Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Project Report
on
"Lung Disease Detection"

[Code No.: COMP-207]
(For partial fulfillment of II Year/II Semester in Computer Engineering)

Submitted by

Abhyudit Adhikari (2)
Aavash Lamichhane (29)
Aayush Pokhrel (42)
Saumya Shrestha (51)

Submitted to

Mr. Dhiraj Shrestha
Department of Computer Science and Engineering

Submission Date: 1st January 2024

Bona fide Certificate

This project work on

"Lung Disease Detection"

is the bona fide work of

Abhyudit Adhikari

Aavash Lamichhane

Aayush Pokhrel

Saumya Shrestha

who carried out the project work under my supervision.

Project Supervisor

____

Mr. Santosh Khanal
Assistant Professor
Department of Computer Science and Engineering

Date:

Abstract

In response to the escalating global challenge of detecting lung infections, this project harnessed the capabilities of machine learning to develop an automated system. Utilizing Python as the foundational language and TensorFlow for model development, the initiative aimed to analyze chest X-ray images efficiently. ReactJS facilitated seamless frontend integration, ensuring clinicians can easily interpret results. The anticipated outcome is a robust diagnostic tool that offers high accuracy in identifying diverse lung infections, bridging the gap between medical imaging and artificial intelligence. Continuous refinement through extensive datasets and clinical validations is recommended to bolster its reliability and efficacy.

Keywords: Machine Learning, Chest X-rays, TensorFlow, Python, ReactJS, Lung Infections, Diagnostic Tool.

Table of Contents

List of Figures

Abbreviations

CNN	Convolutional Neural Networks
COPD	Chronic Obstructive Pulmonary Disease
CT	Computer Tomography
ML	Machine Learning

Chapter 1 Introduction

1.1 Background

The field of pneumonia infection diagnostics has witnessed notable achievements, focused on integrating Artificial Intelligence and Machine Learning into traditional diagnostic methodologies. Recent trends emphasize the potential of these technologies in enhancing the accuracy and efficiency of pneumonia detection through the analysis of medical imaging data, particularly chest X-rays. Machine learning algorithms, such as Convolutional Neural Networks (CNNs), have demonstrated significant capabilities in image classification tasks, including the identification of pneumonia-related abnormalities in chest radiographs. These scans provide detailed insights into the condition of the lungs, enabling healthcare professionals to make informed decisions about patient care. However, the manual interpretation of X-Ray images can be prone to human error, highlighting the necessity for technological interventions.

In response to this challenge, our project harnesses the power of Machine Learning and Artificial Intelligence to automate the analysis of lung CT scans. By developing an intelligent system capable of accurately classifying X-Ray images as pneumonia infected, or indicative of healthy lungs, the project enhances the diagnostic process. This advancement holds the potential to revolutionize healthcare practices and contribute to better outcomes in cases of respiratory infections. The project's emphasis on accurate classification of pneumonia infections can lead to earlier intervention and treatment, ultimately alleviating the burden on healthcare systems and contributing to better patient outcomes.

While strides have been made, existing works in pneumonia diagnostics are not without limitations. One significant drawback lies in the interpretability of models.

1.2 Objectives

These objectives collectively aim to develop a state-of-the-art diagnostic tool that combines technological innovation with clinical expertise, ultimately improving diagnostic outcomes in the field of pulmonary health.

1.Automated Diagnosis: Implement a robust system using convolutional neural networks (CNNs) to autonomously analyze chest X-ray images, aiming for precise identification and classification of various lung infections.

2. Accuracy Enhancement:Strive to achieve optimal levels of accuracy in detecting patterns related to different pulmonary conditions, ensuring that the system's diagnoses align closely with expert evaluations.

3.User-Friendly Interface: Design an intuitive user interface, leveraging technologies like ReactJS, to simplify the interpretation of results for healthcare professionals and facilitate seamless integration into existing workflows.

4.Validation and Refinement: Continuously assess and refine the system's performance through rigorous validation processes, incorporating feedback from medical experts and leveraging diverse datasets to enhance its reliability and effectiveness.

1.3 Motivation and Significance

The selection of our project topic was driven by a strong motivation to make a meaningful impact in the field of healthcare. The motivation to develop an advanced diagnostic tool is driven by the understanding that timely interventions are crucial in improving patient outcomes. Thus, the project aims to provide healthcare professionals with a rapid tool for early detection and intervention.

Our project integrates the CNN technology with a user-friendly interface, ensuring ease of use and accessibility. Notably, our system's features include real-time prediction of lung conditions for the detection of either pneumonia infected lungs or healthy lungs.

Additionally, the model's incorporation of confidence levels, while introducing a level of uncertainty, reflects a commitment to transparency and reliability. The awareness of prediction confidence allows for a more informed interpretation of results and supports the need for additional clinical evaluation when necessary.

Chapter 2 Related Works

2.1 Lung disease detection using feature extraction and extreme learning machine

Ramalho, G.L.B. *et al.* (2014) *Lung disease detection using feature extraction and Extreme Learning Machine*, *Zendy*.

This paper by Geraldo Luis Bezerra Ramalho, Pedro Pedrosa Rebouças Filho, Fátima Nelsizeuma Sombra de Medeiros, Paulo César Cortez tackles COPD diagnosis using CT lung images. It employs ACACM for lung structure segmentation and proposes a novel disease detection method via feature extraction from segmented images using co-occurrence stats. The approach utilizes a Spatial Interdependence Matrix (SIM) for classification, distinguishing healthy lungs, COPD, and fibrosis. Achieving 96% accuracy with ELMNN, it holds potential for clinical decision support systems.

2.2 Pneumonia Detection using CNN

arbazkhan971 (2021a) *Pneumonia detection using CNN(96%+ Accuracy)*, *Kaggle*. This project is mainly focused on pneumonia detection using CNNs, achieving 96% accuracy. The project employs the Python programming language and TensorFlow library to preprocess images, construct and train the CNN architecture on medical image data to classify whether a patient's lungs are affected by pneumonia, and evaluate its performance. This high accuracy indicates the effectiveness of the CNN model in distinguishing between normal and pneumonia-infected lungs. Such machine learning projects have significant potential for aiding medical diagnosis and streamlining healthcare processes.

2.3 Lung Cancer Detection using CT Scan Images

Makaju, S. *et al.* (2018) *Lung cancer detection using CT scan images*, *Zendy*.

This research by Suren Makaju, P.W.C. Prasad, Abeer Alsadoon, A. K. Singh, A. Elchouemi from Charles Sturt University, Sydney, Australia, evaluates existing techniques, identifies the best method's limitations, and proposes enhancements in early detection of lung cancer. While CT scan imaging is valuable, interpreting cancerous cells remains challenging. Computer-aided diagnosis using image processing and machine learning offers solutions. Lung cancer detection techniques were analyzed for accuracy, revealing varying levels of effectiveness. The goal is to improve accuracy toward 100% by addressing drawbacks of current methods. The study aims to enhance early detection and treatment, contributing to more effective management of lung cancer.

Chapter 3 Design and Implementation

The development process was divided in to the following parts:

Data Collection: A dataset consisting of lung x-ray images with further information like the location of the infection/anomaly was collected.

Data Pre-processing: The images present were in '.dcm' format. This format contains the image along with the metadata and other information of the patient like age, the area of the x-ray etc. Since the use of these attributes are out of the scope of this project only the image was extracted and saved as a JPEG file. The dataset was further split into three categories: train, validation and test randomly in the ratio of 80:10:10 since the dataset is sample.

Total Train samples: 21348

Total test and validation samples: 2668 each

Data augmentation: Data augmentation is a technique commonly used in machine learning to artificially increase the size of a training dataset by applying various transformations to the existing data. Since the size of data is respectable enough the only augmentation used was flipping the images half of the time.

Model Selection and Training: ResNet Architecture was chosen for the model because of its benefits like ease the training of very deep networks, improved accuracy, effective feature learning etc.

IoU (Intersection over Union) loss function, often referred to as the Intersection over Union Loss or Jaccard Loss was used to evaluate the model's performance in the validation set.

The model was optimized after fine tuning different parameters like batch size, addition/removal of layers, learning rate, image size etc.

Then the model was evaluated on the test samples to check its performance in unseen data. Then after visualization of the results the model was deployed with a user-friendly interface using ReactJs and Django framework.

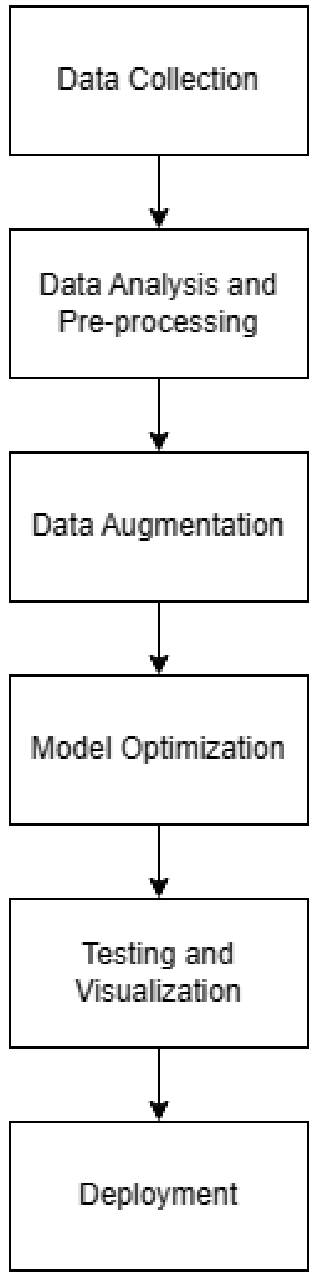


Fig 3.1: Model Training Procedure

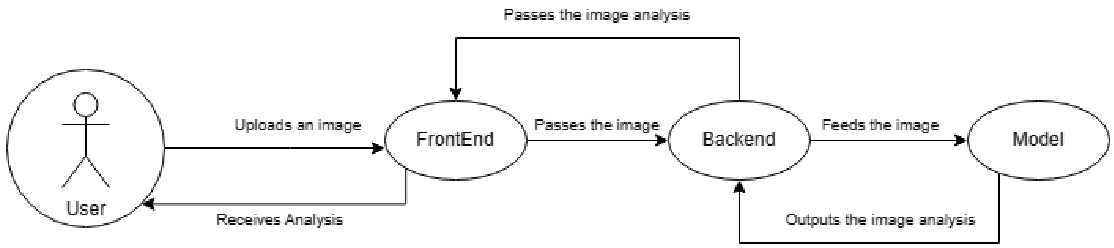


Fig 3.2: Program Flow Diagram

Important Concepts:

Convolution:

Convolution is a mathematical operation for feature detection which involves combining two mathematical functions to produce a third function. In machine learning, it's used to extract meaningful features from input data, especially in image and signal processing. Convolution involves applying a filter (also known as a kernel) to an input, systematically moving the filter across the input data and computing the element-wise multiplication and summation at each position. The result is a new representation of the input, capturing localized patterns or features.

In the context of 2D convolution, a filter (matrix) is systematically applied to an input image by sliding across each position, calculating the element-wise product with the overlapping region, and summing these products to produce a single value in the output. This process, repeated across the entire image, generates a feature map capturing distinctive patterns in the input.

CNN:

A Convolutional Neural Network (CNN) is a specialized deep neural network designed for image and spatial data processing. It comprises convolutional layers that apply filters to capture local patterns, pooling layers for downsampling, and fully connected layers for global feature learning. Utilizing non-linear activation functions like ReLU, CNNs automatically learn hierarchical representations from input data during training. Their architectural flexibility allows adaptation to diverse computer vision tasks, such as image classification, object detection, and segmentation. CNNs have significantly advanced image analysis, enabling applications in fields like medicine, autonomous vehicles, and face recognition, where they learn to discern and interpret complex patterns in visual data.

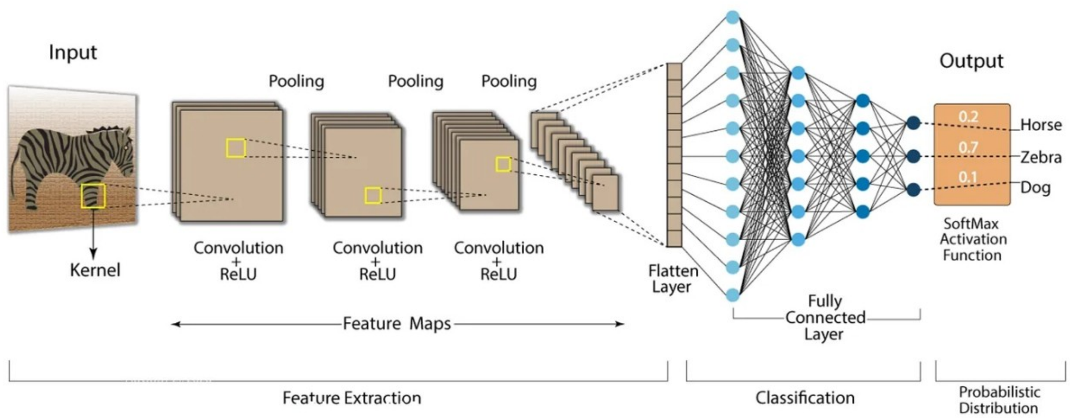


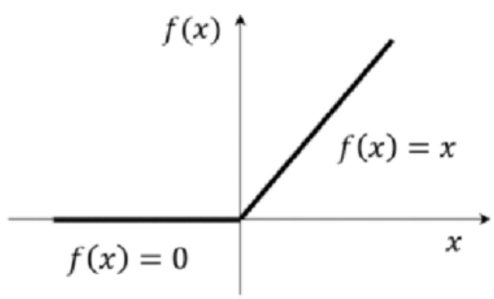
Fig 3.3:CNN Model

Batch Normalization:

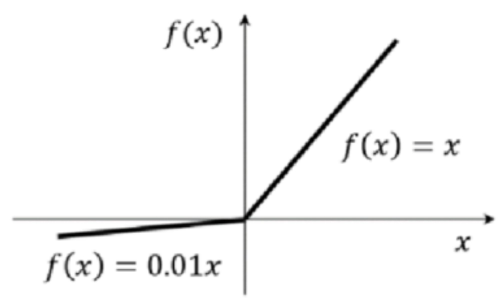
Batch Normalization (BatchNorm) is a technique in deep learning and neural network training that normalizes the input of a layer by adjusting and scaling the activations. It helps stabilize and speed up the training process by addressing issues like internal covariate shift and vanishing/exploding gradients. It addresses issues like internal covariate shift, promoting faster and more stable convergence. It is a standard component in many modern neural network architectures, applied to both fully connected and convolutional layers, contributing to more efficient and effective model training.

Leaky ReLU:

Leaky ReLU is an activation function used in neural networks to overcome the "dying ReLU" problem associated with traditional ReLU. While ReLU sets negative inputs to zero, causing neurons to stop learning, Leaky ReLU introduces a small, non-zero slope to the negative part of the input. This prevents neurons from completely "dying" and helps address the vanishing gradient problem during training. Leaky ReLU retains the computational efficiency and non-linearity of ReLU, making it a popular choice in deep learning architectures.



ReLU activation function



LeakyReLU activation function

Max Pooling:

Max pooling is a down-sampling operation frequently used in Convolutional Neural Networks (CNNs) to reduce the spatial dimensions of input data. It involves selecting the maximum value from a group of neighboring pixels within a window, and this maximum value represents the corresponding region in the downsampled output. Max pooling helps reduce computational complexity, memory requirements, and introduces translation invariance, making CNNs more robust to slight shifts in the input. It preserves essential features by retaining the maximum values and is commonly employed in combination with convolutional layers to extract hierarchical features from input data.

IoU Loss Function:

The Intersection over Union (IoU) loss function, also known as Jaccard loss, measures the dissimilarity between predicted and ground truth binary masks. It is commonly used in image segmentation tasks. The IoU loss is calculated as 1−IoU score, where the IoU score is defined as the intersection of the predicted and ground truth regions divided by their union.

Binary Crossentropy:

Binary Crossentropy, often simply referred to as "BCE" or "log loss," is a loss function used in binary classification tasks. It measures the difference between predicted probabilities and true binary labels. This loss function is commonly employed in neural network models where the output is a probability distribution over two classes (binary classification), and each example in the dataset belongs to one of these classes.

Cosine Annealing:

Cosine Annealing is a learning rate scheduling technique commonly used during the training of neural networks. It dynamically adjusts the learning rate over epochs in a cyclical manner, specifically following a cosine curve. This technique can help improve model convergence and generalization by providing a dynamic and smooth learning rate schedule.

Optimizer:

An optimizer is a crucial component that determines how the model's weights are updated during the training process. An optimizer uses the gradients of the model's parameters (weights and biases) with respect to the loss function to adjust these parameters, aiming to minimize the loss and improve the model's performance.

Different optimizers use various algorithms and strategies for updating weights, and the choice of optimizer can significantly impact the training process.

The optimizer used in the project model, Adam (Adaptive Moment Estimation) is an adaptive learning rate optimization algorithm widely used in training deep neural networks. It combines ideas from RMSprop and Momentum, adapting learning rates individually for each parameter.

3.1 System Requirement Specifications

3.1.1 Software Specifications

To run the project on the client side, the only additional software required is a modern GUI based web browser. For example: MS Edge, Google Chrome etc.

To run the project on the server side following are the software specifications:

- Python 3
- ReactJs
- Following Python libraries and their dependencies:
    - Tensorflow
    - Matplotlib
    - Django

3.1.2 Hardware Specifications

Any additional hardware is not required to run the project on both client side and server side.On the client side almost any modern day computer that can run a browser is able to use the project.

On the server side, an average modern computer system can handle hosting the webpages and using the model to deduce conclusions on the image uploaded without any hiccups. But if the model is to be updated or redefined and recreated locally, a device with high memory and a dedicated GPU is preferred for easier execution.

Chapter 4 Discussion on the achievements

In the course of this project, our team achieved a significant milestone by immersing ourselves in foundational machine learning principles, with an emphasis on Convolutional Neural Networks (CNNs). Additionally, we utilized essential tools such as Jupyter Notebook and Google Colab for code editing and model training. The integration of GitHub for version control aided in efficient code management, collaboration, and project oversight.

The main challenge in the project was the limited processing resources, particularly when training a large amount of data. Initially, the team used Jupyter Notebook for training, but encountered difficulties as the PC's CPU couldn't handle the workload efficiently. The solution was a switch to Google Colab, which provided the necessary computational power to successfully train the model and address the resource limitations. This decision enabled more efficient and effective model training for the project.

4.1 Features:

The Lungs Infection Detector has numerous features.Among them some of them are listed below:

- **Automated Diagnosis**: The project incorporates machine learning techniques to automatically analyze chest X-ray images for the detection of lung infections.
- **User-Friendly Interface**: The diagnostic tool is integrated with a frontend developed using ReactJS, ensuring a seamless experience for users.
- **Area of Interest:** The project utilizes machine learning techniques for accurate diagnosis, highlighting areas likely to be affected by pneumonia in chest X-ray images.

Chapter 5 Conclusion and Recommendation

5.1 Limitations

While our project advances machine learning in pulmonary diagnostics, it's crucial to recognize its limitations that guide future improvements.

- **Binary Classification**: One of the primary limitations is that the current model offers a binary classification, merely determining if an infection is present or absent with a certain confidence level. The system lacks the capability to classify specific types of infections, such as distinguishing between COVID-19, pneumonia, or other pulmonary conditions.
- **Lack of Specificity**: Due to the project's constraints, the diagnostic tool may not provide nuanced insights into the specific nature or type of infection detected. This limitation can restrict clinicians' ability to tailor treatments based on detailed diagnostic information.
- **Confidence Level Dependency**: The diagnostic outcomes are contingent upon confidence levels, implying that there might be instances where the model's predictions fall within a margin of uncertainty, necessitating further clinical evaluation or complementary diagnostic methods.
- **Clinical Applicability**: Given the binary nature of the classification and confidence level dependency, there might be challenges in seamlessly integrating the tool into clinical workflows, potentially limiting its broader applicability and utility for healthcare professionals.

These limitations underscore the need for future enhancements, iterative refinement, and collaborative efforts to develop a more comprehensive, specific, and clinically relevant diagnostic tool in the domain of pulmonary health.

5.2 Future Enhancement

This was the first machine learning project from the team. There are many limitations and problems with the current version of the project so we have a vision to enhance the project in the near future.

- **Introduction of a Classification System**: Develop a system designed to categorize infections into specific diseases, notably distinguishing between COVID-19 infections and typical pneumonia cases, thereby enhancing diagnostic precision.
- **Augmented Training with Comprehensive Datasets**: Expand the model's training regimen by incorporating a broader and more diverse collection of chest X-ray images. This initiative aims to bolster the model's accuracy, reliability, and resilience against imaging variations.
- **Scholarly Contributions**:Pursue an academic trajectory by planning and executing research initiatives. The team aims to publish research articles and papers detailing methodologies, findings, and implications, fostering knowledge dissemination and academic collaboration.
- **Collaborative Endeavors**: Foster collaborations with medical professionals, researchers, and technologists to ensure the project's alignment with clinical needs, technological advancements, and academic rigor.

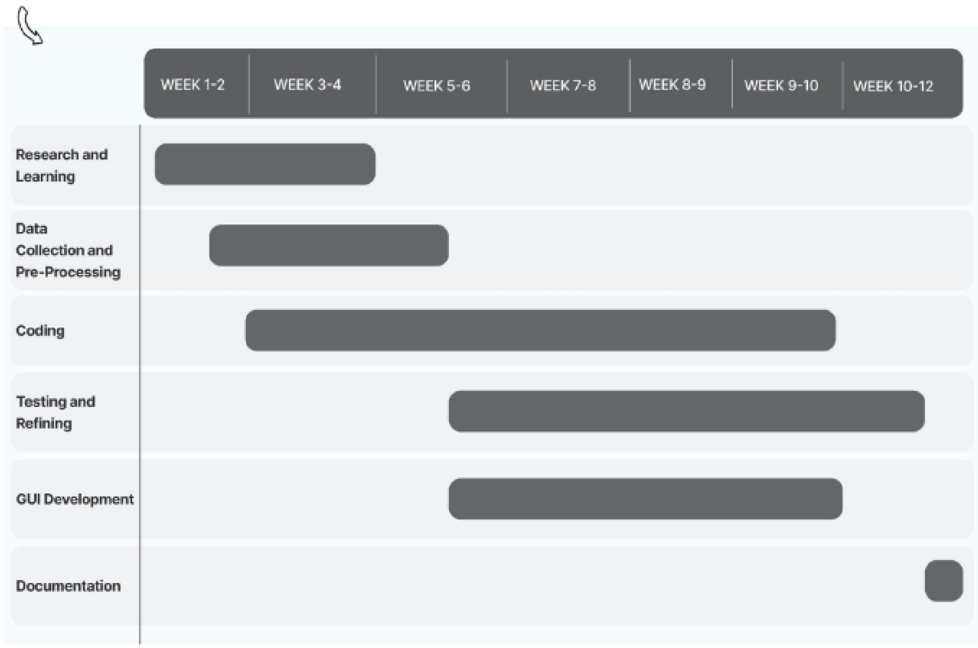
By delineating these future directions, the team underscores its commitment to advancing the project, addressing existing limitations, and contributing meaningfully to the intersection of machine learning and pulmonary health diagnostics.

References

Rahaman, A. (2020, 8 7). *Pneumonia detection with CNN (AUC score 0.94)*. Kaggle. Retrieved December 31, 2023, from https://www.kaggle.com/code/alifrahman/pneumonia-detection-with-cnn-auc-score-0-94/notebook

.

APPENDIX



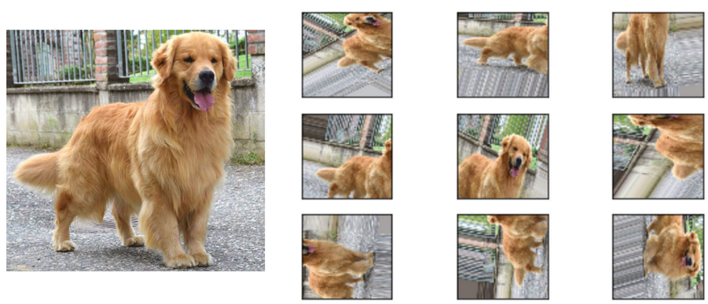
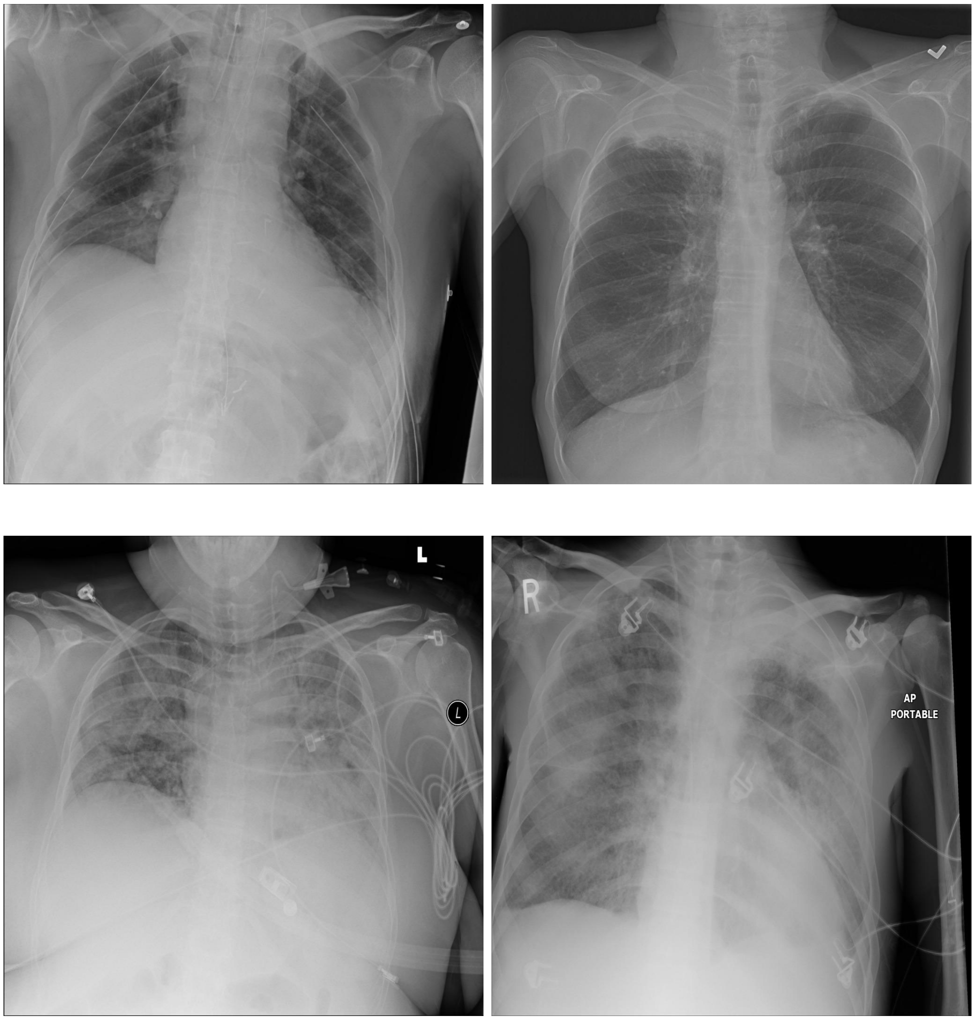


Fig: Example of Data Augmentation

Pneumonia Infected Lungs:



Normal Lungs: