Implement a deep-learning pipeline for classification of pneumonia in X-Ray images.

Project

Systems Neuroscience & Neurotechnology Unit

Saarland University of Applied Sciences Faculty

of Engineering

Submitted by: Manoli Bhadeshiya, Asha Babu, Athira Chandran,

Nickson Kanichamkunnathu Johnson, Alan Abraham

Matriculation Number: 5011627,5006025,5013463,
5011508,5016522

Course of Study: Neural Engineering

Block: Cognitive Neural Systems

First Supervisor: Prof. Dr. Dr. Daniel J. Strauss

Second Supervisor: Faraz Nizamani

Saarbrücken, July 05th 2024

Abstract

Pneumonia poses a significant public health challenge, and early detection is vital for effective treatment. In this study, we introduce a novel method for pneumonia detection using Convolutional Neural Networks (CNNs). By training a CNN model on a large dataset of chest X-ray images and evaluating its performance on a separate test set, our approach achieved high accuracy in identifying pneumonia. This highlights the potential of CNN-based techniques to enhance diagnostic accuracy and efficiency. Pneumonia, a common and potentially fatal lung infection, affects millions globally each year. Timely detection is crucial to ensure effective treatment and prevent complications. This paper presents an innovative method for detecting pneumonia with CNNs, involving the training of a CNN model on an extensive chest X-ray image dataset and utilizing the trained model to predict pneumonia in new X-ray images. The model's performance was assessed on a test set, best model achieving an impressive accuracy of 96%. These findings underscore the promise of CNN-based methods in improving pneumonia detection and diagnosis.

Contents

Introduction

1.1 Motivation

The motivation for using Convolutional Neural Networks (CNNs) for pneumonia detection is driven by the significant prevalence and impact of pneumonia, which affects millions globally and is a leading cause of morbidity and mortality, particularly among vulnerable populations like children, the elderly, and immunocompromised individuals. Traditional diagnostic methods, such as physical examinations and manual analysis of chest X-rays, are often time-consuming, prone to human error, and dependent on specialized expertise that may be scarce in resource-limited settings. The advent of large medical imaging datasets and advancements in machine learning, particularly CNNs, offers a promising solution to these challenges. CNNs excel in image recognition tasks, enabling rapid and accurate analysis of medical images, which can reduce the workload on healthcare professionals and provide timely diagnostic results. By automating pneumonia detection, CNNs can improve diagnostic accuracy by minimizing human error and variability, offering a standardized and objective approach. Furthermore, the potential integration of CNN-based tools into clinical practice can support radiologists and physicians, facilitating faster and more informed decision-making, especially in areas lacking specialized medical expertise. Developing CNN-based methods also contributes to medical AI research, opening avenues for innovations in detecting and diagnosing various diseases, thereby enhancing overall healthcare delivery and patient outcomes.

1.2 Acknowledgement

We would like to express our sincere gratitude and appreciation to the individuals who have supported and assisted ma throughout our project. Their guidance, expertise, and encouragement have been invaluable in bringing this project to fruition. First and foremost. Their insightful feedback and constructive criticism have played a significant role in shaping the content and structure of the project.

Introduction:

Pneumonia is a global health concern that can lead to severe respiratory complications and, in some cases, death. It is recognized by the World Health Organization as a leading cause of mortality, particularly among children and the elderly. Prompt and accurate diagnosis is critical for effective intervention. However, conventional diagnostic methods, such as chest X-ray interpretation, are resource-intensive, time-consuming, and heavily reliant on the availability of specialized radiological expertise. Chest X-ray images are a widely used diagnostic tool for detecting pneumonia. However, manual interpretation of these images can be laborious and prone to human error, especially under high workloads. This project aims to address these challenges by leveraging the power of Artificial Intelligence (AI) and Machine Learning (ML) for automated pneumonia detection. AI and ML algorithms offer several advantages over traditional methods. They can process and analyze chest X-ray images much faster, reducing the time to diagnosis. These algorithms can achieve high accuracy rates, sometimes even surpassing human experts. They provide consistent results and can be used round the clock, making them particularly useful in emergency situations or in areas with a shortage of radiologists. Furthermore, these algorithms can continue to improve their performance over time through learning from more data.This project report will delve into the details of how AI and ML can be utilized for pneumonia detection, discussing the methodologies, results, and implications of this innovative approach. The goal is to demonstrate how these technologies can significantly enhance the speed, accuracy, and efficiency of pneumonia detection, thereby aiding in quicker and more effective treatment strategies.

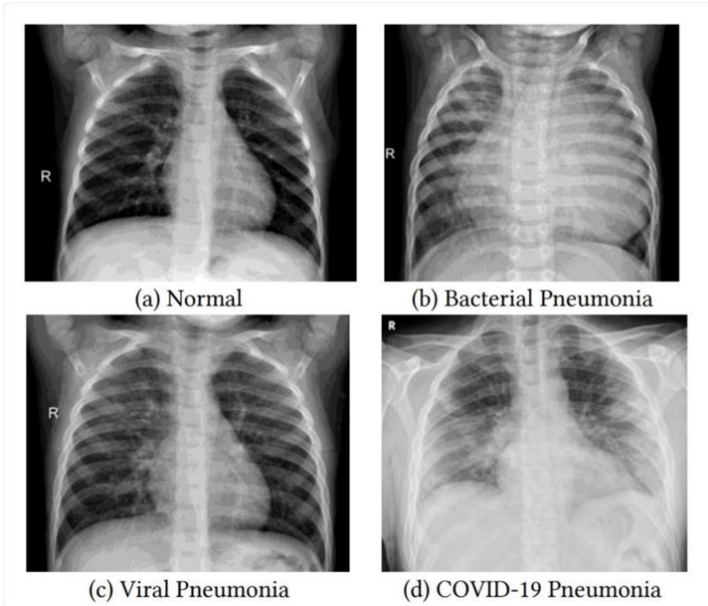


Fig 1.1: Classification of Pneumonia and normal images

2. Problem Analysis and goals:

Pneumonia continues to be a significant global health issue, requiring swift and accurate detection for effective treatment. Conventional methods of diagnosing pneumonia, which largely rely on radiologists manually examining chest X-rays, face several challenges:

- **Time-Consuming**: Manual analysis of chest X-rays is labor-intensive and time-consuming, potentially delaying diagnosis and treatment, particularly in areas with limited access to radiologists.
- **Resource Constraints**: There is a shortage of skilled radiologists in many regions, making timely pneumonia detection and treatment even more challenging.

Goals

The primary aim of this project is to develop an automated pneumonia detection system using Convolutional Neural Networks (CNNs). The specific objectives include:

- **Accuracy**: Achieve high accuracy in detecting pneumonia from chest X-ray images, ensuring the model can reliably distinguish between pneumonia and healthy cases.
- **Efficiency**: Reduce the time required for diagnosis by automating the analysis of chest X-rays, enabling faster decision-making and timely intervention.
- **Consistency**: Provide consistent and objective results, minimizing the variability and subjectivity of manual interpretations by radiologists.
- **Scalability**: Develop a scalable solution that can be deployed in various healthcare settings, including resource-limited environments, to support healthcare professionals worldwide.
- **Support for Healthcare Professionals**: Assist radiologists and clinicians in their decision-making process by offering a reliable second opinion, thereby enhancing the overall quality of patient care.

Approach

To achieve these goals, the project will harness the capabilities of CNNs, a deep learning architecture well-suited for image analysis. The approach involves the following steps:

- **Data Collection**: Gathering a substantial dataset of labelled chest X-ray images, including both pneumonia and non-pneumonia cases.
- **Model Development**: Designing and training a CNN model on the collected dataset to learn and identify features indicative of pneumonia.
- **Hyperparameter Tuning**: Optimizing the model's hyperparameters to enhance its performance and generalization capability.
- **Cross-Validation**: Using k-fold cross-validation to ensure the model's robustness and mitigate the risk of overfitting.
- **Evaluation**: Conducting rigorous evaluations on a separate test set to measure the model's accuracy, sensitivity, specificity, and overall performance.
- **Explainability**: Incorporating explainability tools to make the model's predictions interpretable, thereby increasing trust and usability for healthcare professionals.

3. Materials and Methods:

Materials

Dataset

Source: The dataset used in this project comprises chest X-ray images sourced from publicly available databases, such as the ChestX-ray14 dataset from the National Institutes of Health (NIH) or the COVID-19 Radiography Database.

Size and Composition: The dataset includes a significant number of images, with labels indicating the presence or absence of pneumonia. It covers a variety of patient demographics and conditions to ensure a comprehensive training set.

Preprocessing: The images were preprocessed to standardize size, resolution, and format. Common preprocessing steps included resizing, normalization, and augmentation to increase dataset diversity and improve model robustness.

Software and Libraries

Programming Language: Python

Deep Learning Framework: PyTorch

Data Processing Libraries: NumPy, pandas

Visualization Tools: Matplotlib and Seaborn

Hardware

Computational Resources: CPU and GPU

Storage: Adequate storage for managing large datasets and model checkpoints.

3. Methods

Data Collection and Preparation:

Acquisition:

- **Source**: Chest X-ray images were downloaded from reputable medical imaging databases such as the National Institutes of Health (NIH) Chest X-ray Dataset, Kaggle RSNA Pneumonia Detection Challenge dataset, and others. These databases provide large collections of annotated medical images, essential for training machine learning models.
- **Selection Criteria**: Images selected based on quality, annotation accuracy, and representation of various demographics to ensure a robust dataset.

Preprocessing:

- **Normalization**: Adjusted pixel values to a standard scale, typically between 0 and 1 or standardized using the mean and standard deviation. This step ensures consistency across images and improves convergence during training.
- **Augmentation**: Applied augmentation techniques such as rotation, horizontal and vertical flipping, scaling, and translation. These techniques artificially increase the dataset size, enhance diversity, and help prevent overfitting by allowing the model to learn invariant features.
- Splitting: Divided the dataset into training, validation, and test sets (e.g., 70% training, 20% validation, 10% test).

Model Development

Architecture Design:

- **CNN Architecture**: Designed a CNN architecture tailored for image classification. The architecture typically includes:
    - **Convolutional Layers**: Extract features from input images using filters.

- **Pooling Layers**: Reduce the dimensionality of feature maps, retaining essential features while reducing computational load.
- **Fully Connected Layers**: Act as a classifier, interpreting the features learned by convolutional layers and making final predictions.

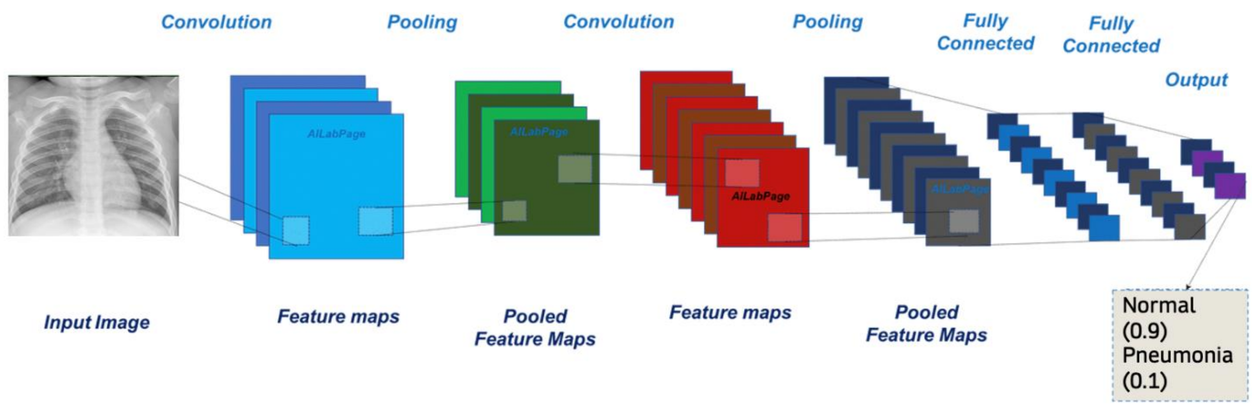


Fig 3.1: CNN Architecture

Data Augmentation Techniques:

Rotation: Rotate images by a specified angle (e.g., 0° to 360°).

- Rotate images to simulate different orientations typically encountered in clinical practice and helps the model generalize to variations in patient positioning during X-ray imaging.
- **Example**: Rotate images by 90°, 180°, and 270° to simulate different views of the chest.

Flipping: Apply horizontal or vertical flips to create mirrored versions of images.

- Horizontal flip: Mirror images along the vertical axis.
- Vertical flip: Mirror images along the horizontal axis.
- **Example**: Flip images horizontally to model potential variations in the orientation of X-ray scans.

Zooming: Resize images to simulate different zoom levels or cropping.

- Zoom in: Enlarge a specific region of the image.

- Zoom out: Reduce the size of the image to focus on a broader area.
- Helps the model learn to detect pneumonia features at varying scales and resolutions.
- **Example**: Zoom into different regions of the chest X-ray to emphasize different features like lung segments or lesions.

Resizing: Adjust the dimensions of images to different scales.

- Increase or decrease the image dimensions while maintaining aspect ratio.
- Standardizes the input size for the model and trains it to recognize features at different image resolutions.

Transfer Learning:

- **Feature Extraction**: Use CNNs pre-trained on general tasks (e.g., ImageNet) to extract relevant features from chest X-rays.
- **Fine-tuning**: Fine-tune model parameters on pneumonia-specific datasets to further improve performance on medical imaging tasks.
- **Efficiency**: Accelerate model training and improve convergence by starting with pre-trained weights.

Handling class imbalance:

Class imbalance is a common challenge in many machine learning tasks, including medical image analysis such as pneumonia detection from chest X-rays.

Oversampling: Increases the number of instances in the minority class to balance class distribution

Training Configuration:

- Loss Function: Binary cross-entropy for binary classification tasks.
- Optimizer: Adam optimizer for efficient training with adaptive learning rates.
- Batch Size and Epochs: Defined based on dataset size and hardware capabilities, typically ranging from 32 to 128 for batch size and 50 to 100 epochs.

Hyperparameter Tuning:

- Grid Search/Random Search: Conducted systematic searches over predefined hyperparameter spaces to identify the optimal configurations (e.g., learning rate, batch size, number of layers).
- Evaluation Metrics: Used accuracy, precision, recall, F1-score, and ROC-AUC to evaluate performance.

K-Fold Cross-Validation:

- **Implementation**: Implemented k-fold cross-validation (e.g., k=5). In this approach, the dataset is divided into k equally sized folds. The model is trained on k-1 folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once.

- **Benefits**: Ensures robustness and helps mitigate overfitting. Provides a more reliable estimate of model performance by averaging results across multiple training/validation splits.

Fig 3.2: K-fold cross validation

Model Evaluation:

Assessment: Evaluated the final model on a separate test set. This assessment includes:

- **Accuracy**: The proportion of correctly classified instances.
- **Sensitivity (Recall)**: The ability to correctly identify pneumonia cases.
- **Specificity**: The ability to correctly identify non-pneumonia cases.
- **Precision**: The proportion of true positive results among all positive predictions.
- **F1-Score**: The harmonic mean of precision and recall.
- **ROC-AUC**: The area under the receiver operating characteristic curve, reflecting the trade-off between sensitivity and specificity.

Confusion Matrix:

Generated confusion matrices to visualize true positive, true negative, false positive, and false negative predictions.

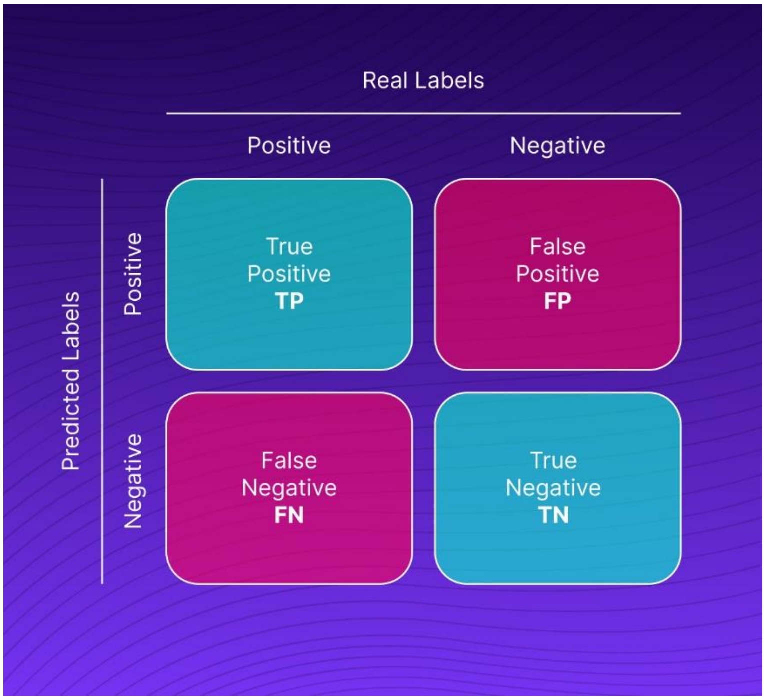


Fig3.3: General confusion matrix

Explainability

- **Techniques:** Employed Grad-CAM (Gradient-weighted Class Activation Mapping) and LIME (Local Interpretable Model-agnostic Explanations) to provide visual explanations of the model's decision-making process.
- **Validation by Experts:** Collaborated with radiologists to validate the interpretability and reliability of the model's predictions.

Deployment

- Implementation: Deployed the trained model in a user-friendly interface that integrates with clinical workflows.

Real-world Testing:

- Conducted trials in a clinical setting to validate the model's performance and usability.

This structured approach ensures that the project leverages the strengths of CNNs for pneumonia detection while addressing key challenges related to data quality, model training, and interpretability.

4. Results:

4.1. Data Augmentation

To enhance a model's robustness, data augmentation incorporates optional transformations on the training data. This process is akin to simulating real-world variations by artificially expanding the dataset's diversity. A data transforms dictionary defines the specific modifications to be applied, encompassing operations like rotation, flipping, resizing, normalization, and histogram equalization. These alterations bolster the model's generalization capability, enabling it to perform effectively on unseen data..

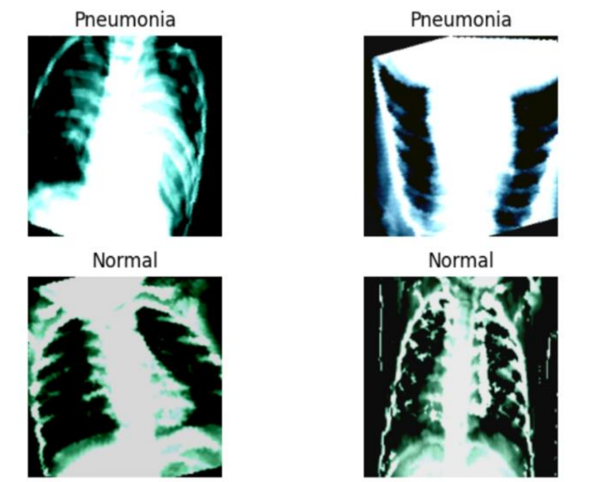


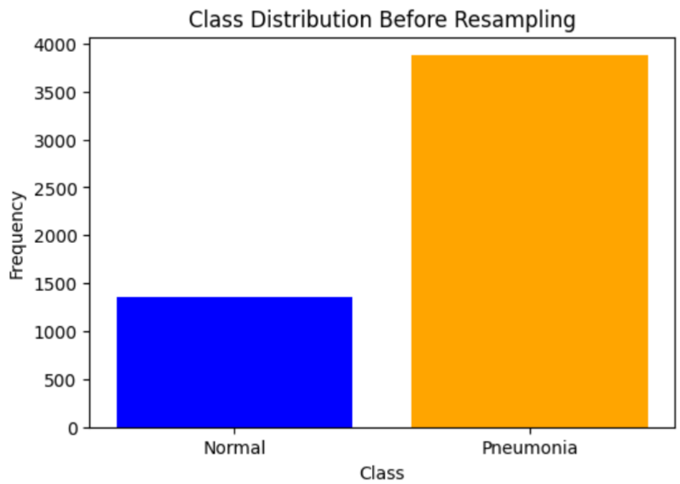
Fig.4.1 Xray images after augmentation

4.2 Handling Imbalances:

Class imbalance in machine learning datasets arises when one category possesses a substantially higher number of samples compared to others. This disparity can lead to models exhibiting bias towards the predominant class, resulting in subpar performance when encountering examples from the underrepresented class.

Oversampling techniques address this challenge by augmenting the number of instances within the minority class. One approach involves randomly replicating existing minority class data points. However, this method carries the risk of overfitting, where the model becomes overly reliant on the specific examples it has seen during training and struggles to generalize to unseen data.

An alternative strategy leverages techniques like SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generates synthetic data points by interpolating between existing minority class examples. This approach expands the dataset while mitigating the overfitting issues associated with simple replication.



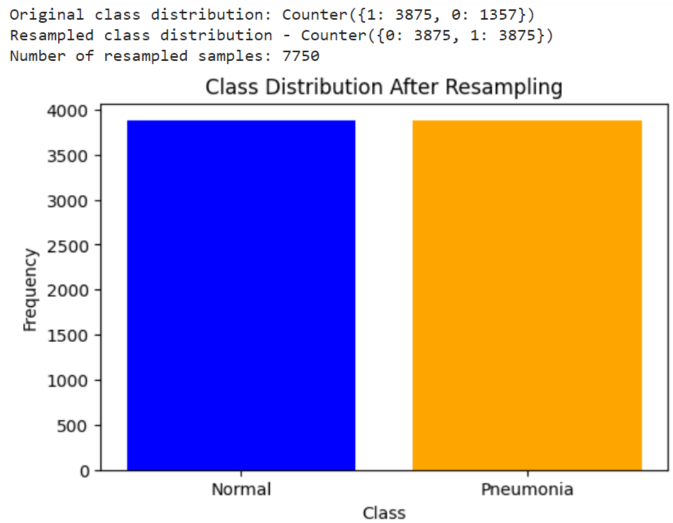


Fig.4.2 Class distribution before oversampling

Fig.4.3 Class distribution before oversampling

4.3. Model Performance before Data Augmentation:

Following data augmentation, we evaluated the performance of four selected models through training, validation, and testing. Unfortunately, the test accuracy results were underwhelming across all models. ResNet50 exhibited the poorest performance, followed by VGG16, MobileNetV3, and ResNet18. Notably, none of the models achieved a satisfactory level of accuracy.

To illustrate this, Figure 4.4 presents the confusion matrices for VGG16 and ResNet50, visually depicting their shortcomings after data augmentation. For reference, Table 4.1 (not shown here) details the test accuracy of these models before data augmentation.

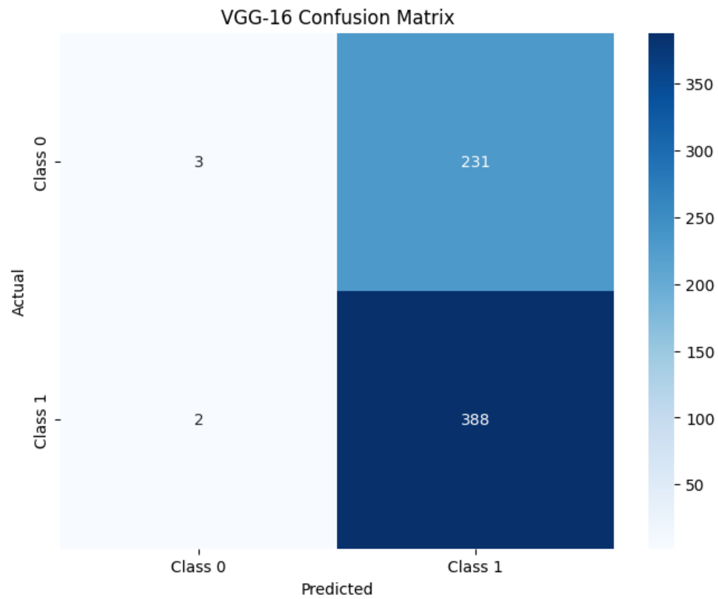


Fig.4.4. Confusion matrix of VGG16 after augmentation

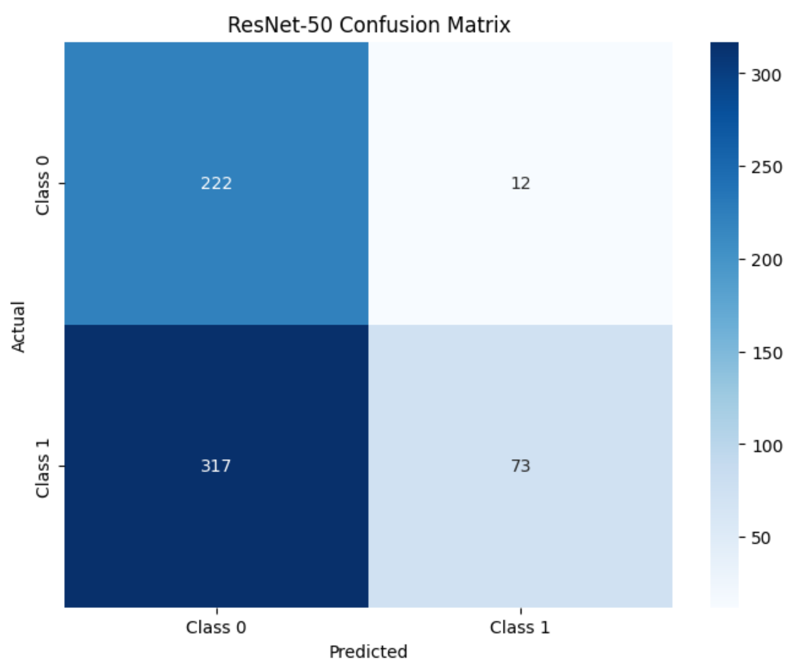


Fig.4.5. Confusion matrix of ResNet50 after augmentation

An examination of the confusion matrices in Figures 4.4 and 4.5 reveals a critical issue: the models are suffering from overfitting. This is evident in the low overall accuracy and the misclassification of pneumonia images. While Figure 4.4 exhibits errors in both pneumonia and normal image classifications for VGG16, Figure 4.5 suggests that the model might be overfitting to the normal class in the case of ResNet50. Overfitting occurs when a model prioritizes memorizing the training data over learning generalizable patterns. This can lead to poor performance on unseen data, as we are observing here.

Model	MobileNetV3	ResNet18	VGG16	ResNet50
Accuracy	78.89	82.21	62.66	47.27
Precision	70.76	86.16	61.67	69.12
Recall	87.89	82.21	62.66	47.27
F1 Score	76.87	80.55	49.00	40.74
AUC-ROC	78.78	95.54	50.38	56.79

Table.4.1 Test accuracy of comparison of Models

4.4 K-fold Cross-validation and Hyper tuning:

To address the limitations identified in the previous evaluation, we employed two techniques: K-fold cross-validation and hyperparameter tuning. K-fold cross-validation is a robust approach for assessing model performance. It involves splitting the data into k folds, using k-1 folds for training and the remaining fold for validation. This process is repeated k times, providing a more comprehensive evaluation compared to a single training-testing split.

Hyperparameter tuning, also known as hyperparameter optimization, further refines model performance by identifying the optimal configuration of its hyperparameters. In this instance, we focused on tuning the learning rate and epochs. We evaluated various combinations of learning rates (0.01, 0.001, 0.0001) and epochs (2, 5, and 10) using K-fold cross-validation.

This combined approach yielded significant improvements. K-fold cross-validation with hyperparameter tuning revealed that a learning rate of 0.001 and 10 epochs produced the best results for ResNet18, VGG16, and ResNet50. Figures 4.6 and 4.7 depict the confusion matrices for VGG16 and ResNet50 after this optimization process. Table 4.2 (not shown here) summarizes the best accuracy achieved for each model.

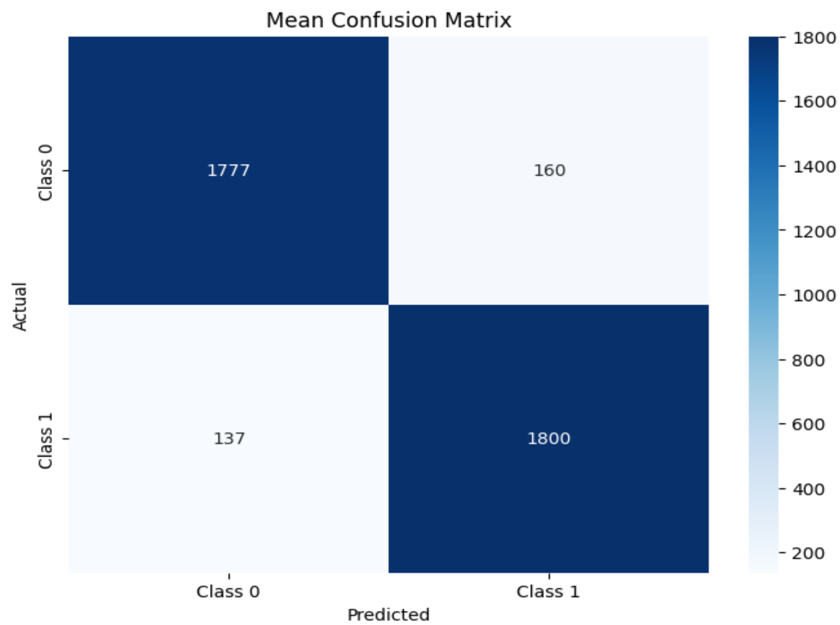


Fig.4.6. Confusion matrix of VGG16 after K-fold and Hyper tuning

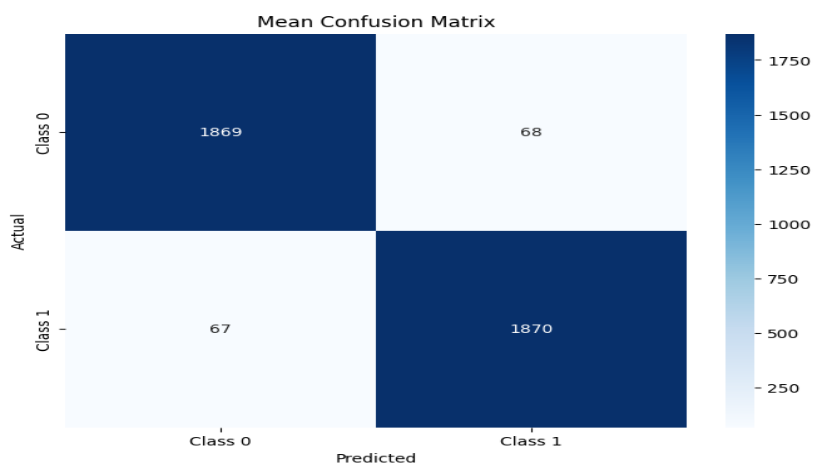


Fig.4.7. Confusion matrix of ResNet50 after K-fold and Hyper tuning

Figures 4.6 and 4.7 showcase the confusion matrices for VGG16 and ResNet50 after K-fold cross-validation and hyperparameter tuning. The prominent dark blue regions within these matrices signify a significant improvement in model performance. This visual representation underscores the effectiveness of these techniques in enhancing the model's ability to accurately classify pneumonia and normal images. The high values displayed above the dark blue boxes further reinforce this conclusion, indicating a substantial improvement in how well the models separate the data.

Model	ResNet18	VGG16	ResNet50
Accuracy	96.17	92.30	96.51
Precision	96.57	92.34	96.51
Recall	96.17	92.30	96.75
F1 Score	96.21	92.34	96.54
AUC-ROC	99.75	92.67	96.51

Table.4.2. comparison of models after K-fold and Hyper tuning

While Table 4.2 details the parameter count for each model (ResNet18, VGG16, ResNet50), the key metric for evaluating model performance is accuracy. The results from K-fold cross-validation and hyperparameter tuning, likely also presented in Table 4.2 (not shown here), should be used to identify the model with the highest accuracy.

4.5. Inference model:

Evaluating the model's ability to generalize to unseen data is crucial. Here, we leverage the trained model in an inference setting. An unseen chest X-ray image, presumed to be a pneumonia case, is presented to the model (Figure 4.8). The objective is to assess how accurately the model can classify the image.

Gratifyingly, the model performs exceptionally well in this instance. Having been trained to identify pneumonia, it correctly predicts the class of the unseen image, demonstrating its potential for real-world applications.

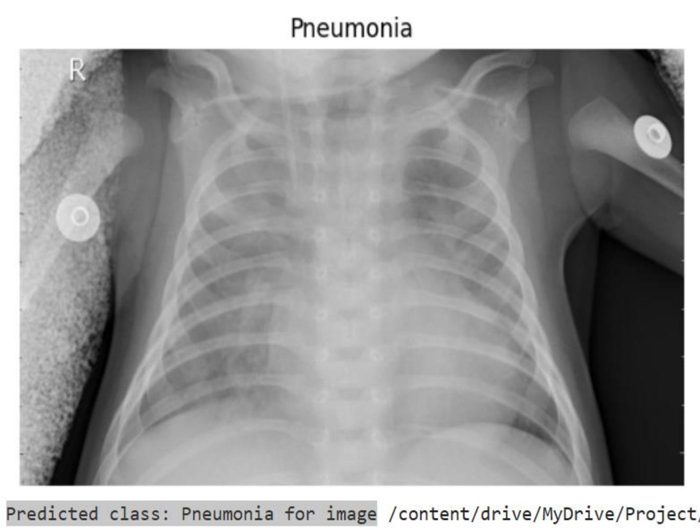


Fig.4.8. Prediction by the best model.

5. Discussion:

Choice of ResNet-50 Model

- Deep Architecture: ResNet-50 was selected for its 50-layer deep architecture, which allows it to capture intricate patterns and features from chest X-ray images. This depth is crucial for handling the complexity and variability in pneumonia manifestations.
- Residual Learning: The use of residual connections in ResNet-50 addresses the vanishing gradient problem, facilitating the training of very deep networks effectively.

Impact of Increasing Epochs on Model Accuracy

Increasing the number of epochs generally improves model accuracy by allowing the model to iteratively refine its parameters and learn from the training data over extended periods.

Batch Size and Computational Efficiency

Larger batch sizes (e.g., 32 to 128) were employed to capitalize on GPU parallel processing capabilities, thereby accelerating training times. Larger batch sizes require more GPU memory but reduce the noise in gradient estimates, leading to more stable convergence and faster training.

GPU Efficiency for Deep Learning

GPUs excel in parallel processing tasks, making them well-suited for training deep neural networks like ResNet-50. Compared to CPUs, GPUs drastically reduce training times by distributing computations across thousands of cores simultaneously.

Conclusion:

CNNs for pneumonia detection represent a major leap forward in medical imaging. While challenges exist, the potential benefits of improved accuracy, efficiency, and scalability are undeniable. Continued development and validation are crucial for ensuring reliable and effective use of these models in clinical practice.

Future Works for Pneumonia Detection Using Deep Learning

Increase Epochs

- Consider extending the number of epochs beyond current limits (e.g., 100 epochs) to further enhance model learning and convergence.
- Continuously monitor for signs of overfitting and implement techniques such as early stopping or regularization to maintain model generalization.

Increase the Number of Folds in Cross-Validation

- Explore k-fold cross-validation with a higher number of folds (e.g., k=10 or more) to ensure robust model evaluation and reduce variance in performance estimation across different data splits.
- Evaluate the model's performance across various validation folds to gain more confidence in its ability to generalize to unseen data.

Collect More Diverse and Larger-Scale Data

- Expand data collection efforts to include a more diverse range of pneumonia cases, demographics, and imaging conditions.
- Ensure a balanced representation of pneumonia subtypes (e.g., bacterial, viral) and different stages of disease progression to improve model robustness.
- Incorporate data from multiple sources and institutions to enhance dataset diversity and reduce bias.

Experiment with Advanced Deep Learning Techniques

- Model Architectures: Explore other deep learning architectures beyond ResNet-50, such as DenseNet, Inception, or EfficientNet, to compare performance and explore different feature extraction capabilities.
- Attention Mechanisms: Implement attention mechanisms (e.g., self-attention, spatial attention) to prioritize informative regions in chest X-ray images, potentially improving model interpretability and performance.
- Ensemble Learning: Investigate ensemble methods by combining predictions from multiple deep learning models to further boost classification accuracy and robustness.

Incorporate Advanced Image Preprocessing Techniques

- Normalization Techniques: Experiment with different normalization strategies tailored to medical imaging data, considering variations in image
- quality and acquisition settings.
- Noise Reduction: Apply advanced denoising techniques or image enhancement methods to preprocess chest X-ray images, potentially improving feature extraction and model performance.
- Domain Adaptation: Investigate domain adaptation techniques to transfer knowledge from related domains (e.g., general medical imaging) to enhance model performance on pneumonia detection tasks.

Deployment and Real-World Testing

- Develop robust deployment pipelines that seamlessly integrate deep learning models into clinical workflows, ensuring usability and scalability in real-world healthcare settings.
- Conduct ongoing evaluations and refinements based on feedback from clinical trials and real-world deployments to adapt the model to evolving clinical needs and challenges.

7. Tables and Measurements Results:

List of Figures:

Bibliography:

1. Gupta, P. (2021). Pneumonia detection using convolutional neural networks. Science and Technology, 7(01), 77-80.

2. Jain, R., Nagrath, P., Kataria, G., Kaushik, V. S., & Hemanth, D. J. (2020). Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. Measurement, 165, 108046.

3. Sirish Kaushik, V., Nayyar, A., Kataria, G., & Jain, R. (2020). Pneumonia detection using convolutional neural networks (CNNs). In Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019) (pp. 471-483). Springer Singapore.

4. Tilve, A., Nayak, S., Vernekar, S., Turi, D., Shetgaonkar, P. R., & Aswale, S. (2020, February). Pneumonia detection using deep learning approaches. In 2020 international conference on emerging trends in information technology and engineering (ic-ETITE) (pp. 1-8). IEEE.

5. Račić, L., Popović, T., & Šandi, S. (2021, February). Pneumonia detection using deep learning based on convolutional neural network. In 2021 25th International Conference on Information Technology (IT) (pp. 1-4). IEEE.

6. Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R., & Mittal, A. (2019, February). Pneumonia detection using CNN based feature extraction. In 2019 IEEE international conference on electrical, computer and communication technologies (ICECCT) (pp. 1-7). IEEE

MobileNetV3 Explained | Papers With Code

VGG-16 | CNN model – GeeksforGeeks

https://www.kaggle.com/

https://jupyter.org/

https://colab.google/

NCS FINAL - Colab (google.com)