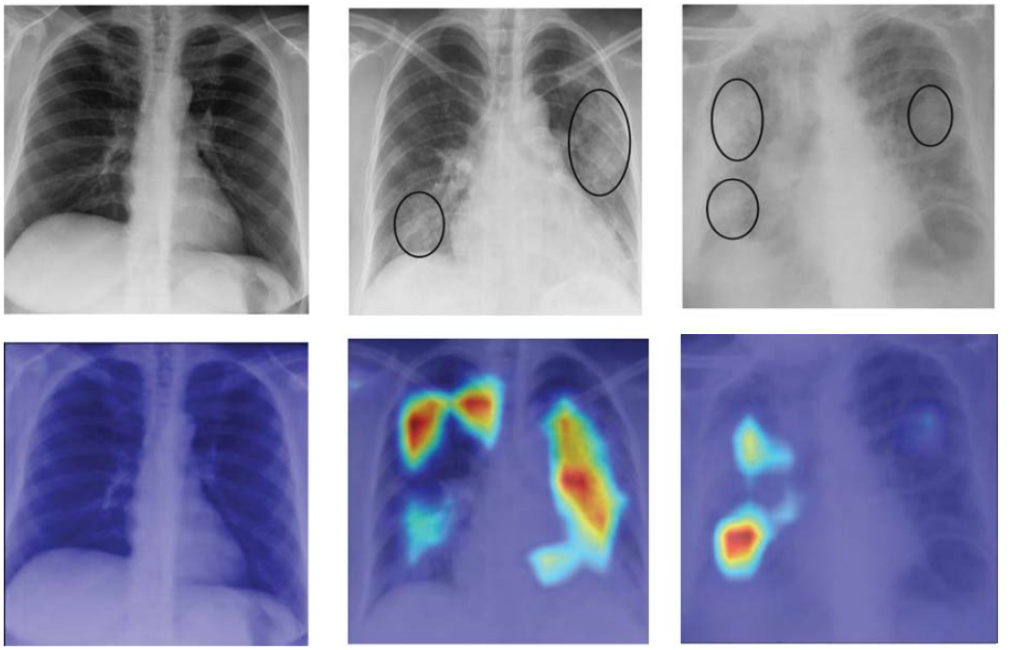
3.4.1 Metrics

- **Accuracy:** Proportion of correct predictions.
- **Precision:** The proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** The proportion of true positive predictions among all actual positive cases.
- **Specificity:** The proportion of true negative predictions among all actual negative cases.
- **F1-Score:** Harmonic mean of precision and recall, providing a balanced measure of model performance.
- **AUC/ROC:** The Area Under the Receiver Operating Characteristic curve measures the model's ability to distinguish between classes at various thresholds, providing a comprehensive evaluation of its discriminatory power.
- **Confusion Matrix:** A tabular summary of true positives, true negatives, false positives, and false negatives, offering detailed insight into the model's classification performance and areas for improvement.

3.4.2 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) provides visual explanations for a model's predictions by highlighting the regions of an image that contribute most to the classification. This technique enhances interpretability in medical imaging, helping clinicians understand model outputs.

Figure 4: Examples of probabilistic Grad-CAM of COVID-19 class. The first row is the original X-ray image and the second row is corresponding visualization map, where the COVID-19 infection areas were marked (using ellipses) by an experienced radiologist.



(a) Normal case (b) COVID-19 (Public dataset) (c) COVID-19 (UTMB dataset)

Source: Adapted from Zhuang et al. (2022, p. 860).

3.5 Frameworks and Tools

3.5.1 PyTorch Lightning

PyTorch Lightning simplifies the development of deep learning models by structuring code for reproducibility and scalability. It abstracts repetitive boilerplate code, enabling faster experimentation and easier debugging. PyTorch Lightning is particularly useful for transfer learning and fine-tuning complex architectures.

3.5.2 U-Net for Segmentation

U-Net, as introduced earlier, is integral for segmenting lung regions in medical imaging datasets. Its encoder-decoder structure enables precise region isolation, improving model focus on diagnostically relevant areas while ignoring irrelevant background features.

4 Methodology

This chapter documents the methodology employed for pneumonia detection using deep learning, structured chronologically to reflect the evolution of the experimental approach. Starting with raw data preparation, each phase of experimentation informed subsequent steps, ultimately culminating in a flexible classification pipeline capable of handling multiple architectures and robust evaluations. The code and specific implementations for each step are available in the project's GitHub repository, with relevant Jupyter Notebook files referenced throughout this chapter.

4.1 Chronology of Experiments

4.1.1 Phase 1: Testing ResNet18 with Raw Dataset

The study began by testing the ResNet18 architecture on the raw Kaggle chest X-ray dataset. This baseline experiment provided insights into the dataset's suitability for model training and highlighted significant class imbalances that needed to be addressed in subsequent phases. While ResNet18 showed adequate initial performance, the results emphasized the need for preprocessing and dataset reorganization.

4.1.2 Phase 2: Reordering the Dataset

To mitigate class imbalances, the dataset was reordered to balance the training, validation, and test splits. This step ensured equitable distribution of pneumonia and normal cases across subsets, reducing potential bias during training and evaluation.

4.1.3 Phase 3: Adding Class Weights

Class weights were introduced during training to further address dataset imbalances. By assigning higher weights to the minority class, the loss function emphasized pneumonia cases, ensuring the model's sensitivity to these critical instances.

4.1.4 Phase 4: Equalizing Images

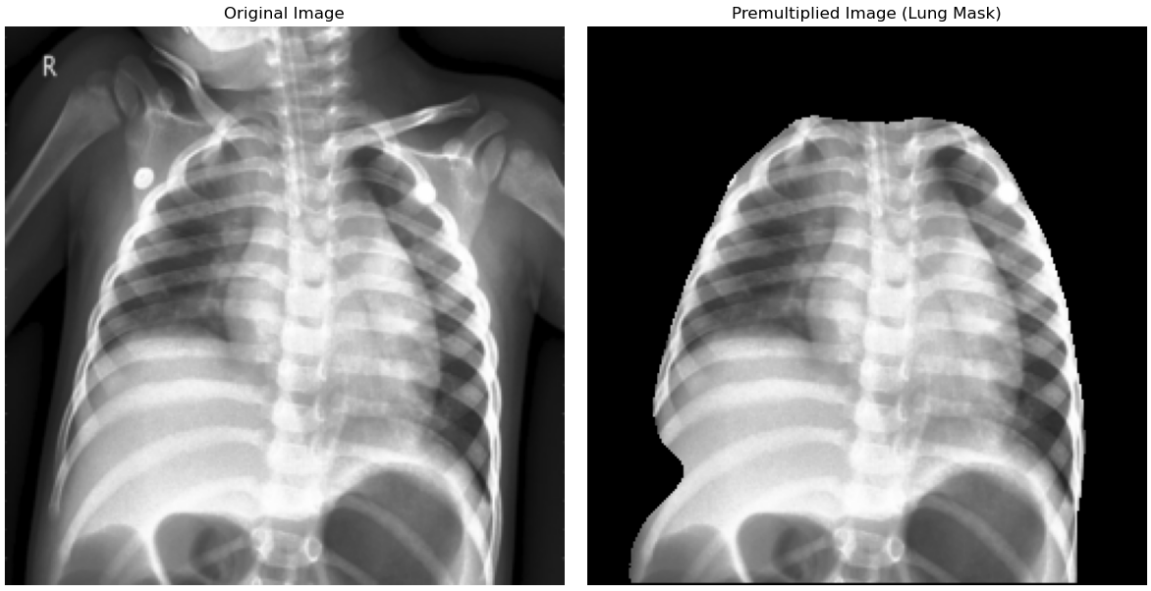
Histogram equalization was applied to normalize image intensity distributions, enhancing the visibility of subtle diagnostic features like lung opacities. This preprocessing step improved feature extraction and model interpretability.

4.1.5 Phase 5: Creating Masks and Using Premultiplied Images

A U-Net model was trained on annotated lung segmentation data to isolate lung regions, focusing the model on diagnostically relevant areas. This phase involved:

- Training the U-Net model to generate lung masks.
- Applying these masks to the equalized images.
- Using the masked, premultiplied images as inputs for training.

Figure 5: Lung Segmentation using custom trained Unet Model



Source: Own representation.

These steps reduced noise and improved the models' performance on pneumonia detection tasks.

4.1.6 Phase 6: Upsampling Images

An experiment was conducted to upscale the dataset images to higher resolutions, aiming to provide more detailed inputs for training. However, this approach did not yield significant performance improvements. This phase informed the decision to focus on other preprocessing enhancements instead.

4.1.7 Phase 7: Unfreezing All Layers

Initially, all layers of the pre-trained models were unfrozen to allow end-to-end training. This approach leveraged the general features learned from pre-training while adapting to the pneumonia dataset. However, this method introduced instability in some architectures, leading to exploration of alternative strategies.

4.1.8 Phase 8: Gradual Unfreezing

A gradual unfreezing strategy was implemented. Layers were unfrozen incrementally, starting from the top, to:

- Prevent catastrophic forgetting.
- Ensure stable fine-tuning.
- Optimize the balance between specificity and recall.

This approach produced the best results across all evaluation metrics, making ResNet50 with gradual unfreezing the top-performing architecture, with a focus on specifity and recall.

4.1.9 Phase 9: Evaluating Vision Transformers

Following the success of ResNet50 with gradual unfreezing, Vision Transformers (ViTs) were evaluated to explore their potential for pneumonia detection. While ViTs showed promise in capturing global patterns due to their self-attention mechanisms, they failed to outperform ResNet50. Specifically, ResNet50 with gradual unfreezing achieved a better balance between specificity and recall, making it the most reliable model for this task under the current experimental conditions.

4.1.10 Phase 10: Experimenting with different model Architectures

In order to find the best performing model, several model architectures were tested, with ResNet50 winning the experiment:

- ResNet18
- ResNet50
- DenseNet131
- EfficientNet
- google/vit-base-patch16-224
- google/vit-large-patch16-384
- google/vit-large-patch32-384

4.2 Hyperparameter Tuning

Optimizing deep learning models for pneumonia detection requires careful hyperparameter tuning, particularly when working with limited medical imaging datasets. This section details the tuning strategies applied to CNNs and Vision Transformers (ViTs), highlighting how the classifier adapted architectures for small-scale data and why synthetic data augmentation was not chosen for this project.

4.2.1 CNN Hyperparameter Tuning

Initial Learning Rate:

CNNs require gradual fine-tuning, learning rates in the range of 1e-4 to 1e-6 were tested. The best performance was achieved with 1e-4, using a step decay scheduler that reduced the learning rate after every 5 epochs.

Optimizer Choice:

Both the Adam and SDG optimizer were tested, with SDG leading to faster results. SDG was implemented with a weight decay of 1e-4 was used to prevent overfitting.

Batch Size and Regularization

Batch sizes between 8 and 32 were tested. A batch size of 20 provided the best trade-off between memory efficiency and gradient stability on a Nvidia GeForce RTX 4070 Ti card.

L2 weight regularization (0.004) was added to convolutional layers to improve generalization.

Dropout (0.2) was applied to fully connected layers to reduce overfitting.

Fine-Tuning Transfer Learning Models

Rather than training CNNs from scratch, the study leveraged pretrained models from ImageNet. Gradual unfreezing of deeper layers achieved the most successful results, with an unfreeze interval of 5 epochs.

These optimizations resulted in a 2-3% improvement in AUC-ROC scores across CNN models.

4.2.2 Vision Transformer Hyperparameter Tuning

Patch Size Optimization

The Vision Transformer (ViT) architecture processes images by dividing them into fixed-size patches before passing them through transformer blocks. Since chest X-rays contain critical fine-grained details, the study tested different patch sizes to determine the optimal configuration:

- 16×16 patches on a 224-pixel image (ViT-Base-Patch16-224)
  Achieved 0.8850 accuracy and a strong F1-score of 0.9160, providing a good balance between precision (0.9804) and recall (0.8595).
- 16×16 patches on a 384-pixel image (ViT-Large-Patch16-384)
  Offered the highest accuracy (0.8998) and F1-score (0.9278), demonstrating that increasing image resolution with the same patch size enhances model performance.
- 32×32 patches on a 384-pixel image (ViT-Large-Patch32-384)
  Resulted in a lower accuracy (0.8850) and F1-score (0.9163), indicating that increasing patch size reduces spatial resolution and hinders feature extraction.

The best performing model used 16×16 patches on a 384-pixel image (ViT-Large-Patch16-384) as it delivered the best performance in pneumonia detection, aligning with standard ViT implementations while maximizing both accuracy and recall.

Regularization Techniques

Dropout (0.2) was applied to self-attention layers.
Layer Normalization was applied instead of traditional Batch Normalization, which was ineffective due to the self-attention mechanism.

Learning Rate and Optimization

A learning rate scheduler was used with:

- Initial Learning Rate: 1e-5
- Minimum Learning Rate: 1e-7

The Adam optimizer with weight decay (1e-5) helped control overfitting. Despite these optimizations, ViTs struggled to match CNN performance on pneumonia detection, likely due to dataset size constraints.

4.2.3 Synthetic Data

Given the limited dataset, synthetic data generation was considered as a potential solution to increase training samples. However, this study decided against it for the following reasons:

Risk of Artifacts Affecting Model Learning

Methods like Generative Adversarial Networks (GANs) or Diffusion Models could synthesize additional X-rays. However, GAN-generated images often introduce artifacts that do not exist in real medical scans. These subtle artifacts could bias model predictions, leading to unreliable medical classifications.

Difficulty in Ensuring Clinical Validity

Unlike natural images, medical X-rays require domain expert validation to ensure synthetic images represent real pathology. Without expert annotation, synthetic pneumonia cases might be inaccurate, misleading the model.

Augmentation Limitations in Medical Imaging

Traditional rotation, flipping, and cropping were tested but introduced problems: Drastic rotation changed the relative position of lung features. Flipping was deemed non-biologically valid. Cropping risked removing pathological features. As a result, only mild augmentations (brightness shifts, contrast adjustments, slight rotations) were used.

Focus on Transfer Learning Over Artificial Expansion

Instead of synthetic data, the study focused on: Transfer learning with large pretrained models. Fine-tuning on real chest X-ray samples. This approach ensured that models learned clinically relevant features rather than relying on potentially misleading synthetic images.

4.3 Evaluation Metrics

The models were evaluated using a comprehensive set of metrics to ensure accuracy, reliability, and interpretability:

4.3.1 Metrics

- **Accuracy:** Proportion of correct predictions.
- **Precision:** Proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** Proportion of true positive predictions among all actual positive cases.
- **Specificity:** Proportion of true negative predictions among all actual negative cases.
- **F1-Score:** Harmonic mean of precision and recall, balancing their contributions.

4.3.2 Importance

Sensitivity (true positive rate) and specificity (true negative rate) are particularly critical in medical applications to minimize misdiagnoses and were therefore prioritized for deciding on the best performing model. False negatives in pneumonia detection could lead to untreated patients spreading the infection, while false positives could result in unnecessary treatments and hospital admissions. These consequences make it essential to optimize both metrics simultaneously, ensuring the model reliably identifies both disease presence and absence to support accurate clinical decisions.

4.3.3 Confusion Matrix

Confusion matrices provided detailed breakdowns of true positives, true negatives, false positives, and false negatives, offering insights into the models' classification performance.

4.3.4 ROC-AUC Curve

Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores assessed the models' ability to distinguish between classes at various thresholds.

4.3.5 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) visualized the regions of input images that influenced model predictions. This interpretability tool validated the clinical relevance of the trained models by highlighting diagnostically important areas.

4.4 Observations and Insights

The chronological approach revealed several key insights:

- Preprocessing steps, particularly lung segmentation and histogram equalization, significantly improved model performance.
- ResNet50 with gradual unfreezing emerged as the best-performing model, demonstrating the effectiveness of controlled fine-tuning.
- ViTs, while promising for global pattern recognition, require larger datasets to achieve optimal performance in medical imaging tasks.

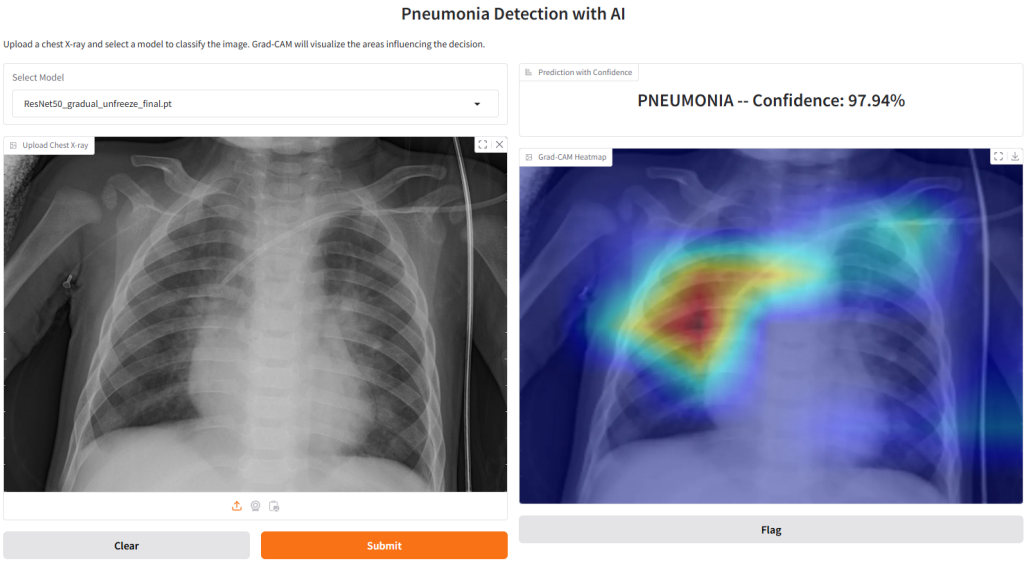
4.5 Scalability and Real-World Application

As artificial intelligence and deep learning models continue to be integrated into medical workflows, it is crucial to assess their scalability and real-world applicability. This study not only focuses on achieving high accuracy on pneumonia detection but also ensures that the developed models can be efficiently deployed in diverse clinical environments.

4.5.1 User Interface with Gradio

To demonstrate a Real-World Application, an interactive Gradio interface was developed (Figure 6). This interface allows users to upload chest X-ray images and receive pneumonia classification results along with a confidence rating. Additionally, it provides Grad-CAM visualizations, enabling users to understand the key regions in the X-ray that influenced the model's prediction. The interface simplifies interaction with the deep learning model, making it easier for healthcare professionals and researchers to leverage AI-driven diagnostics without requiring extensive technical knowledge. This implementation bridges the gap between complex machine learning models and real-world medical applications, ensuring seamless integration into clinical workflows.
The Gradio Interface can be executed via '`python interface.py`' inside the conda environment.

Figure 6: Gradio Interface



Source: Own representation.

4.5.2 Scalability and Infrastructure

- The model pipeline supports multiple architectures, making it adaptable to different computational resources. Pretrained models could be deployed on cloud-based systems, while additional models could be trained on remote hardware (e.g Google Colab).
- Pre-trained models and transfer learning techniques reduce the need for extensive datasets, allowing the system to be trained with limited but high-quality medical data.
- Hospitals with high computational demands can utilize cloud-based services such as AWS SageMaker, Google Cloud AI, or Microsoft Azure ML for model inference. This enables centralized model updates and continuous improvements through federated learning.

4.5.3 Integration with Hospital Systems

For seamless adoption, the model must integrate with existing Picture Archiving and Communication Systems (PACS), which store and manage medical imaging data. A practical integration strategy involves:

- Developing a RESTful API that allows PACS workstations to send X-ray images for automated classification.
- Ensure compatibility with existing radiology infrastructure.
- Embedding the model within Radiology Information Systems (RIS) to provide real-time decision support for radiologists.

The model's predictions could be displayed within radiology workstations, along with Grad-CAM heatmaps to enhance interpretability and support radiologists' decision-making.

4.5.4 Regulatory Considerations and Clinical Validation

To ensure real-world applicability, the deployment strategy must comply with medical AI regulatory standards such as:

- **FDA and CE Mark Approval:** Compliance with the US Food and Drug Administration (FDA) and European CE marking regulations is necessary for clinical deployment.
- **Bias and Fairness Testing:** Model performance should be validated across diverse demographic groups to ensure fairness and avoid biases in diagnosis.
- **Clinical Trials:** Before full-scale deployment, the model should undergo retrospective and prospective clinical validation studies in collaboration with hospitals.

4.5.5 Future Work: Scaling Beyond Pneumonia Detection

A scalable pipeline should allow for extensions beyond pneumonia detection. Future enhancements include:

- **Multi-class classification:** Expanding the model to detect multiple thoracic diseases, such as tuberculosis and lung cancer. To adapt the project for multiclass prediction, the output layer must be modified to accommodate multiple classes by changing the classifier. The loss function should be updated from binary cross-entropy to cross-entropy loss (nn.CrossEntropyLoss), which is suitable for categorical labels. Evaluation metrics such as accuracy, precision, recall, and F1-score need to be adjusted to multiclass mode.
- **Federated Learning for Privacy-Preserving Training:** Training on decentralized hospital datasets without transferring sensitive patient data.
- **Active Learning for Continuous Model Improvement:** Implementing a feedback loop where radiologists review AI-assisted diagnoses and provide corrections to refine model performance.

4.5.6 Real-World Deployment

Deploying deep learning models for pneumonia detection in clinical settings requires careful consideration of scalability, integration with existing healthcare infrastructure, inference efficiency, and regulatory compliance. This section outlines a proposed deployment roadmap for the developed classification pipeline.

- An easy-to-use Gradio interface has been developed to provide seamless interaction for medical professionals. Users can upload chest X-ray images, receive real-time classification results with confidence ratings, and view Grad-CAM visualizations that highlight critical areas influencing the model's decision.
- The model and interface can be deployed as a web-based application, making it accessible for remote diagnostics and telemedicine applications, particularly in under-resourced regions.
- The flexibility of the model allows for continuous updates with new datasets, ensuring that the system remains robust and adaptable to evolving medical imaging techniques.
- Hospitals with high computational demands can utilize cloud-based services such as AWS SageMaker, Google Cloud AI, or Microsoft Azure ML for model inference. This enables centralized model updates and continuous improvements through federated learning.

5 Implementation

This chapter outlines the technical implementation of the methodology described earlier. It includes details of the flexible classifier, data preprocessing, model training, evaluation, and tools used in the process.
Detailed technical documentation is available at:

https://szabadosa.github.io/pneumonia_detection/

5.1 Classifier Overview

The classifier (`./code/classifier.py`) is designed to be a flexible and modular backbone for training, validating, and testing multiple deep learning architectures. It is implemented using PyTorch Lightning, enabling scalability and streamlined integration of advanced features.

At the core of this design is the `PneumoniaClassifier` base class, which provides shared functionality and structure for different architectural paradigms. This base class is inherited by two specialized classes:

- `CNNPneumoniaClassifier`: Tailored for convolutional neural networks (CNNs) like ResNet, DenseNet, and EfficientNet, leveraging their ability to extract hierarchical spatial features from images.
- `ViTPneumoniaClassifier`: Designed for Vision Transformers (ViT), which rely on self-attention mechanisms to capture global relationships across an image.

This inheritance-based structure ensures that architectural differences are handled appropriately while maintaining a consistent interface for preprocessing, training, and evaluation.

5.1.1 Core Features

1. Modular Design The classifier supports a wide range of backbone architectures, including ResNet, DenseNet, EfficientNet, and Vision Transformers (ViTs). The modularity allows users to easily add new architectures by extending the backbone configuration and dynamically adjusting input and output layers.

2. Advanced Metrics The classifier incorporates key evaluation metrics using `torchmetrics`, including Accuracy, Precision, Recall (Sensitivity), Specificity and F1-Score.

3. Flexible Training Strategies The classifier supports both freezing and gradual unfreezing of layers for transfer learning:

- **Freezing Layers:** Initially freezes all pre-trained layers to retain generic features.
- **Gradual Unfreezing:** Allows controlled fine-tuning by incrementally unfreezing layers based on the configuration, improving stability and performance.

4. Custom Checkpointing and Logging The classifier tracks training progress, saving checkpoints that include:

- Metrics such as accuracy, loss, F1-score, Specificity and Recall
- Hyperparameters like learning rate and weight decay.
- Details of unfrozen layers during training.

This ensures thorough tracking for reproducibility and experimentation.

5. Data Handling The classifier integrates preprocessing steps, including resizing, normalization, and augmentations. These are dynamically adjusted based on the dataset configuration. For example:

```
train_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

5.1.2 Configuring and Using the Classifier

The classifier is configured via a dedicated configuration class. Below is an example configuration for using ResNet50 with gradual unfreezing:

```
config = Config(
    backbone_name="resnet50",
    transfer_learning=True,
    learning_rate=1e-4,
    batch_size=20,
    max_epochs=50,
    weight_decay=1e-4,
    dropout=0.2,
    num_workers=16,
    model_name="ResNet50_gradual_unfreeze",
    version="001",
    optimizer_name="sgd",
    use_class_weights=True,
    image_res=224,
    patience=10,
    gradually_unfreeze=True,
    unfreeze_interval=5,
    num_layers_to_unfreeze=2,
    frozen_lr=1e-6,
    unfrozen_lr=1e-5
)
```

To initialize and train the classifier:

```
# Initialize classifier
model = CNNPneumoniaClassifier(config)

# Set up device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = model.to(device)

# Begin training
trainer = pl.Trainer(max_epochs=config.max_epochs, gpus=1)
trainer.fit(model, train_loader, val_loader)
```

This modular design allows researchers to rapidly iterate over various architectures and training configurations, ensuring flexibility and adaptability to diverse medical imaging tasks.

5.1.3 Integration with PyTorch Lightning

Using PyTorch Lightning as the core framework, the classifier benefits from:

- Simplified management of training loops and logging.
- Built-in support for distributed training and mixed-precision.
- Easy integration of callbacks, such as learning rate schedulers and early stopping.

This combination of modularity and advanced capabilities ensures that the classifier meets the demands of both exploratory research and production-scale deployments.

5.2 Data Preprocessing

The dataset preprocessing pipeline involves organizing data, applying transformations, and generating lung segmentation masks.

5.2.1 Organizing and Balancing the Dataset

The raw dataset was organized into balanced training, validation, and test splits using the script `./code/organize_dataset.py`. This script ensures proportional representation of normal and pneumonia cases across all subsets, as shown in Table 2.

Table 2: Comparison of the number of samples in the raw dataset versus the reordered dataset.

Category	**Raw Dataset**	**Reordered Dataset**
train_NORMAL	1,341	1,108
train_PNEUMONIA	3,875	2,991
test_NORMAL	234	238
test_PNEUMONIA	390	641
val_NORMAL	8	237
val_PNEUMONIA	8	641

Source: Own representation

Due to the over representation of pneumonia cases additional class weights were implemented in the classifier to balance the classes:

```
def compute_class_weights(self):
    """
    Computes class weights to handle imbalanced datasets.
    Returns:
    torch.Tensor: Tensor of class weights.
    """
    # Count occurrences of each class
    class_counts = Counter(self.train_loader.dataset.labels)
    total_samples = sum(class_counts.values())  # Total number of samples
    # Compute weights: total samples divided by the number of samples for each class
    class_weights = [total_samples / class_counts[i] for i in range(len(class_counts))]
    # Convert to tensor
    return torch.tensor(class_weights, dtype=torch.float32).to(self.device)
```

5.2.2 Lung Segmentation

Lung segmentation was a critical preprocessing step aimed at isolating diagnostically relevant regions in chest X-ray images. The process involved the following steps:

1. Annotation of Images: A subset of 10 chest X-ray images was manually annotated using the `labelme` software. The lung regions were marked with polygons, and the annotations were saved in JSON format. Each annotation specified the lung boundaries as polylines, which served as the foundation for mask generation.

2. Dataset Creation: The notebook `./notebooks/lung_masks.ipynb` defines a LungSegmentation-Dataset class, which pairs annotated images with their corresponding masks. The class processes the JSON files to create binary masks, where lung regions are represented with pixel values of 1 and the background with 0. Masks were resized to a target resolution of 256×256 to standardize the input size.

3. Model Architecture: A U-Net model was implemented for segmentation, leveraging its encoder-decoder structure and skip connections for precise boundary detection. The U-Net comprises convolutional blocks for downsampling and upsampling, with transposed convolutions to recover spatial resolution.

4. Training the U-Net: The U-Net was trained using the annotated dataset. Binary cross-entropy (BCE) loss was used to optimize the model, and data augmentations (e.g., random flips and rotations) were applied to improve generalization. Training was conducted for 200 epochs with the Adam optimizer and a learning rate of 1×10^{-4}.

5. Generating Masks for the Full Dataset: Once trained, the U-Net model was applied to the full dataset to generate lung masks. The masks were premultiplied with the equalized images, effectively isolating lung regions and reducing noise from irrelevant areas.

Key Benefits:

- **Noise Reduction:** Segmentation eliminated non-lung areas, preventing the model from learning irrelevant features.
- **Improved Focus:** The classifier concentrated on diagnostically significant regions, enhancing accuracy and interpretability.

This segmentation process was crucial for enhancing the dataset quality, enabling more effective training of pneumonia detection models.

5.3 Model Training

The training framework was designed for flexibility, supporting multiple architectures such as ResNet, DenseNet, EfficientNet, and Vision Transformers.

The models were trained with the following parameters:

- **Learning Rate (1e-4):** A moderate learning rate was chosen to balance convergence speed and stability, ensuring gradual improvement in the loss without overshooting minima.
- **Batch Size (20):** Selected to optimize memory usage while providing sufficient gradient signal for effective learning, particularly on hardware with constrained GPU memory.
- **Maximum Epochs (50):** Allowed the models ample time to converge, considering the dataset size and transfer learning setup.
- **Weight Decay (1e-4):** Regularization was applied to reduce overfitting by penalizing large weights, promoting generalization.
- **Dropout (0.2):** Introduced to prevent co-adaptation of neurons and improve generalization, particularly in deeper models like ResNet50 and DenseNet131.

These parameters were carefully tuned to achieve a balance between computational efficiency and model performance. The use of transfer learning, combined with these hyperparameters, allowed the models to converge quickly while adapting effectively to the pneumonia detection task.

5.4 Unfreezing Layers.

All layers were initially unfrozen for end-to-end training. Gradual unfreezing was implemented for ResNet50 to evaluate its impact on performance. This strategy is encapsulated in the notebook `./notebooks/Resnet50_gradual_unfreeze.ipynb`.

```
def unfreeze_next_layers(self, num_layers_to_unfreeze=1):
    layers = list(self.feature_extractor.children())
    for i in range(self.currently_unfrozen,
                   self.currently_unfrozen + num_layers_to_unfreeze):
        for param in layers[i].parameters():
            param.requires_grad = True
    self.currently_unfrozen += num_layers_to_unfreeze
```

5.5 Custom Checkpointing

To track model metadata, a custom checkpointing system was developed using `./code/custom_checkpoint.py`. Metadata includes metrics, epoch details, and hyperparameters, ensuring detailed record-keeping.

```
metadata = {
    "epoch": trainer.current_epoch,
    "train_loss": trainer.callback_metrics.get("train_loss"),
    "val_loss": trainer.callback_metrics.get("val_loss"),
    "val_acc": trainer.callback_metrics.get("val_acc_epoch"),
}
```

5.6 Evaluation

5.6.1 Visualization Tools

Grad-CAM was used to interpret predictions and visualize regions influencing model decisions. Additionally every notebook utilizes the trained model to make random predictions on the test set to verify its results.

5.6.2 Metrics and Reporting

Metrics were logged using PyTorch Lightning, while evaluation results were visualized through confusion matrices, ROC-AUC curves, and Grad-CAM overlays.

5.7 Reproducibility and Scalability

The implementation was structured for reproducibility and scalability. Key features include:

- Modular design with reusable scripts and notebooks.
- Centralized folder management in `./code/project_globals.py`.
- Support for distributed training and mixed-precision using PyTorch Lightning.

6 Results

This chapter presents the results of the experiments conducted, evaluating the performance of the selected models on the pneumonia detection task. Key metrics such as accuracy, F1-score, precision, recall, and specificity are reported, along with insights from Grad-CAM visualizations and confusion matrix analyses.

6.1 Model Performance Metrics

The performance of the tested models (ResNet18, ResNet50, DenseNet131, EfficientNet, and Vision Transformer) was evaluated on the test set using the defined metrics. Table 3 summarizes the results.

Table 3: Model Testing Results Across Configurations

Configuration	Accuracy	F1-Score	Precision	Recall	Specificity
ResNet18_raw	0.7853	0.8521	0.7481	0.9897	0.4444
ResNet18_reordered	0.9477	0.9638	0.9730	0.9548	0.9286
ResNet18_reordered_weighted	0.9226	0.9451	0.9799	0.9126	0.9496
ResNet18_equalized	0.9124	0.9373	0.9796	0.8986	0.9496
ResNet18_premult	0.9124	0.9384	0.9638	0.9142	0.9076
ResNet18_upscale	0.8362	0.8750	0.9863	0.7863	0.9706
ResNet18_unfreeze	0.9238	0.9455	0.9881	0.9064	0.9706
ResNet18_upscale_unfreeze	0.8476	0.8849	0.9847	0.8034	0.9664
ResNet18_gradual_unfreeze	0.9511	0.9658	0.9838	0.9485	0.9580
ResNet50_premult	0.9295	0.9501	0.9817	0.9204	0.9538
ResNet50_unfreeze	0.9261	0.9475	0.9832	0.9142	0.9580
ResNet50_gradual_unfreeze	0.9681	0.9780	0.9842	0.9719	0.9580
DenseNet131_premult	0.9590	0.9714	0.9887	0.9548	0.9706
DenseNet131_unfreeze	0.9590	0.9716	0.9825	0.9610	0.9538
DenseNet131_gradual_unfreeze	0.9647	0.9756	0.9857	0.9657	0.9622
EfficientNet_premult	0.9204	0.9431	0.9847	0.9048	0.9622
EfficientNet_unfreeze	0.9352	0.9547	0.9725	0.9376	0.9286
EfficientNet_gradual_unfreeze	0.9625	0.9737	0.9951	0.9532	0.9874
vit-base-patch16-224	0.8850	0.9160	0.9804	0.8595	0.9537
vit-large-patch16-384	0.8998	0.9278	0.9775	0.8829	0.9453
vit-large-patch32-384	0.8850	0.9163	0.9770	0.8627	0.9453

Source: Own representation

6.1.1 Best Performing Model

Among the tested models, ResNet50 achieved the highest performance across all metrics, specifically with the focus on Specificity and Recall. It performed with accuracy of 96.81%, F1-Score of 97.80%, Precision of 98.42%, Recall of 97.19% and Specificity of 95.80%. The gradual unfreezing strategy

employed during its training likely contributed to its superior results. The final results can be found in `./notebooks/Resnet50_gradual_unfreeze.ipynb`

Choosing ResNet Over DenseNet: While both ResNet50 and DenseNet131 demonstrated exceptional performance, ResNet50 was selected as the best-performing model due to the following considerations:

- **Higher Specificity and Recall:** ResNet50 with gradual unfreezing achieved slightly higher recall (97.19%) compared to DenseNet131 (96.57%), ensuring fewer missed pneumonia cases. Additionally, ResNet50 provided comparable specificity (95.80%), balancing the reduction of false positives with accurate classification of negative cases.
- **Training Stability:** ResNet50 exhibited smoother training dynamics during gradual unfreezing, with consistent improvements across epochs for recall and specificity.
- **Computational Efficiency:** ResNet50 requires fewer parameters compared to DenseNet131, resulting in faster training and inference times, which are critical for clinical deployment.
- **Practical Scalability:** The modular and scalable architecture of ResNet50 made it easier to integrate into the flexible training framework, simplifying experimentation and deployment.

6.1.2 ResNet50 vs. Vision Transformers

Although Vision Transformers (ViTs) represent a modern approach to image classification, ResNet50 outperformed ViTs in this study due to several key factors:

1. Data Requirements: ViTs rely on self-attention mechanisms to model global dependencies within an image, which makes them highly effective for capturing long-range patterns. However, this architecture inherently requires large datasets to achieve optimal performance. Despite leveraging transfer learning, the comparatively smaller dataset used in this study limited ViT's ability to generalize effectively.

2. Tested ViT Architectures: Three different ViT models were tested in this study:

- **ViT-Base-Patch16-224**
- **ViT-Large-Patch16-384**
- **ViT-Large-Patch32-384**

This study experimented with different patch sizes in an attempt to capture improtant features within the X-ray images, while ViT-Large-Patch16-384 showed the best performance due to its relatively small size against the image size.

3. Computational Efficiency: ResNet50 requires fewer computational resources than ViTs, particularly during fine-tuning. This efficiency enabled deeper exploration of hyperparameters, such as gradual unfreezing, which significantly boosted ResNet50's performance.

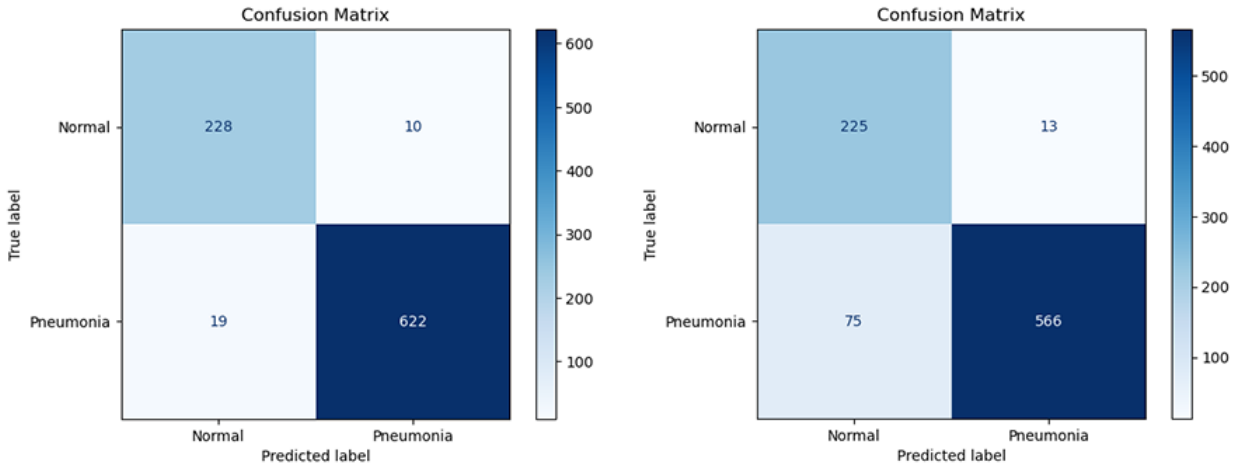
4. Performance on Key Metrics: For medical imaging, specificity and recall are critical to minimize false positives and false negatives. ResNet50 with gradual unfreezing achieved higher recall (97.19%) and comparable specificity (95.80%) compared to the best performing Vit (ViT-Large-Patch16-384) with recall at 88.28% and specificity at 94.53%. These metrics highlight ResNet50's superior reliability in identifying pneumonia cases and correctly classifying non-pneumonia cases.

While ViTs hold promise for tasks with large-scale datasets, ResNet50 proved to be the better choice for this study due to its adaptability, robustness, and ability to achieve high performance in specificity and recall on a smaller dataset.

6.2 Confusion Matrices

Confusion matrices were used to provide a detailed breakdown of the models' predictions. Figure 7 shows the confusion matrices for ResNet50 and vit-large-patch16-384.

Figure 7: Confusion Matrices for ResNet50_gradual_unfeeze and vit-large-patch16-384



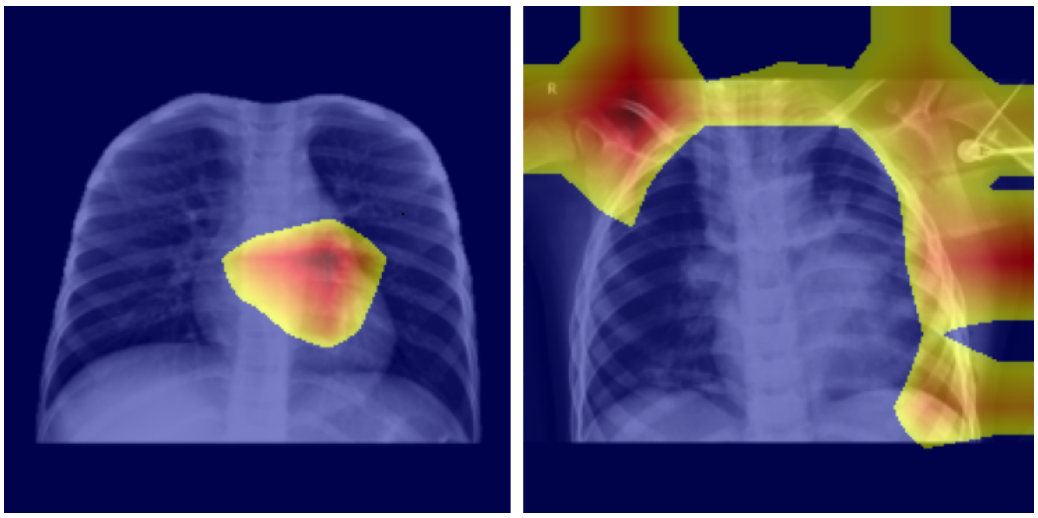
Source: Own representation.

While both models showed comparable performance on detecting normal cases, ResNet50 significantly outperformed the Visual Transfer in detecting Pneumonia cases.

6.3 Grad-CAM Visualizations

Grad-CAM visualizations were employed to interpret the model's predictions by identifying the regions of the input images most influential to the classification. Figure 8 provides an example of Grad-CAM outputs for both correctly and incorrectly focused models.

Figure 8: Grad-CAM Visualizations for ResNet50_gradual_unfeeze and ResNet18_equalized



Source: Own representation.

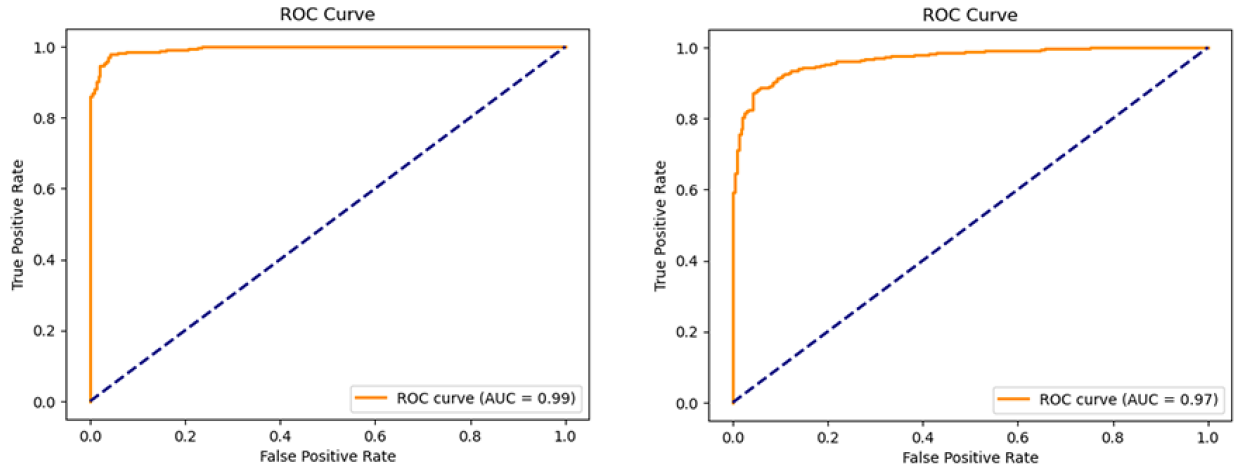
These visualizations demonstrate how the lung segmentation drastically improved the focus, avoiding overfitting to non relevant features like the shoulders in the unpremultiplied ResNet18 Experiment.

6.3.1 Grad-CAM Interpretation

6.4 ROC-AUC Analysis

Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) scores provide additional insights into the models' classification performance. Figure 9 presents the ROC curves for both ResNet50_gradual_unfeeze and vit-large-patch16-384.

Figure 9: ROC Curves for ResNet50_gradual_unfeeze and vit-large-patch16-384



Source: Own representation.

ResNet50 achieved the highest AUC score, further validating its robust performance in distinguishing between pneumonia and normal cases.

6.5 Observations and Insights

Key observations from the results include:

- ResNet50 outperformed other models, with gradual unfreezing significantly contributing to its success.
- Vision Transformers, while promising, underperformed relative to CNN-based models, likely due to the limited dataset size.
- Preprocessing techniques, particularly lung segmentation and histogram equalization, improved performance across all models.
- Grad-CAM visualizations confirmed that the models consistently focused on relevant regions, ensuring interpretability and reliability.

The results demonstrate the effectiveness of CNN-based models, particularly ResNet50, for pneumonia detection. The integration of preprocessing, fine-tuning strategies, and robust evaluation metrics ensured high performance and clinical relevance. These findings establish a strong foundation for future research and potential deployment in medical imaging workflows.

7 Conclusion

This study has demonstrated the viability of deep learning models for pneumonia detection, with a particular focus on creating a flexible pipeline capable of supporting diverse architectures. Through a systematic approach encompassing data preprocessing, model training, and comprehensive evaluation, the research has addressed key challenges in medical imaging, including limited datasets, class imbalances, and interpretability.

7.1 Key Findings

The experiments highlighted several important insights:

- **Model Performance:** ResNet50 emerged as the most effective model. The gradual unfreezing strategy employed for this model demonstrated its ability to adapt pre-trained networks effectively to domain-specific tasks.
- **Preprocessing Impact:** Techniques like histogram equalization and lung segmentation significantly improved model performance, ensuring the focus remained on diagnostically relevant regions.
- **Vision Transformers:** While promising, ViTs underperformed compared to CNN-based models, emphasizing the importance of large datasets and tailored training strategies for such architectures.
- **Interpretability:** Grad-CAM visualizations provided valuable insights into the models' decision-making processes, validating their clinical relevance by highlighting regions consistent with diagnostic features.

7.2 Contributions

This research makes several notable contributions:

- Development of a modular, flexible pipeline that accommodates multiple model architectures and supports iterative improvements.
- Integration of advanced preprocessing techniques, such as lung segmentation, to enhance feature extraction and reduce noise.
- Comprehensive evaluation framework incorporating metrics like accuracy, F1-score, precision, recall, specificity, ROC-AUC, and interpretability tools like Grad-CAM.

7.3 Limitations

Despite its successes, the study has some limitations:

- **Dataset Size:** The relatively small dataset constrained the performance of data-intensive models like ViTs.
- **Computational Requirements:** Advanced models required significant computational resources, limiting experimentation with more complex architectures or hyperparameter tuning.
- **Generalizability:** The pipeline's performance has been validated on a single dataset, and its applicability to other datasets or imaging modalities remains untested.

7.4 Future Work

Building on the findings of this study, future work can explore:

- **Expanding Dataset Size:** Leveraging larger datasets or synthetic data generation to improve the performance of data-hungry models like ViTs.
- **Hybrid Architectures:** Investigating combinations of CNNs and transformers to harness the strengths of both approaches.
- **Real-World Deployment:** Adapting the pipeline for deployment in clinical settings, including user-friendly interfaces and integration with existing workflows.
- **Broader Applications:** Extending the pipeline to detect other medical conditions or process multi-modal data, such as CT or MRI scans.

7.5 Final Remarks

The study underscores the transformative potential of deep learning in medical imaging, particularly when combined with thoughtful preprocessing and interpretability tools. By focusing on modularity, flexibility, and comprehensive evaluation, this research lays a solid foundation for future advancements in automated pneumonia detection and beyond.

Bibliography

Hasse, F./Leiser, F./ Sunyaev, A.: *Informed machine learning for cardiomegaly detection in chest x-rays: A comparative study*. In: *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, IEEE (**2024**), 1–5.

Mahr, F./Schmidt, K./Thielen, N./Sindel, T./ Franke, J.: *Optimizing machine learning performance via dataset generation for x-ray image classification*. In: *2024 25th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, IEEE (**2024**), 1–6.

Manzari, O. N./Ahmadabadi, H./Kashiani, H./Shokouhi, S. B./ Ayatollahi, A. (**2023**): *MedViT: A Robust Vision Transformer for Generalized Medical Image Classification*. Jg. 157, 106791.

Minaee, S./Kafieh, R./Sonka, M./Yazdani, S./ Soufi, G. J. (**2020**): *Deep-COVID: Predicting COVID-19 From Chest X-Ray Images Using Deep Transfer Learning*.

Niu, S./Liu, M./Liu, Y./Wang, J./ Song, H. (**2020**): *Distant Domain Transfer Learning for Medical Imaging*.

Peng, L./Liang, H./Luo, G./Li, T./ Sun, J. (**2021**): *Rethinking Transfer Learning for Medical Image Classification*.

Roy, A./Bhattacharjee, A./Oliva, D./Ramos-Soto, O./Alvarez-Padilla, F. J./ Sarkar, R. (**2024**): *FA-Net: A Fuzzy Attention-aided Deep Neural Network for Pneumonia Detection in Chest X-Rays*.

Sharma, S. V., P.: *Classification of covid-19 utilizing ct scan images employing the efficientnet model*. In: *2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)* (**2024**), 1–7.

Siddiqi/Raheel, J./ Sameena (**2024**): *Deep Learning for Pneumonia Detection in Chest X-ray Images: A Comprehensive Survey*. Jg. 10, 176.

VinLab (**2025**): *Convolutional neural networks overview: The heart of deep learning algorithms*. Accessed: 2025-01-11.

Zhuang, Y./ et al. (**2022**): *An Interpretable Multi-task System for Clinically Applicable COVID-19 Diagnosis Using CXR*. 847–862.

Çelika/Ahmet, D./ Selim (**2023**): *Enhanced Pneumonia Diagnosis Using Chest X-Ray Image Features and Multilayer Perceptron and k-NN Machine Learning Algorithms*. Jg. 40, 1015–1023.