GED-FR-14

Republic of Tunisia
Higher Education and Scientific Research Ministry
University of Carthage
National School of Sciences and Advanced Technologies of Borj Cedria

Synthesis Project I

Title

Pneumonia Detection Using Machine learning and Deep learning Techniques

by :

Hiba Messai and Dhaouadi Jihen(1TA2)

Presented 13/05/ 2025 in front of the jury :

Mr Sami Hafsi	Supervisor - ENSTAB
Mme Ismail Adla	Examinator - ENSTAB

Academic Year : 2024-2025

Ref. :PSI/6

Dedication

I dedicate this project to my beloved family, whose unwavering support and encouragement have been my source of strength throughout this journey.

Special thanks to my mentors and friends who have guided me and believed in me every step of the way.

Thanking

First of all, we would like to express our deepest appreciation to all those who provided us the possibility to complete this report, starting with our supervisor **" Mr. Sami Hafsi"**

We are so grateful for his support and his patience, which have been invaluable to us in carrying out our work successfully.

We would like to thank all **the pedagogical team** of the national school of sciences and advanced technologies and **the jury** for the time they are taking to check our work.

Thank you so much.

Table of contents

List of figures

List of Tables

List of Abbreviations

AI: Artificial intelligence

AUC: Area Under Curve

CT: computed tomography

CXR : chest x-ray

DL : Deep Learning

FN : False Negatives

FP: False Positives

FPR : False Positive Rate

HPC : High-Performing-computing

LASSO : Least Absolute Shrinkage and Selection Operator

ML: Machine Learning

max_depth : (maximum depth of each tree),

min_samples_split : (minimum number of samples required to split an internal node),
min_samples_leaf : (minimum number of samples required to be at a leaf node),
max_features : (the number of feature es to consider when looking for the best split).

n_estimators : (number of trees in the forest)

RF: Random Forest

ROC: Receiver Operating Characteristic

TP: True Positives

TN : True Negatives

XGBoost : Extreme Gradient Boosting

General Introduction

Computer vision is a relevant application field of artificial intelligence (AI), which comes as no surprise, mainly because it offers solutions to a vast amount of problems that we humans face in the 21st century. One of the fields of computer vision that has proved to be fruitful again and again is medical image diagnosis with the aid of AI.

Pneumonia is an acute respiratory infection that has led to significant deaths of people worldwide. This lung disease is more common in people older than 65 and children under five years old. Although the treatment of pneumonia can be challenging, it can be prevented by early diagnosis using Computer-Aided Diagnosis (CAD) systems. Chest X-Rays (CXRs) are currently the primary imaging tool for detection of pneumonia, which are widely used by radiologists. While the standard approach of detecting pneumonia is based on clinicians' decisions, various Deep Learning (DL) methods have been developed for detection of pneumonia considering CAD system. In this regard, a novel hybrid Convolutional Neural Network (CNN) model is proposed using three classification approaches. In the first classification approach, Fully-Connected (FC) layers are utilized for the classification of CXR images. This model is trained for several epochs and the weights that result in the highest classification accuracy are saved. In the second classification approach, the trained optimized weights are utilized to extract the most representative CXR image features and Machine Learning (ML) classifiers are employed to classify the images. In the third classification approach, an ensemble of the proposed classifiers is created to classify CXR images.

Chapter1:Project Context and Motivation

1. Introduction

In an era marked by rapid technological advancements, organizations and institutions are continually seeking innovative solutions to enhance efficiency, productivity, and service quality. This pursuit of innovation often manifests through projects aimed at addressing specific challenges or optimizing existing processes.

The primary objective of this chapter is to present the general context of the project, highlighting the problem it addresses, the goals set, and the methodology adopted for its execution.

2 .Medical Background of Pneumonia

Pneumonia is a respiratory infection caused by bacteria, viruses, and fungi , leading to inflammation of the lungs, with symptoms including fever, muscle aches, coughing and breathing difficulties . If not treated on time, pneumonia can be fatal. Each year, more than four million people die from pneumonia . Antibiotics and antivirals are drugs that can treat pneumonia-caused viruses or bacteria; even so, the ability to diagnose and treat pneumonia early on is necessary for the best possible patient care .

3. Role of Medical Imaging in Diagnosis

Chest X-ray is the frequently used imaging mode for detecting pneumonia, due to its low cost and accessibility . Despite the benefits that X-rays provide, there are only a few expert radiologists whose predictions are required in several areas of the world, according to ref. , and, presently, the use of chest X-rays to detect pneumonia can result in delayed detection and treatment; as such, the use of computer-aided detection (CAD) systems can serve as a solution to these problems .

4. Importance of Early Detection

Early detection of pneumonia plays a vital role in improving patient outcomes and reducing the risk of severe complications or death. Pneumonia can rapidly progress, especially in vulnerable populations such as infants, the elderly, and individuals with weakened immune systems.

Identifying the infection at an early stage allows for timely medical intervention, including antibiotic therapy, oxygen support, and in some cases, hospitalization.

From a clinical perspective, early detection helps prevent the spread of infection, reduces the duration of illness, and significantly lowers healthcare costs. It can also reduce the need for intensive care and lower the chances of long-term lung damage or secondary infections.

In summary, early detection is not only essential for individual patient care but also crucial in improving public health outcomes and reducing the overall burden on healthcare systems.

5. Limitations of Traditional Methods

Traditional methods of pneumonia diagnosis typically involve clinical examination, patient history, and radiological assessment, primarily through chest X-ray interpretation by radiologists. While these approaches are effective in many settings, they present several limitations:

Subjectivity in Interpretation: The accuracy of chest X-ray interpretation largely depends on the expertise and experience of the radiologist. This introduces variability and can lead to misdiagnosis, especially in subtle or early-stage cases.

1. **Delayed Diagnosis**: In busy healthcare environments or rural areas with limited access to imaging experts, the time from examination to diagnosis may be prolonged, delaying critical treatment.

2. **Resource Dependence**: Access to radiologists, imaging equipment, and laboratory support is not always guaranteed, particularly in low-resource settings. This limits the scalability of traditional methods for mass screening or rapid diagnostics.

3. **Limited Sensitivity**: Clinical signs and symptoms of pneumonia (e.g., fever, cough, difficulty breathing) often overlap with other respiratory illnesses, making clinical diagnosis alone unreliable. X-rays may not always show clear signs of infection, especially in the early stages or in patients with pre-existing lung conditions.

4. **High Workload for Radiologists**: With increasing volumes of imaging data, radiologists face a growing workload. This can lead to fatigue, increased error rates, and longer turnaround times.

6. Role of AI in Medical Diagnosis

The computer can help the human expert in establishing the diagnosis of pneumonia due to its high speed and objective reproducible judgment, as the signs of pneumonia in X-ray images are not always visible or legible for the human eye.

Machine learning is being utilized in medical imaging tasks, mapping, and diagnosing images . Machine learning can solve complex computer vision problems in medical imaging . However, deep learning has taken giant steps in these models because of its ability to learn hierarchical features from raw image data.

Deep learning techniques achieve classification such that the key features are spontaneously extracted and classified . Compared to other deep learning techniques, CNN is one of the effective approaches for pattern recognition because of its layering topography .

7. Problem Statement

Given the growing volume of medical imaging data and the increasing pressure on healthcare systems, there is a critical need for automated diagnostic tools capable of assisting or augmenting human expertise.

This project aims to address the following challenge: How can we design a reliable AI-based system to automatically detect pneumonia from chest X-ray images, while achieving accuracy comparable to that of medical experts?

8. Objectives of the Project

The specific objectives are as follows:

1. **Develop an Automated Pneumonia Detection System**
   Create an AI-based system capable of detecting pneumonia from chest X-ray images to assist healthcare professionals in diagnosis.
2. **Compare Different Machine Learning Models**
   Implement and compare three models :
   A **Convolutional Neural Network (CNN)** for deep learning-based image classification.

A **Random Forest (RF)** classifier using manually extracted features.
An **XGBoost** model, leveraging gradient boosting for enhanced prediction accuracy.

3. **Assess Model Performance**
 Evaluate and compare the effectiveness of each model using key performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.
4. **Enhance Early Diagnosis**
 Contribute to the early detection of pneumonia by providing a reliable tool that supports medical professionals in making faster, more accurate decisions.
5. **Identify Strengths and Weaknesses**
 Analyze the strengths and weaknesses of each model to identify which approach offers the most reliable and efficient solution for pneumonia detection.

9.Conclusion

In this chapter, we established the foundation and motivation for developing an AI-based system for pneumonia detection. We began by understanding the medical significance of pneumonia, a condition that continues to pose a serious global health threat, especially among vulnerable populations. Traditional diagnostic methods, though widely used, often suffer from limitations such as subjectivity, time consumption, and dependency on clinical expertise.

We highlighted the vital role of medical imaging in diagnosis and the increasing relevance of early detection in improving patient outcomes. Against this backdrop, artificial intelligence—particularly deep learning—emerges as a promising solution that can enhance diagnostic accuracy, efficiency, and accessibility.

By defining the problem clearly and outlining the key objectives of this project, we set the stage for exploring the design, development, and evaluation of an AI-based tool to assist in pneumonia detection from chest X-ray images. The following chapters will delve into the technical details, implementation, and validation of this system.

Chapter 2: Theoretical and Technical Background

1. Introduction

To develop an effective AI-based system for pneumonia detection, it is essential to understand the theoretical and technical foundations that underpin such technologies. This chapter provides a comprehensive overview of the key machine learning and deep learning concepts that form the basis of the proposed system.

We begin by introducing fundamental principles of machine learning, followed by an exploration of deep learning, a subset of machine learning that excels in tasks involving complex data such as medical images. Classical algorithms like Decision Trees, Random Forests, Gradient Boosting, and XGBoost are discussed to provide context on alternative modeling approaches and their relevance.

Special focus is given to Convolutional Neural Networks (CNNs), the architecture most commonly used for image classification tasks, including medical imaging. Understanding the structure and operation of CNNs is critical to appreciating how such models detect pneumonia from chest X-rays.

Finally, we introduce evaluation metrics used to measure model performance in the medical domain. These metrics help ensure that the AI system not only performs well statistically but also aligns with clinical priorities such as sensitivity and specificity.

2. Machine Learning overview

Machine learning [1] is a branch of artificial intelligence that enables algorithms to uncover hidden patterns within datasets. It allows them to predict new, similar data without explicit programming for each task. Machine learning finds applications in diverse fields such as image and speech recognition, natural language processing, recommendation systems, fraud detection, portfolio optimization, and automating tasks.Machine learning's impact extends to autonomous vehicles, drones, and robots, enhancing their adaptability in dynamic environments. This approach marks a breakthrough where machines learn from data examples to generate accurate outcomes, closely intertwined with data mining and data science.

Today, machine learning is the technology behind ChatGPT, self-driving cars, and all so-called “intelligent” systems developed over the past thirty years. It can be said that machine learning is truly the core discipline of artificial intelligence.

2.1 Decision Trees and Random Forests

2.1.1 Decisions Trees

Decision tree [2] is a simple diagram that shows different choices and their possible results helping you make decisions easily.

A decision tree is a graphical representation of different options for solving a problem and show how different factors are related. It has a hierarchical tree structure starts with one main question at the top called a node which further branches out into different possible outcomes where :

- **Root Node** is the starting point that represents the entire dataset.
- **Branches**: These are the lines that connect nodes. It shows the flow from one decision to another.
- **Internal Nodes** are Points where decisions are made based on the input features.
- **Leaf Nodes**: These are the terminal nodes at the end of branches that represent final outcomes or predictions.

2.2.2 Random Forest

Random forest [3] is a commonly-used machine learning algorithm, combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.Also, Random Forest provides insights into feature importance, aiding in understanding which variables most influence the model's predictions. Its ensemble approach ensures resilience to noise and outliers, contributing to more reliable and generalized outcomes.While Random Forest offers numerous benefits, there are also certain challenges to be aware of such as Complexity **;** The ensemble of numerous trees in a Random Forest can make the model less interpretable compared to a single decision tree .Also, Computational Cost ; Training multiple trees can be resource-intensive, especially with large datasets.

2.2 Gradient Boosting and XGBoost

2.2.1 Gradient boosting

Gradient boosting[4] is a machine learning technique based on boosting in a functional space, where the target is pseudo-residuals instead of residuals as in traditional boosting. It gives a prediction model in the form of an ensemble of weak prediction models, i.e., models that make very few assumptions about the data, which are typically simple decision trees When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. As with other boosting methods, a gradient-boosted trees model is built in stages, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

2.2.2 XGBoost

XGBoost (which stands for Extreme Gradient Boosting) is an advanced machine learning algorithm that belongs to the family of ensemble methods, specifically boosting techniques. Boosting is a method that builds a strong model by combining multiple weak models (typically decision trees). In XGBoost :

- Each new tree tries to correct the mistakes made by the previous trees.
- It uses a mathematical technique called gradient descent to minimize errors during training.
- It includes several optimizations like regularization (to prevent overfitting), parallel processing, and handling missing data efficiently.
- XGBoost is known for being fast, scalable, and highly accurate, making it a favorite in data science competitions like those on Kaggle.

3. Deep learning

Deep Learning is transforming the way machines understand, learn, and interact with complex data. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data.

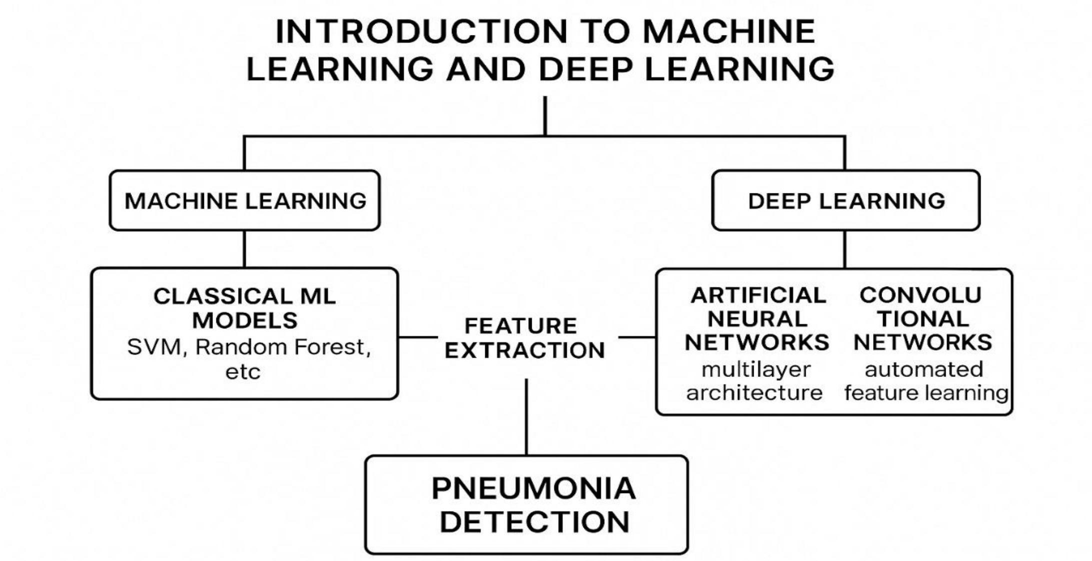


Fig_ 2.1**:** Overview of Machine Learning and Deep Learning Approaches for Pneumonia Detection

3.1 CNNs(Convolutional Neural Network)

3.1.1 CNN Architecture for Image Classification

A Convolutional Neural Network (CNN) [5] , also known as ConvNet, is a specialized type of deep learning algorithm mainly designed for tasks that necessitate object recognition, including image classification, detection, and segmentation. CNNs are employed in a variety of practical scenarios, such as autonomous vehicles, security camera systems, and others.

The importance of CNNs

There are several reasons why CNNs are important in the modern world, as highlighted below:

- CNNs are distinguished from classic machine learning algorithms such as SVMs and decision trees by their ability to autonomously extract features at a large scale, bypassing the need for manual feature engineering and thereby enhancing efficiency.
- The convolutional layers grant CNNs their translation-invariant characteristics, empowering them to identify and extract patterns and features from data irrespective of variations in position, orientation, scale, or translation.
- A variety of pre-trained CNN architectures, including VGG-16, ResNet50, Inceptionv3, and EfficientNet, have demonstrated top-tier performance. These models can be adapted to new tasks with relatively little data through a process known as fine-tuning.

- Beyond image classification tasks, CNNs are versatile and can be applied to a range of other domains, such as natural language processing, time series analysis, and speech

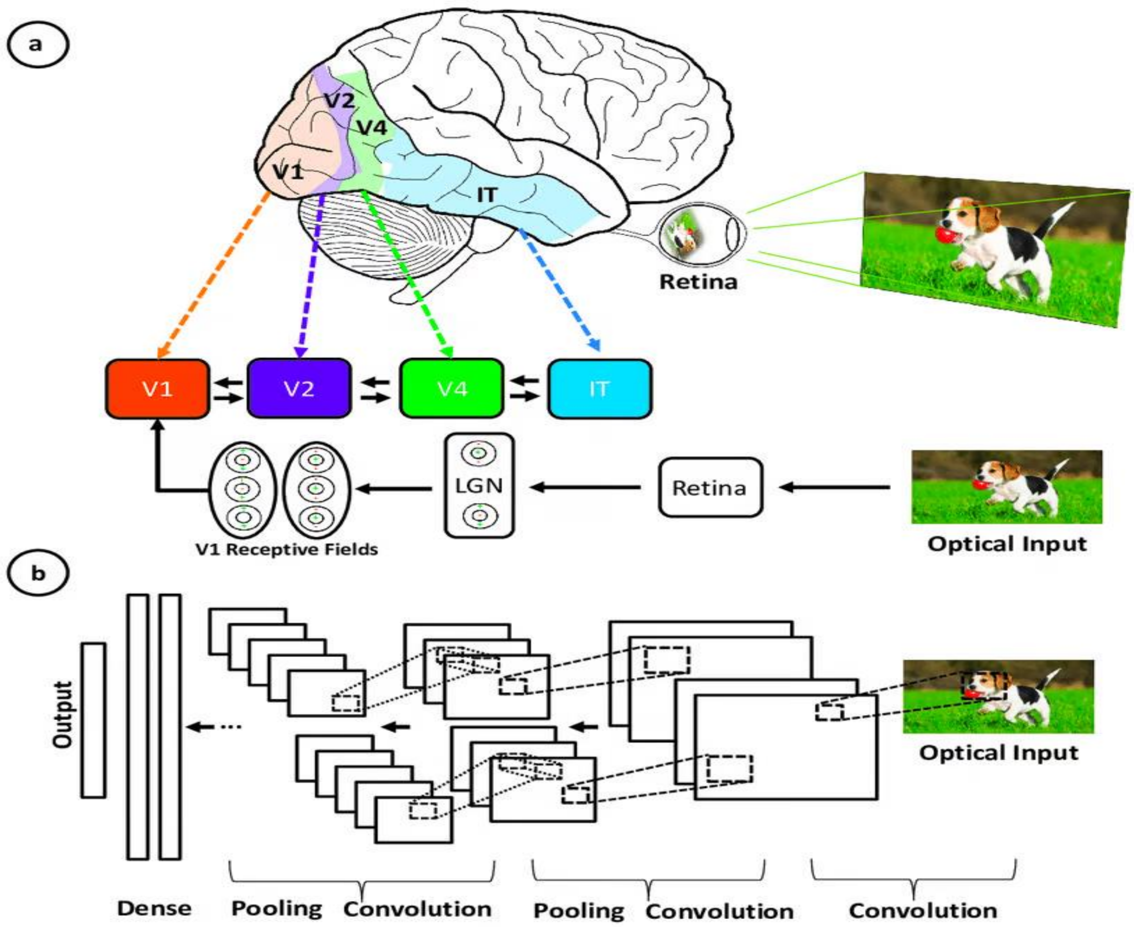


Fig _2.2:Inspiration Behind CNN and Parallels With The Human Visual System [6]

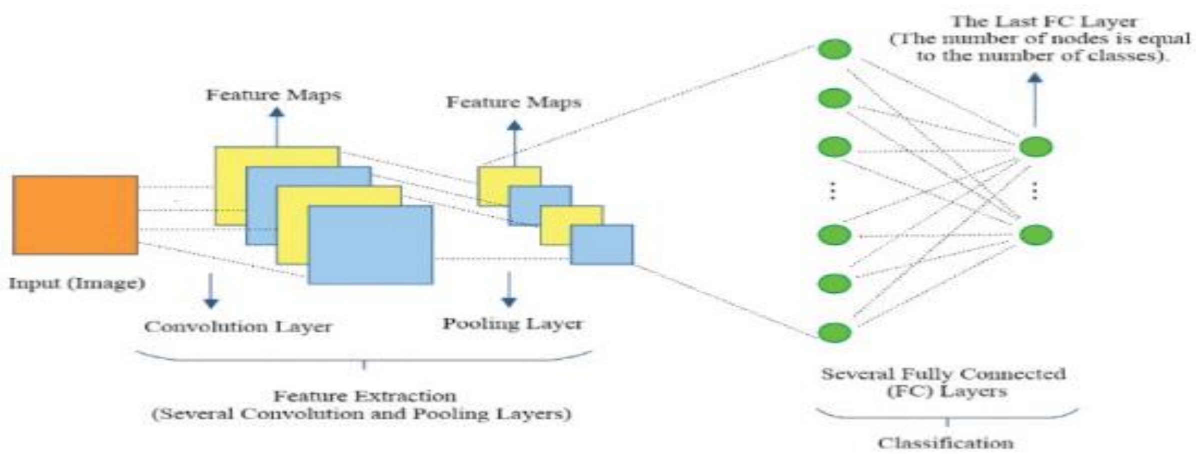


Fig 2.3: Architecture of a Convolutional Neural Network (CNN) for Image Classification [7]

Fig 2.4:The architecture of VGG16 network. [8]

Fig 2.5:The architecture of VGG19 network [9]

4. Evaluation Metrics in Medical AI

In medical AI, especially in tasks such as disease detection from medical images, evaluating the performance of models using suitable metrics is crucial. Traditional metrics like accuracy may not be sufficient, particularly in imbalanced datasets where one class (e.g., healthy patients) significantly outnumbers the other (e.g., pneumonia cases). Therefore, a combination of metrics is typically used to assess the model's effectiveness from various angles.

A confusion matrix is a table used to evaluate the performance of a classification model. It compares the predicted labels from the model to the true labels from the data .For a binary classification problem, the confusion matrix looks like this :

Table 2.1: confusion matrix table

	Predicted Positive	Predicted Négatif
Actual Positive	True Positive (TP)	False Negative(FN)
Actual Negative	False Positive(FP)	True Negative(TN)

4.1 Accuracy ,Precision ,Recall and F1 Score

4.1.1 Accuracy

Accuracy is the most common metric to be used in everyday talk. Accuracy answers the question “Out of all the predictions we made, how many were true?”

$$\text{Accuracy} = \frac{\text{True positives+True Negatives}}{\text{True Positives+True Negatives+False Negatives+False Positives}} \qquad (2.1)$$

As we will see later, accuracy is a blunt measure and can sometimes be misleading.

4.1.2 Precision

Precision is a metric that gives you the proportion of true positives to the amount of total positives that the model predicts. It answers the question **“**Out of all the positive predictions we made, how many were true ? “

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives +False Positives}} \qquad (2.2)$$

4.1.3 Recall

Recall focuses on how good the model is at finding all the positives. Recall is also called true positive rate and answers the question “Out of all the data points that should be predicted as true, how many did we correctly predict as true?”

$$\text{Recall} = \frac{\text{True positives}}{\text{True Positives+False Negatives}} \qquad (2.3)$$

4.1.4 F1 Score

F1 Score is a measure that combines recall and precision. As we have seen there is a trade-off between precision and recall, F1 can therefore be used to measure how effectively our models make that trade-off.

$$\text{F1}=2.\frac{\text{Precision.Recall}}{\text{Precision+Recall}} \qquad (2.4)$$

One important feature of the F1 score is that the result is zero if any of the components (precision or recall) fall to zero. Thereby it penalizes extreme negative values of either component.

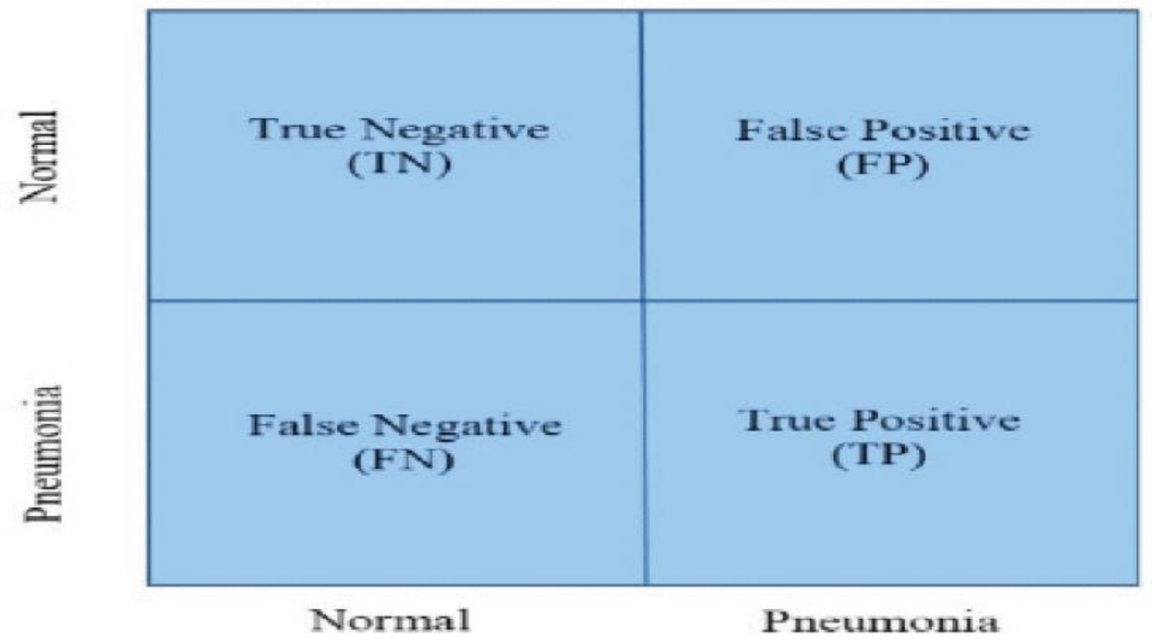


Figure 2.6 : A general confusion matrix for evaluating the performance of the models

4.2 ROC Curve and AUC

4.2.1 ROC curve

The ROC curve [10] is a visual representation of model performance across all thresholds. The long version of the name, receiver operating characteristic, is a holdover from WWII radar detection.

The ROC curve is drawn by calculating the true positive rate (TPR) and false positive rate (FPR) at every possible threshold (in practice, at selected intervals), then graphing TPR over FPR. A perfect model, which at some threshold has a TPR of 1.0 and a FPR of 0.0, can be represented by either a point at (0, 1) if all other thresholds are ignored.

4.2.2 Area under the curve (AUC)

The area under the ROC curve (AUC) [11] represents the probability that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative.

The perfect model above, containing a square with sides of length 1, has an area under the curve (AUC) of 1.0. This means there is a 100% probability that the model will correctly rank a randomly chosen positive example higher than a randomly chosen negative example. In other words, looking at the spread of data points below, AUC gives the probability that the model will place a randomly chosen square to the right of a randomly chosen circle, independent of where the threshold is set .

4.2.3 Oversampling Technique

In machine learning, an imbalanced dataset occurs when certain classes ("normal" images) have significantly more or fewer examples than others (like "pneumonia" images). This imbalance can lead to biased model training, where the model may become biased towards the majority class and underperform on the minority class. To prevent this, we use techniques like oversampling.

4.2.4 Binary Cross Entropy

Binary Cross-Entropy [12] is a loss function used in binary classification problems. It measures the difference between the predicted probability and the actual class label. It is defined as :

$$\text{BCE} = \frac{-1}{N}\sum_{I=1}^{N}[yi.\log(\widehat{yi}) + (1 - yi)\log(1 - \widehat{yi})] \qquad (2.5)$$

Where :

N is the Numbers of samples .

yi is the actual label (0 or 1).

$\widehat{yi}$ is the predicted probability of class 1(output of a sigmoid fuction).

Utility in Practice :

- **Interpretable**: BCE directly penalizes incorrect predictions based on how confident the model is. A confident wrong prediction (e.g., predicting 0.99 for class 0) is penalized more than an uncertain one.
- **Probabilistic Output**: It works with probabilistic outputs, making it suitable for models using a sigmoid activation in the final layer.

- **Standard for Binary Classification**: Widely used in medical imaging tasks like pneumonia detection, where the model decides whether a condition is present (1) or not (0).

Why It's Important :

- Ensures the model focuses on minimizing the probability error, not just the hard classification result.
- Especially useful when dealing with imbalanced datasets, as it takes the predicted probabilities into account instead of only the predicted class.

5.Conclusion

This chapter presented the theoretical and technical foundations essential for understanding the development of an AI-based pneumonia detection system. We began with an overview of machine learning and deep learning, highlighting their significance in modern data-driven healthcare applications. Specific algorithms such as Decision Trees, Random Forests, Gradient Boosting, and XGBoost were introduced to illustrate classical machine learning approaches.

We then transitioned into deep learning, focusing on Convolutional Neural Networks (CNNs), which have proven particularly effective in image classification tasks such as analyzing chest X-rays. A detailed explanation of CNN architecture provided insight into how these models process visual information to detect patterns indicative of pneumonia.

Finally, we explored evaluation metrics commonly used in medical AI, including accuracy, precision, recall, F1 score, and ROC-AUC. Understanding these metrics is crucial for assessing the performance and reliability of predictive models in a clinical setting.

This foundational knowledge sets the stage for the next chapter, where we will discuss the methodology used to build and train our pneumonia detection model.

Chapter 3 : Implementation of an Intelligent Model in an Interface

1.Introduction

This chapter presents the dataset used in this project along with the complete implementation pipeline for pneumonia detection. We begin by introducing the dataset, describing its structure, labeling, and preprocessing steps necessary to ensure reliable model training and evaluation. Following that, we outline the implementation of several machine learning and deep learning techniques employed during experimentation.

We first explore traditional machine learning classifiers—Random Forest (RF) and XGBoost—highlighting how they were trained on extracted features and optimized for performance. We then transition to a deep learning approach, implementing a Convolutional Neural Network (CNN) designed from scratch to learn spatial features directly from chest X-ray images. Finally, we leverage the power of transfer learning by fine-tuning a pretrained CNN model, which enables improved accuracy and faster convergence thanks to prior knowledge learned from large-scale image datasets.

his structured approach allows us to compare the effectiveness of conventional algorithms with deep learning solutions in detecting pneumonia from medical images.

2 Dataset Description

2.1 Chest-X-ray Dataset (source ,size ,classes)

The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

Normal Samples

Pneumonia Samples

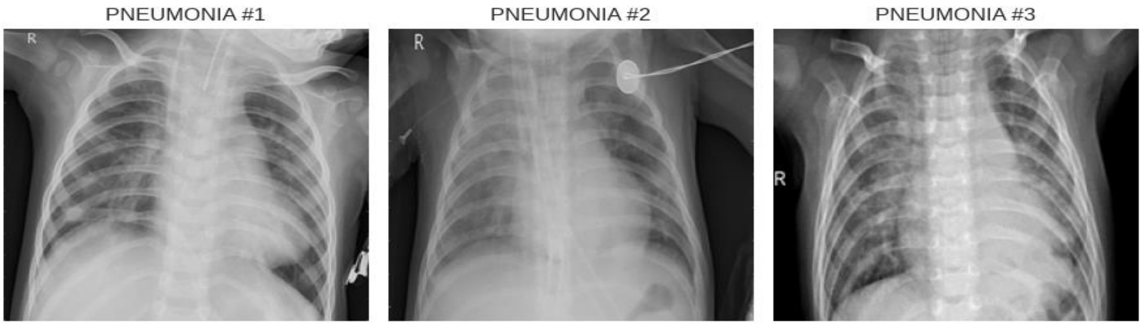


Fig. 3.1 Some examples of input images with their ground truth [13]

2.2 Data Preprocessing and Data Generator

2.2.1 Image Normalization and Resining

Although standard RGB images contain three color channels (Red, Green, and Blue), the chest X-ray images used in this project are grayscale, meaning they contain only a single intensity channel. These images were originally in grayscale format and were resized to 224×224 pixels to standardize input dimensions for the model.

To ensure compatibility with certain deep learning models that expect 3-channel inputs (e.g., pretrained convolutional neural networks such as ResNet or VGG), the grayscale images were converted to 3-channel format by duplicating the single grayscale channel across all three

RGB channels. This conversion preserves the visual information while meeting the model's input requirements.

All pixel values were then normalized by dividing by 255, scaling them from the range [0, 255] to [0, 1]. This normalization step helps ensure a consistent data distribution across images, which promotes faster and more stable convergence during training.

2.2.2 Data Augmentation Techniques

Data Augmentation

Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. It includes making minor changes to the dataset or using deep learning to generate new data points.

Data Augmentation Techniques

To enhance the robustness and generalization of the model, data augmentation was applied using Keras' ImageDataGenerator. This technique artificially increases the diversity of the training dataset by applying random transformations to the input images during training. This helps prevent overfitting and improves the model's ability to generalize to new, unseen data.

The following augmentations were used:

- rotation_range = 30 : Randomly rotates images within a range of ±30 degrees, allowing the model to become invariant to slight orientation changes.
- width_shift_range = 0.2 & height_shift_range=0.2: Shifts the image horizontally or vertically by up to 20% of its width or height, helping the model learn spatial invariance.
- shear_range = 0.2 : Applies random shearing transformations (like slanting the image), which helps simulate different viewing angles.
- zoom_range = 0.2 : Randomly zooms in or out by up to 20%, encouraging the model to recognize patterns at varying scales.
- horizontal_flip = True : Randomly flips images horizontally, which is useful for medical images like chest X-rays where left-right symmetry can be present.
- fill_mode = 'nearest' : Determines how newly created pixels are filled after transformations. 'nearest' fills empty pixels with the value of the nearest pixel.

- These augmentations are applied on the fly during training, meaning the original dataset is not permanently changed, but each batch of training data can contain different variations of the same images.

Data Generator

A data generator is a tool or function in machine learning that efficiently loads, preprocesses, and feeds data (like images) to a model in small batches during training. Instead of loading an entire dataset into memory at once, it processes data on-the-fly, often applying transformations (e.g., resizing, normalization, or augmentation) to improve model performance. In your case, it would take image file paths and labels from a DataFrame (like train_df), prepare them (e.g., resize to 224x224 and scale pixel values), and deliver them in batches for training a model to classify "NORMAL" vs. "PNEUMONIA".

2.3 Train-Test Split Strategy

Normalized images are then split into :

Train set: 3846 samples (65.0%)

Validation set: 1243 samples (21.0%)

Test set: 829 samples (14.0%)

Total: 5918 samples

The images were split in a way that retains the same proportion of classes in the train and test sets that are found in the entire original dataset, meaning that the proportion of healthy and unhealthy lungs in the training and testing dataset are identical.

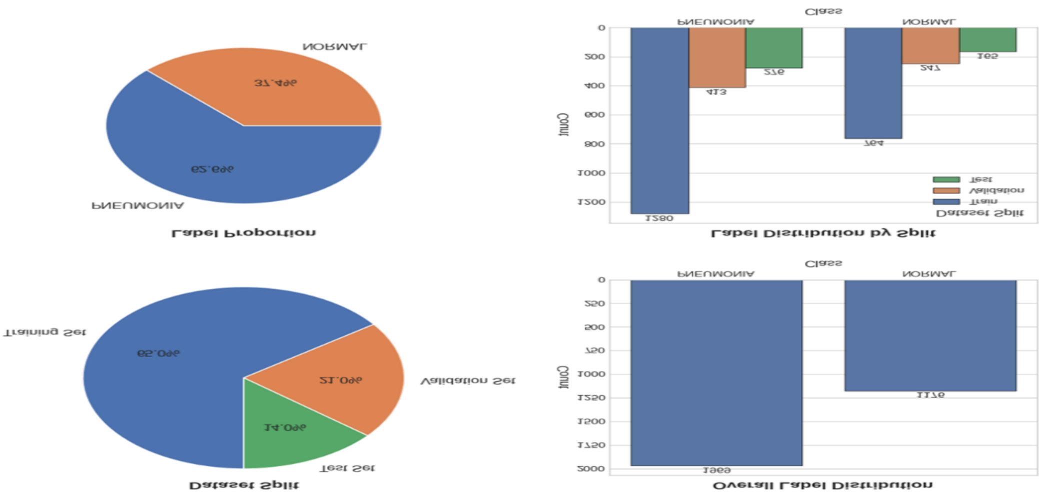


Figure 3.2:Dataset and Label Distribution Analysis [5]

2.4 Handling Data Imbalance with Oversampling

For Chest-X Ray dataset, we have chosen to make the number of "normal" images equal to **2777**. This is done by creating more copies of the "normal" class images until their count matches the specified number. Oversampling will ensure that your model is trained with a balanced distribution, reducing the bias toward the majority class and helping the model learn more effectively to distinguish between the two classes.

By balancing the dataset, the model can now have a fair opportunity to learn from both classes equally. This will likely improve the model's generalization ability and reduce the risk of overfitting, which is a common problem when dealing with imbalanced datasets.

3. Tools and Libraries Used

To carry out this project efficiently, a comprehensive set of libraries and tools was utilized across various stages of development

Data Manipulation and Numerical Operations

- Pandas: For loading, cleaning, and analyzing tabular data.
- NumPy: For numerical computations and array-based operations.

Data Visualization

- Matplotlib: For creating basic plots and visualizations (e.g., accuracy/loss curves).
- Seaborn: For advanced statistical visualizations (e.g., heatmaps, class distributions).

Image Processing

- OpenCV: For resizing, converting, and preprocessing medical images.
- PIL (Python Imaging Library): For basic image manipulation and format conversion.

Machine Learning and Evaluation

- Scikit-learn:
  - For train-test splitting and model evaluation.
  - Provided metrics such as accuracy, precision, recall, and F1-score.

Deep Learning

- Tenserflow and keras

  - For building and training convolutional neural networks (CNNs).
  - Provided pretrained models (e.g., VGG19, EfficientNet) for transfer learning.
  - Enabled callbacks such as EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint.
- ImageDataGenerator: For real-time image augmentation and data normalization.

- image_dataset_from_directory: For efficiently loading image datasets as tf.data objects.

  tensorflow: The core deep learning framework used to build and train neural networks.
- tensorflow.keras.layers: Provides various building blocks (e.g., Dense, Conv2D, Dropout) for constructing custom neural network architectures.
- tensorflow.keras.regularizers: Used to apply regularization techniques like L2 to prevent overfitting.
- tensorflow.keras.Model: Enables the creation of flexible, custom model architectures using the functional API.

  - tensorflow.keras.utils.plot_model: Allows for visualization of model architecture, helpful for documentation and understanding layer connectivity.

Utilities :

- TQDM: For displaying progress bars during lengthy loops (e.g., image loading or feature extraction).
- Warnings module: To suppress unnecessary warning messages and keep the output clean

4.Pneumonia Detection Using Machine Learning

Table 3.1: Pneumonia Detection –Model Summary

Aspect	**Random Forest**	**XGBoost**
Feature Selection	LASSO (removes irrelevant features by shrinking coefficients to zero)	LASSO (same method for feature reduction and overfitting control)

Aspect	Random Forest	XGBoost
Model Type	Ensemble of decision trees (Bagging)	Gradient boosting of decision trees (Additive learning)
Hyperparameter Tuning	Grid Search with Cross-Validation	Grid Search with Cross-Validation
Key Hyperparameters	- n_estimators- max_depth- min_samples_split- min_samples_leaf- max_features	- learning_rate- n_estimators- max_depth- subsample- colsample_bytree
Evaluation Metric	Binary Cross-Entropy Loss (Log Loss)	Log Loss (eval_metric = 'logloss')
Goal	Improve accuracy and reduce overfitting by selecting optimal tree structures	Enhance performance through fine-grained boosting and regularization

4.1 Feature Selection Methods

Feature selection is an essential preprocessing step in machine learning, especially when working with high-dimensional datasets such as those used for pneumonia detection. In this study, we initially applied LASSO (Least Absolute Shrinkage and Selection Operator) to reduce feature dimensionality and prevent overfitting. However, we observed that LASSO did not lead to any significant improvement in model accuracy, indicating that it may have removed features that still held predictive value.

As a result, we adopted the K-Best feature selection method, which selects the top k features based on univariate statistical tests. This approach allowed for greater control over the number of features retained and led to better performance in our machine learning models. The table below summarizes the differences between LASSO and K-Best in terms of methodology and suitability.

Table 3.2: Feature Selection Methods – LASSO vs K-Best

Aspect	LASSO	K-Best (SelectKBest)
Type	Regularization-based feature selection	Statistical test-based feature selection
Technique	Shrinks coefficients of less important features to zero during model fitting	Selects top K features based on univariate statistical tests
Model Dependency	Model-based (Lasso regression)	Model-agnostic (depends only on feature-target relationship)
Handles Multicollinearity	Yes – by penalizing coefficients, it can reduce multicollinearity	No – it evaluates each feature independently
Tuning Parameter	Regularization strength (alpha)	Number of top features to select (k)
Computational Cost	Higher (involves model fitting and optimization)	Lower (only requires computation of test statistics)
Output	Selected features with non-zero coefficients	Top K ranked features based on scores
Use Case Suitability	Best when integrated into model pipelines with high-dimensional data	Best for quick filtering or when domain knowledge informs feature count (k)

4.2. Random Forest Model Evaluation

The Random Forest classifier was trained and evaluated to detect pneumonia from chest X-ray data. The model demonstrated strong performance across all key classification metrics. It achieved an overall accuracy of 92%, indicating that 92% of the test samples were correctly classified.

For the pneumonia class (label 1), the model reached a precision of 94%, a recall of 93%, and an F1-score of 94%. These metrics reflect the model's ability to accurately identify pneumonia cases while minimizing false positives and false negatives. Similarly, for the normal class (label 0), the model achieved a precision of 91%, recall of 92%, and F1-score of 91%.

The confusion matrix provides further insight into the classifier’s performance, with 542 true positives and 382 true negatives, compared to 40 false negatives and 35 false positives. These low misclassification rates highlight the model's robustness.

Additionally, the ROC AUC score of 0.9752 indicates excellent discriminatory power, confirming that the model can effectively distinguish between pneumonia and non-pneumonia cases.

Overall, the Random Forest classifier demonstrates high accuracy, balanced precision and recall, and excellent class separability, making it a reliable choice for pneumonia detection in this context.

```
Random Forest:
              precision    recall  f1-score   support

           0       0.91      0.87      0.89       417
           1       0.91      0.94      0.92       591

    accuracy                           0.91      1008
   macro avg       0.91      0.90      0.90      1008
weighted avg       0.91      0.91      0.91      1008

Confusion Matrix:
 [[362  55]
 [ 38 553]]
ROC AUC: 0.9645441007600011
```

Fig 3.3: Model Performance with Random Forest

4.3 XGBoost Model Evaluation

The XGBoost classifier was applied to the pneumonia detection task and demonstrated strong and reliable performance. It achieved an overall accuracy of 92%, indicating that the model correctly classified 92% of the test samples.

For the pneumonia class (label 1), XGBoost reached a precision of 94%, a recall of 93%, and an F1-score of 93%. These results reflect the model's effectiveness in correctly identifying pneumonia cases with minimal false positives and false negatives. For the normal class (label 0), the precision was 90%, recall was 92%, and the F1-score was 91%.

The confusion matrix shows 539 true positives and 382 true negatives, with only 43 false negatives and 35 false positives, indicating strong sensitivity and specificity.

Importantly, the model achieved a ROC AUC score of 0.9780, the highest among the tested models, indicating excellent ability to distinguish between pneumonia and normal cases. This high AUC confirms that the XGBoost classifier has superior discriminative power.

In summary, XGBoost delivers high accuracy, balanced precision and recall, and exceptional class separation, making it a highly suitable model for reliable pneumonia detection.

```
XGBoost:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88       417
           1       0.91      0.93      0.92       591

    accuracy                           0.91      1008
   macro avg       0.90      0.90      0.90      1008
weighted avg       0.91      0.91      0.91      1008

Confusion Matrix:
 [[364  53]
 [ 42 549]]
ROC AUC: 0.9627627846961011
```

Fig 3.4 : Model Performance with XGBoost

4.4 .CNN Model Evaluation

The Convolutional Neural Network (CNN) was developed and trained for pneumonia detection using chest X-ray images. The model achieved a strong overall accuracy of 91%, correctly classifying the majority of the samples in the test set.

For the pneumonia class (label 1), the CNN achieved a precision of 93%, recall of 92%, and an F1-score of 92%, indicating a high ability to detect pneumonia cases while keeping false positives and false negatives relatively low. For the normal class (label 0), the model reached a precision of 89%, recall of 90%, and F1-score of 89%.

The confusion matrix shows 536 true positives and 375 true negatives, with 46 false negatives and 42 false positives, reflecting the model's slightly higher misclassification rate compared to the other classifiers tested.

Despite this, the model demonstrated excellent overall discrimination capability, as indicated by a ROC AUC score of 0.9605, confirming its effectiveness in distinguishing between pneumonia and non-pneumonia cases.

In conclusion, the CNN model performed well, especially in terms of detecting pneumonia cases with high precision and recall. While slightly behind the XGBoost and Random Forest classifiers in some metrics, it remains a competitive deep learning approach for medical image classification.

```
CNN Evaluation
[[375  42]
 [ 46 536]]
              precision    recall  f1-score   support

           0       0.89      0.90      0.89       417
           1       0.93      0.92      0.92       582

    accuracy                           0.91       999
   macro avg       0.91      0.91      0.91       999
weighted avg       0.91      0.91      0.91       999

ROC AUC: 0.9605
```

*Fig 3.5 : Model Performance with CNN*5. Comparative Performance

5.Comparitive Performance

5.1. Binary Cross-Entropy Loss Evaluation

Two additional models (likely neural networks) were evaluated using the Binary Cross-Entropy (BCE) Loss:

```
Manual XGBoost Binary Cross-Entropy Loss (BCE Loss): 0.2422510842236655
Manual Random Forest Binary Cross-Entropy Loss (BCE Loss): 0.20674331422809547
```

Interpretation:

- BCE Loss quantifies the difference between predicted probabilities and actual labels. Lower values indicate better performance.
- Therefore, XGBoost (Loss = 0.2453) performs better in terms of prediction accuracy and confidence.

5.2 Comparative Performance

In this section, we evaluate and compare the performance of five well-established Convolutional Neural Network (CNN) architectures_VGG16,Inception, DensNet,EfiicientNet and MobileNet. Each of these models offers unique design philosophies and trade-offs in terms of accuracy, computational efficiency , and model complexity. By analyzing their performance on the same Dataset and under consistent training conditions, we aim to identify the most suitable architecture for pneumonia based on metrics such as accuracy , precision , recall and inference time .

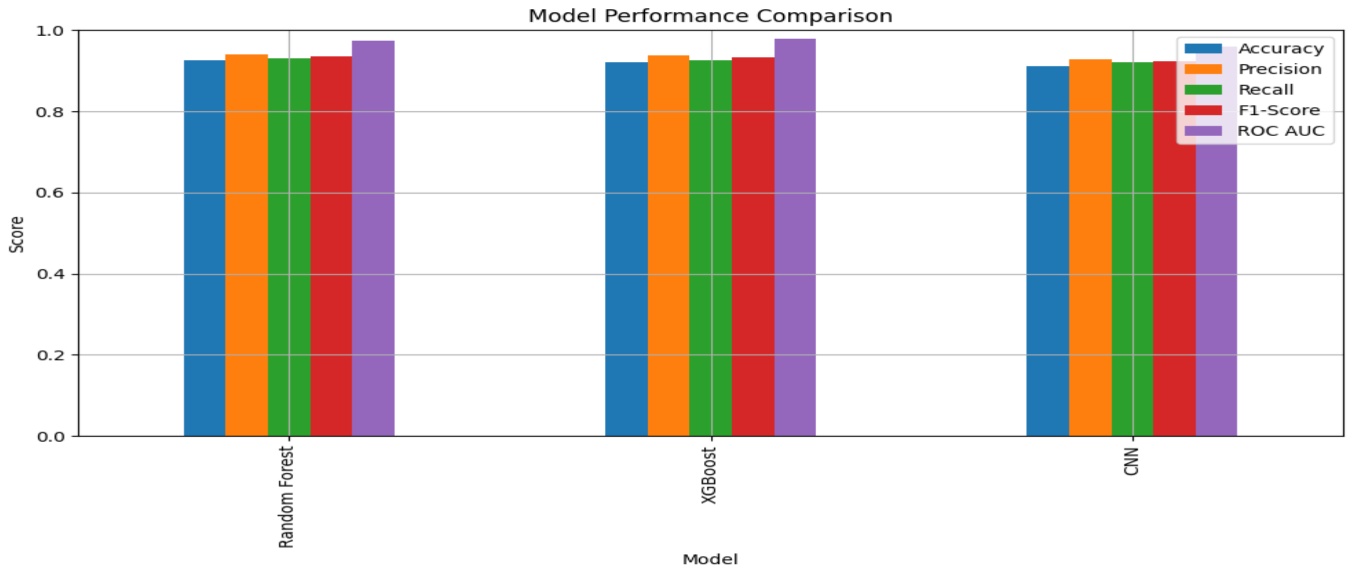


Fig 3.6 Metrics for the three models used in this study

Interpretation of Model Performance Results :

Table 3.3: comparative performance

Metric	CNN	Random Forest	XGBoost

Accuracy	91%	92%	92%
Precision (Class 1)	93%	94%	94%
Recall (Class 1)	92%	93%	93%
F1-Score (Class 1)	92%	94%	93%
ROC AUC	0.9605	0.9752	0.9780
False Positives	42	35	35
False Negatives	46	40	43

CNN demonstrated solid performance, particularly in classifying pneumonia cases, but had slightly higher false positive and false negative rates compared to the tree-based models. It achieved a ROC AUC of 0.9605, indicating good discriminative ability, though slightly lower than the other models.

Random Forest offered a well-balanced performance with improved accuracy, precision, and fewer misclassifications. Its ROC AUC of 0.9752 and low false positive/negative rates highlight its strength in both sensitivity and specificity.

XGBoost emerged as the top-performing model overall. It achieved the highest ROC AUC (0.9780) and consistently strong precision, recall, and F1-scores. Its confusion matrix showed excellent balance, with minimal false classifications, making it the most reliable of the three.

5.3 Result

While all three models demonstrated strong performance, XGBoost provided the most robust and consistent results across all evaluation metrics, closely followed by Random Forest. The CNN model also performed well, particularly in detecting pneumonia cases, but showed slightly lower performance in terms of overall accuracy and ROC AUC. Based on these findings, XGBoost is recommended as the most effective model for deployment in the pneumonia detection system.

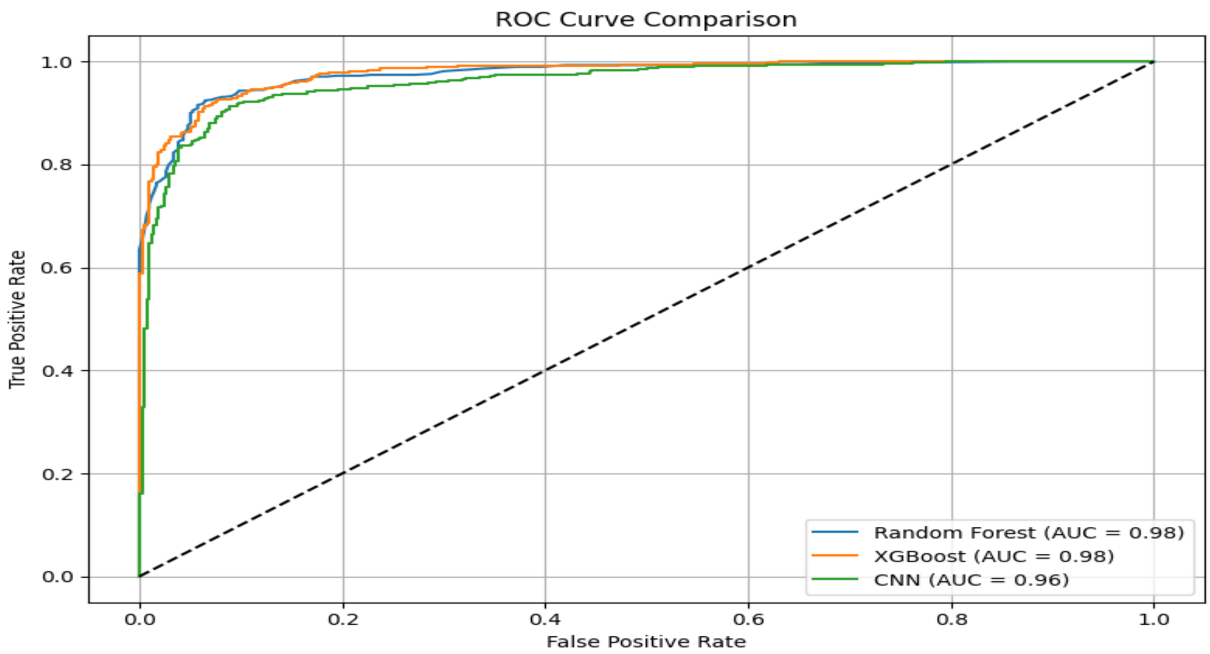


Fig 3.7 ROC Curve comparison

The Random Forest and XGBoost models both demonstrate nearly identical and excellent classification performance, with AUC = 0.98, indicating strong discrimination between pneumonia and non-pneumonia cases.

The CNN model, with a slightly lower AUC of 0.96, still performs well but shows slightly less effectiveness in distinguishing the two classes compared to the other models.

This visualization provides a clear, comparative view of the discriminative power of these models, supporting the earlier findings that Random Forest and XGBoost are more robust in this context.

6.Exploring Alternative CNN Architectures

While the initial Convolutional Neural Network (CNN) model demonstrated strong performance with an accuracy of 91% and a ROC AUC score of 0.9605, the evaluation revealed slightly higher misclassification rates compared to tree-based models such as Random Forest and XGBoost. In particular, the CNN recorded 42 false positives and 46 false negatives, which suggests potential room for improvement in both sensitivity and specificity.

To enhance the model's diagnostic accuracy and reduce misclassification, it is therefore appropriate to investigate more advanced CNN architectures that have demonstrated superior performance in image classification tasks. Architectures such as ResNet, DenseNet, and EfficientNet offer deeper networks with improved gradient flow and feature reuse, which may lead to better generalization and feature extraction in medical imaging contexts.

By adopting and evaluating these alternative CNN architectures, the objective is to improve overall classification performance—particularly in reducing false predictions—while maintaining computational efficiency and robustness in pneumonia detection.

6.1.VGG16

VGG16 is a classical deep CNN architecture introduced by Simonyan and Zisserman in 2014. It is characterized by its simplicity and depth, consisting of 13 convolutional layers followed by 3 fully connected layers. All convolutional layers use small receptive fields (3×3), stacked sequentially, allowing the network to learn increasingly complex features. Despite its high computational cost, VGG16 remains a powerful baseline for transfer learning due to its robust feature representation.

Depth : 16 weight layers

Key Feature : Uniform architecture with repeated block

Strength : Strong feature extractor with deep hierarchical learning

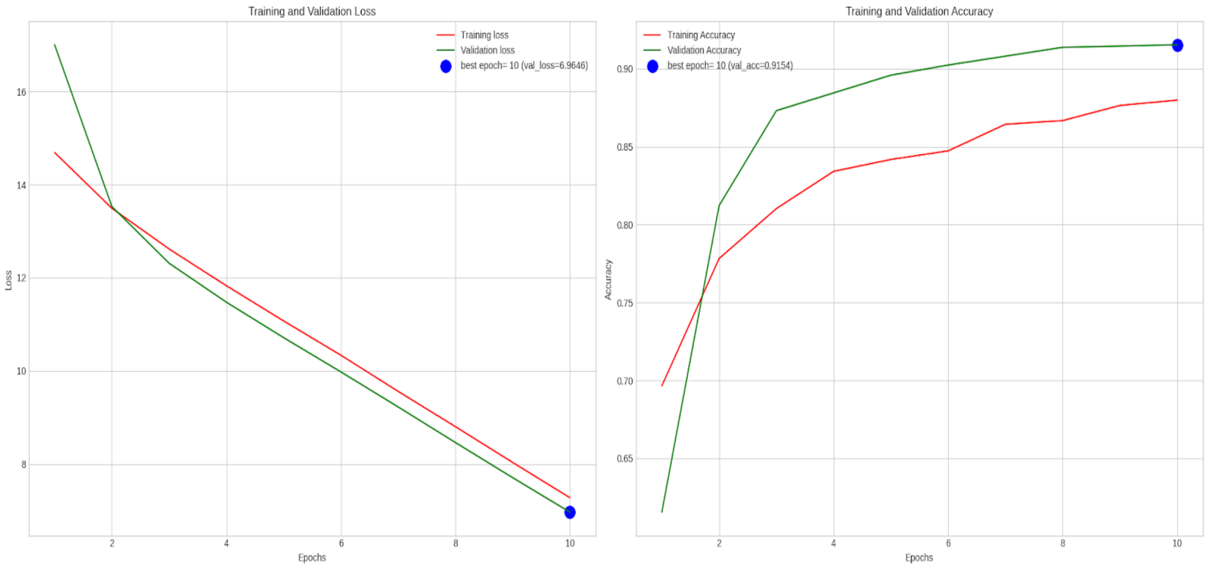


Fig 3.8 Learning Curves : Training vs. Validation Performance of VGG16

These curves show a well-trained CNN model with :

- Strong learning behavior
- No overfitting
- High final performance (91.5% val accuracy)

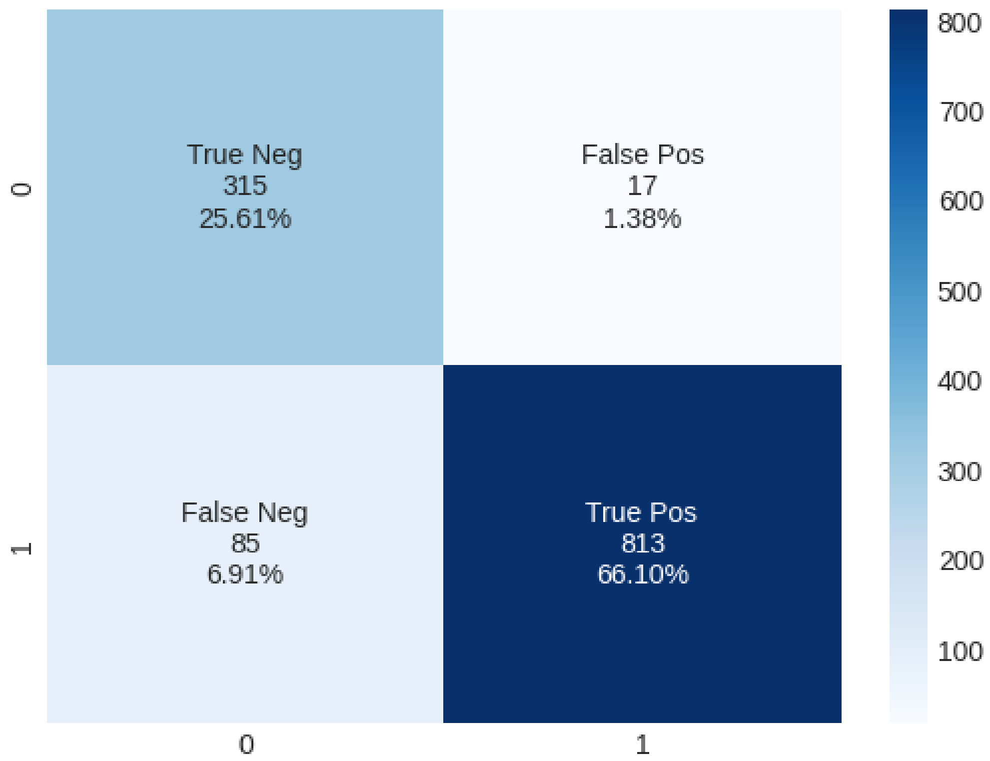


Fig 3.9 confusion matrix for VGG16

6.2 DenseNet (e.g., DenseNet121)

DenseNet, or Densely Connected Convolutional Network, introduces direct connections between any two layers with the same feature-map size. Each layer receives input from all preceding layers and passes its own feature maps to all subsequent layers, encouraging feature reuse and gradient flow. This structure improves learning efficiency and mitigates the vanishing gradient problem.

- Depth : Varies (e.g., DenseNet121 has 121 layers)
- Key Feature : Dense connections between layers
- Strength : Improved parameter efficiency and stronger gradient propagation

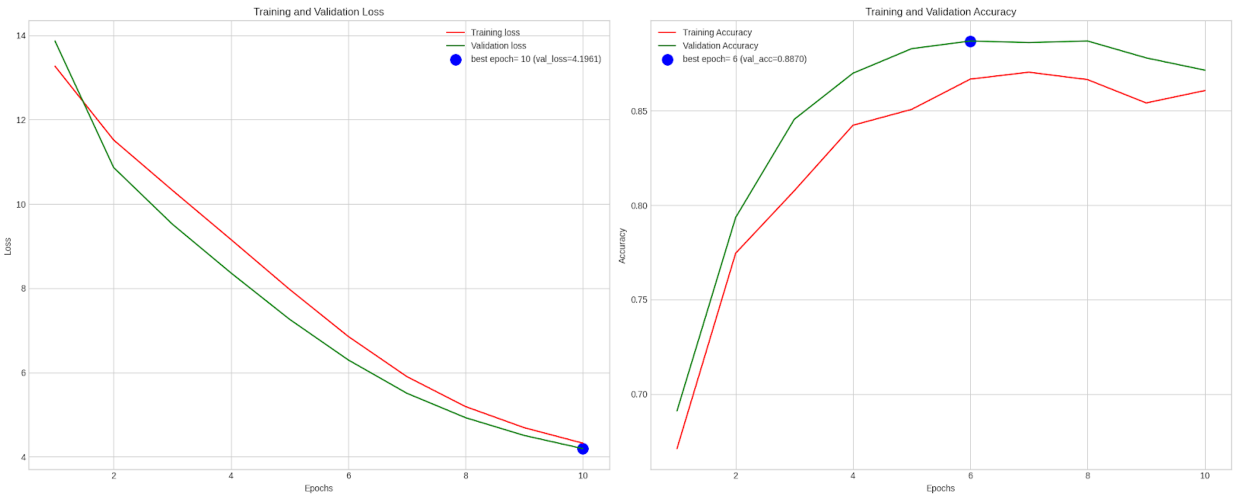


Fig 3.10 Learning Curves : Training vs. Validation Performance of DenseNet

Overall Accuracy: 87.1545%

Weighted Precision: 0.8998

Weighted Recall: 0.8715

Weighted F1-score: 0.8768

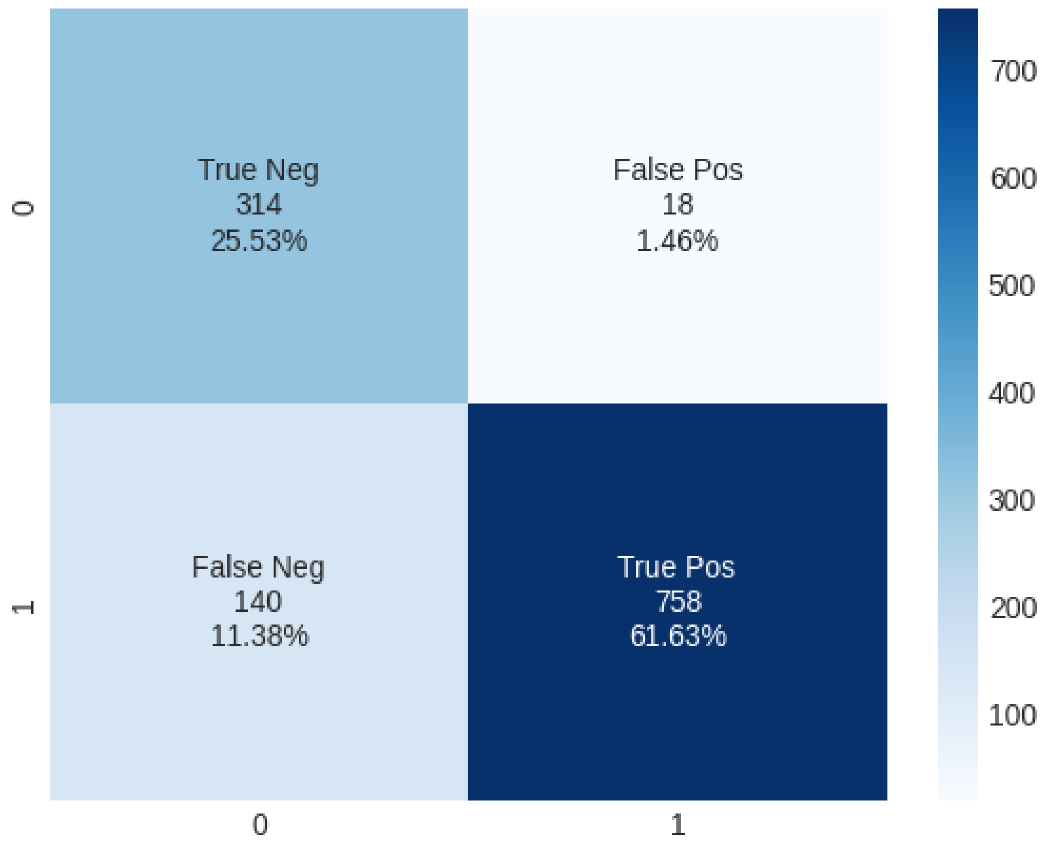


Fig 3.11 confusion matrix for DenseNet

6.3 InceptionV3

InceptionV3 is an advanced CNN architecture from Google's Inception family. It introduces Inception modules, which apply multiple types of convolutional filters (1×1, 3×3, 5×5) in parallel within the same layer, enabling the model to capture features at multiple scale

Additionally, it incorporates techniques such as factorized convolutions, batch normalization, and label smoothing for optimized performance.

- Depth: ~48 layers
- Key Feature: Multi-scale processing within Inception modules
- Strength: High accuracy with reduced computational complexity

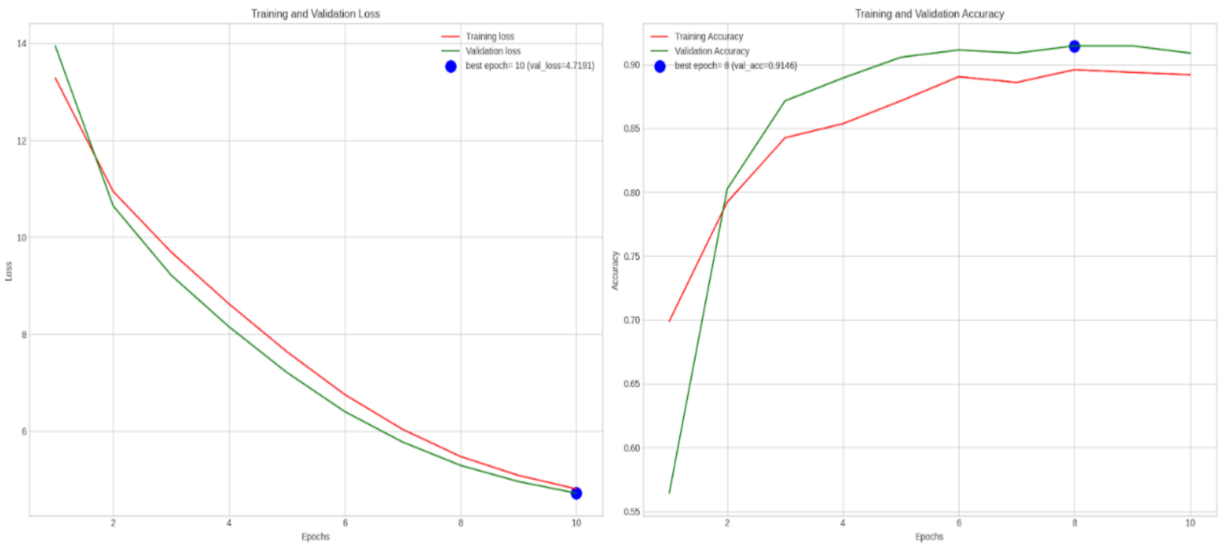


Fig 3.12 Learning Curves: Training vs. Validation Performance InceptionV3

Overall Accuracy: 90.8943%

Weighted Precision: 0.9218

Weighted Recall: 0.9089

Weighted F1-score: 0.9116

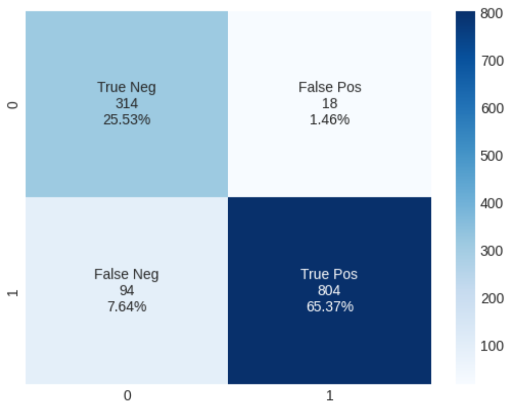


Fig 3.13 confusion matrix for InceptionV3

6.4 MobileNetV2

MobileNetV2 is a lightweight CNN architecture designed for mobile and embedded vision applications. It utilizes depthwise separable convolutions to reduce computational cost, and introduces inverted residual blocks with linear bottlenecks. These design choices allow the model to maintain competitive accuracy while significantly reducing the number of parameters and floating-point operations (FLOPs).

- Depth: ~53 layers
- Key Feature: Depthwise separable convolutions + inverted residuals
- Strength: Efficiency in low-resource environments with minimal accuracy loss

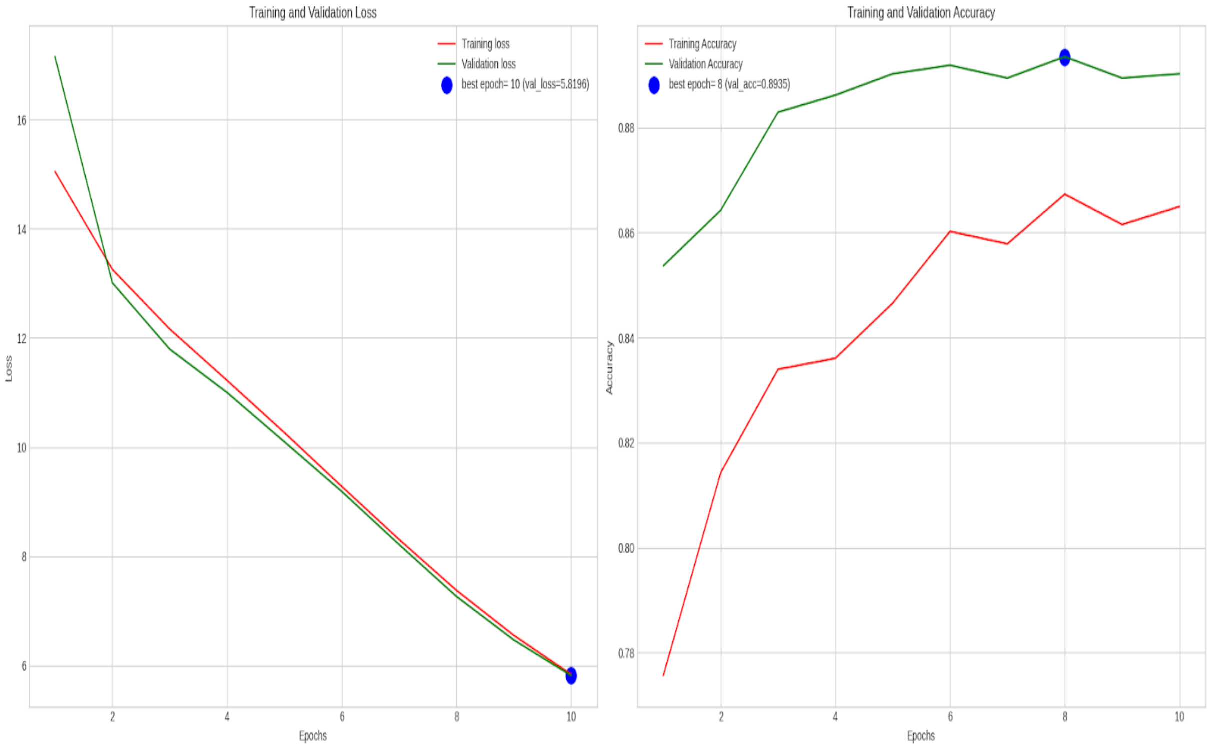


Fig 3.14 Learning Curves: Training vs. Validation Performance MobileNetV2

- Best epoch for loss: Epoch 10 (lowest validation loss).
- Overall Accuracy : 87.2358%
- Weighted Precision : 0.9023
- Weighted Recall : 0.8724
- Weighted F1-score : 0.8777

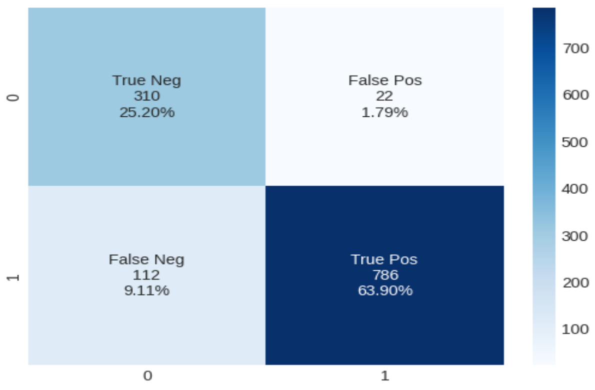


Fig_ 3.15 : Confusion Matrix Breakdown of MobileNetV2

6.5 EfficientNetB0

EfficientNetB0 is the baseline model from the EfficientNet family, which uses a compound scaling method to scale the network's width, depth, and input resolution

uniformly. Built on the Mobile Inverted Bottleneck Convolution (MBConv) and squeeze-and-excitation blocks, EfficientNet achieves superior performance with fewer parameters and less computational cost compared to earlier architectures.

- Depth: ~237 layers (but highly optimized)
- Key Feature: Compound model scaling and squeeze-and-excitation
- Strength: Best-in-class performance/efficiency tradeoff

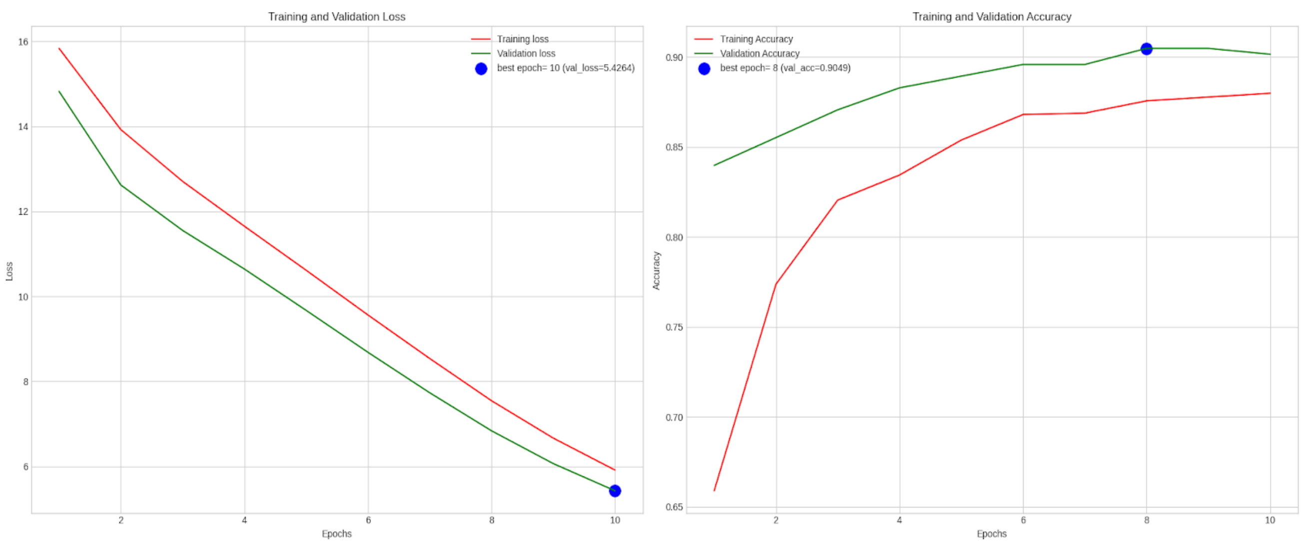


Fig 3.16 Learning Curves: Training vs. Validation Performance EfficientNetB0

- Overall Accuracy: 90.4878%
- Weighted Precision: 0.9236
- Weighted Recall: 0.9049
- Weighted F1-score: 0.9082

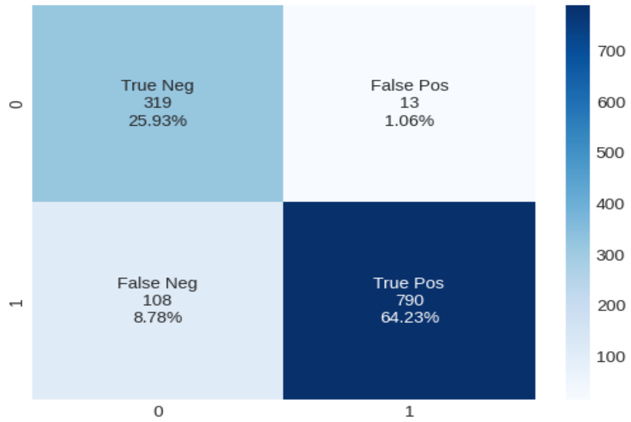


Fig 3.17: Confusion Matrix Breakdown EfficientNetB0

7. Model Training Details

To optimize model learning and ensure stable convergence, several hyperparameters and training strategies were carefully configured during the development of the CNN-based pneumonia detection system.

7.1. Optimizers and Learning Rate

Different CNN architectures were trained using the Adam optimizer or Adamax, depending on the experiment. Adam is widely used for its adaptive learning rate and computational efficiency. The initial learning rate was typically set to:

- 0.0001 for transfer learning (to avoid distorting pretrained weights)
- 0.001 in cases where the model was trained from scratch or fine-tuned

Learning rate scheduling was handled using ReduceLROnPlateau, which dynamically reduces the learning rate when the validation loss plateaus, improving fine-tuning performance.

7.2. Batch Size and Epochs

- Batch Size: 32
- Epochs: Models were trained for 10 epochs. However, the effective training duration was automatically controlled using early stopping.

These settings provided a balance between stability and efficiency during model convergence.

8. Hyperparameter Tuning

Hyperparameter tuning is crucial for maximizing model performance. In this project, a combination of manual tuning and empirical testing was used, guided by validation accuracy and loss.

8.1. Tuning Strategy

- Manual search was conducted over parameters such as dropout rate, number of dense units, learning rate, and choice of optimizer.
- Callback monitoring provided insight into optimal learning rate adjustments.
- Dropout rates were tested between 0.3 to 0.5 to reduce overfitting in dense layers.
- Dense layer sizes were tested in the range of 64 to 512 neurons.

  More advanced methods like grid search or Bayesian optimization were not applied due to resource constraints, but the manual tuning was effective given the use of transfer learning.

9. Validation Setup

Validation strategies were implemented to ensure model generalization and prevent overfitting.

9.1. Early Stopping

The training process used an EarlyStopping callback monitoring validation loss with a patience of 3.

- This helped in halting training when no improvement was observed, reducing training time and preventing overfitting.

9.2. Learning Rate Scheduling

- ReduceLROnPlateau was used to reduce the learning rate by a factor (e.g., 0.2) if validation loss stopped improving, allowing fine-tuned training in later stages.

9.3. Monitoring Loss Curves

- Training and validation accuracy/loss curves were plotted after each epoch.
- These plots were analyzed to detect signs of underfitting or overfitting, and to guide hyperparameter adjustments.

10.Comparative Performance of Models

In this section, we evaluate and compare the performance of five well-established Convolutional Neural Network (CNN) architectures—VGG16, Inception, DenseNet, EfficientNet, and MobileNet. Each of these models offers unique design philosophies and trade-offs in terms of accuracy, computational efficiency, and model complexity. By analyzing their performance on the same dataset and under consistent training conditions, we aim to identify the most suitable architecture for pneumonia detection based on metrics such as accuracy, precision, recall, and inference time.

Table 3.4 : Performance Comparison of Deep Learning Models: Accuracy, Precision, Recall, and F1-Score

	model	accuracy (%)	precision	recall	f1
0	VGG16	91.71	0.927689	0.917073	0.919293
1	Inception	90.89	0.921826	0.908943	0.911607
2	EfficientNetB0	90.16	0.919911	0.901626	0.905035
3	MobileNet	89.11	0.908484	0.891057	0.894685
4	DenseNet	87.15	0.899830	0.871545	0.876833

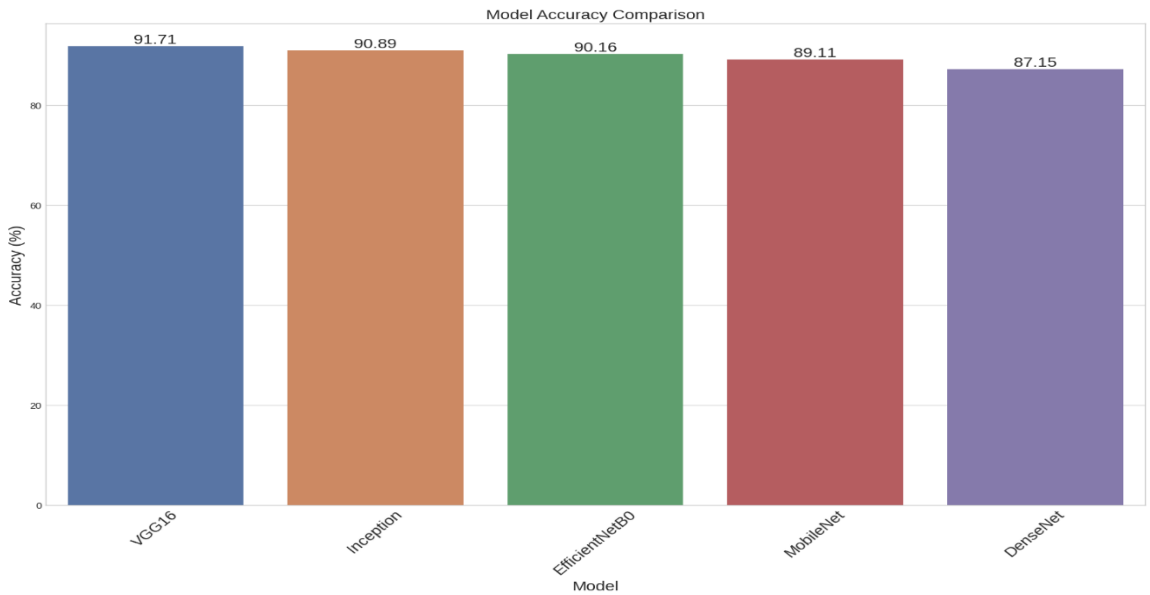


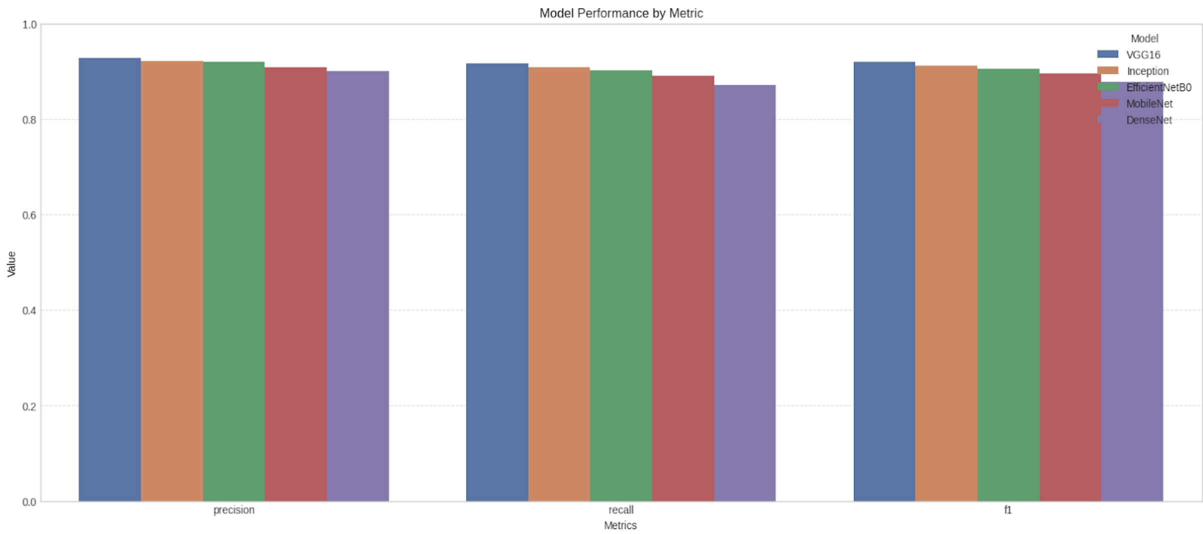
Fig 3.18 Model Accuracy Comparison

10.1 Model Comparison Overview

As shown in the figure 3.13, all models performed comparably well, with performance metrics consistently above 0.88, reflecting strong predictive capabilities for binary classification (pneumonia vs. normal).

- VGG16 slightly outperformed others in all three metrics, indicating better generalization on unseen data. Its architectural simplicity combined with fine-tuning contributed to stable and accurate predictions.
- InceptionV3 followed closely, showing a balanced performance across precision and recall, leveraging its multi-scale feature extraction blocks.
- EfficientNetB0, although known for computational efficiency, trailed slightly behind in recall, suggesting it may have missed some positive pneumonia cases in favor of precision.
- MobileNetV2 performed well but marginally lower than others, which might be attributed to its lightweight structure and reduced capacity to capture complex patterns in X-ray images.
- DenseNet121, despite its deep connectivity, recorded slightly lower metrics compared to VGG16 and InceptionV3. This could be due to overfitting or lack of sufficient regularization in this configuration .

To evaluate the effectiveness of various transfer learning-based CNN architectures for pneumonia detection from chest X-rays, a comparative performance analysis was conducted across VGG16, InceptionV3, EfficientNetB0, MobileNetV2, and DenseNet121. The evaluation metrics considered were precision, recall, and F1-score.



- The *Fig 3.19 Model Performance by Metric*

10.2 Interpretation of Metrics

- **Precision** : VGG16 and InceptionV3 yielded the highest precision, suggesting a lower false positive rate. This is crucial in medical diagnostics where false positives can lead to unnecessary anxiety or treatment.

- **Recall** : All models performed well, with VGG16 leading. High recall indicates fewer pneumonia cases were missed, which is essential for early detection and treatment.

- **F1-Score** : The F1-score, a harmonic mean of precision and recall, shows VGG16 again slightly ahead, confirming its balanced and robust performance.

The use of multiple architectures allowed for a robust comparative study, and VGG16 emerged as the most effective model under the current training configuration. However, performance differences are minimal, and trade-offs between model size, inference speed, and accuracy should be considered for deployment.

10.3 Strengths and Weaknesses of Each Model

To better understand the relative performance and suitability of each CNN architecture, the following summarizes their key strengths and weaknesses:This comparative analysis helps in selecting the most appropriate model depending on the deployment scenario: VGG16 for accuracy-critical tasks, MobileNetV2 for resource-constrained environments, and EfficientNetB0 for a balance between performance and efficiency.

Model	Strengths	Weakness
VGG16	• Simple and well-understood architecture • High performance in precision, recall, and F1-score • Performs well with minimal tuning	•Computationally expensive due to many parameters • High memory usage •Lacks architectural innovations like skip connections or bottlenecks
InceptionV3	• Efficient multi-scale feature extraction • Good balance between depth and cost • Strong performance across all metrics	•Complex architecture, harder to fine-tune •Slightly slower inference time
EfficientNetB0	•Optimized for both accuracy and efficiency • Compact model and fast inference • Scalable to larger versions	• Slightly lower recall, may miss some pneumonia cases •Needs careful preprocessing (normalization/scaling)

MobileNetV2	•Lightweight, mobile friendly Fast training and inference • Decent performance with fewer parameters	• Lower capacity to learn complex patterns • Slightly weaker recall and F1-score
DenseNet121	• Dense connections promote feature reuse • Deep with fewer parameters than plain CNNs	• May overfit small datasets • Requires more training time and GPU memory than lightweight models

Table 3.5 : Strengths and weakness of Each Model

11. Fine-Tuning best model: EfficientNet

Found 3806 validated image filenames belonging to 2 classes.

Found 1230 validated image filenames belonging to 2 classes.

Found 820 validated image filenames belonging to 2 classes.

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 4s 202ms/step - accuracy: 1.0000 - loss: 1.0386
20/20 ━━━━━━━━━━━━━━━━━━━━ 4s 179ms/step - accuracy: 0.9776 - loss: 1.0639
20/20 ━━━━━━━━━━━━━━━━━━━━ 5s 236ms/step - accuracy: 0.9820 - loss: 1.0793
Train Loss:  1.037268042564392
Train Accuracy:  1.0
--------------------
Validation Loss:  1.065765142440796
```

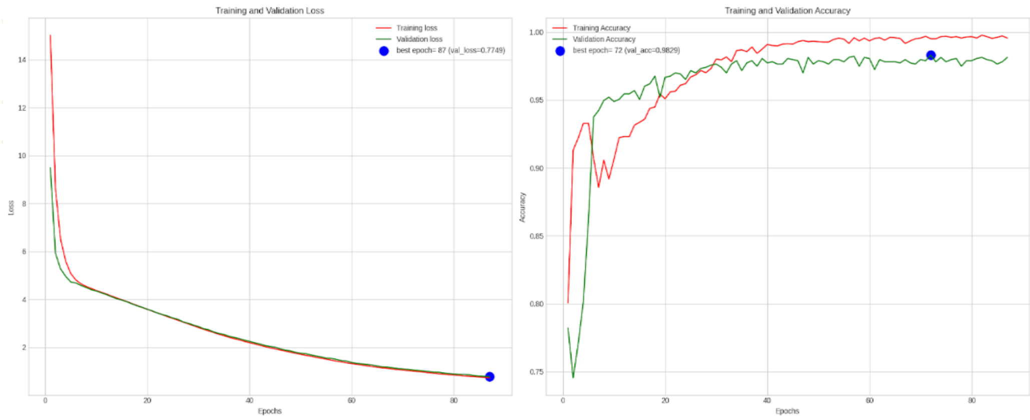


Fig 3.20: Learning Curves: Training vs. Validation Performance EfficientNetB0

12 Prediction

12.1 Prediction on valid data

To evaluate the generalization capability of the trained models, predictions were made on the validation dataset. This dataset was not seen during training and thus provides an unbiased estimate of the model's performance

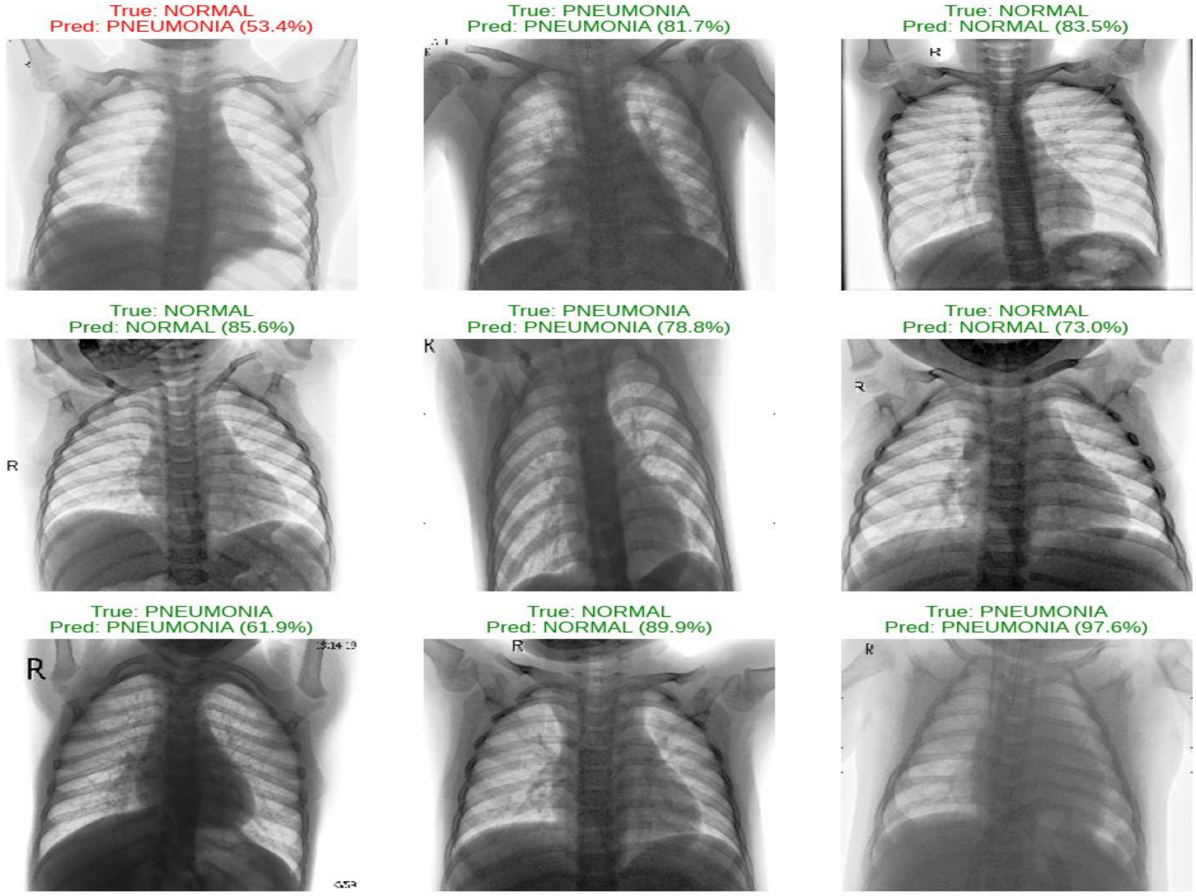


Figure 3.21 : predicted samples on validation data

12.2 Prediction on test data

After evaluating the models on the validation dataset, the final performance was assessed using the test dataset. This set was entirely held out during both training and validation phases, ensuring a fair and unbiased measure of the model's real-world performance.

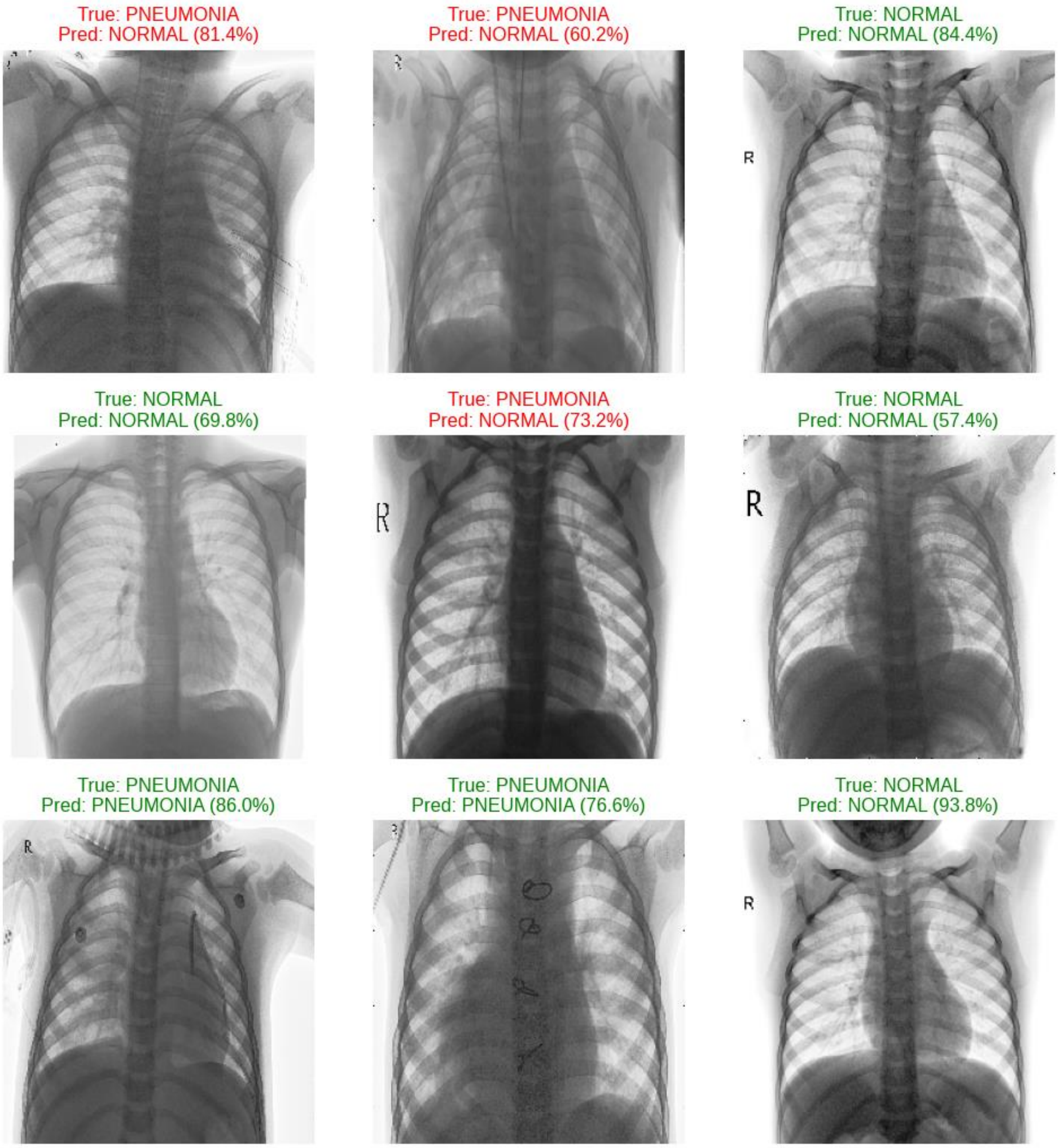
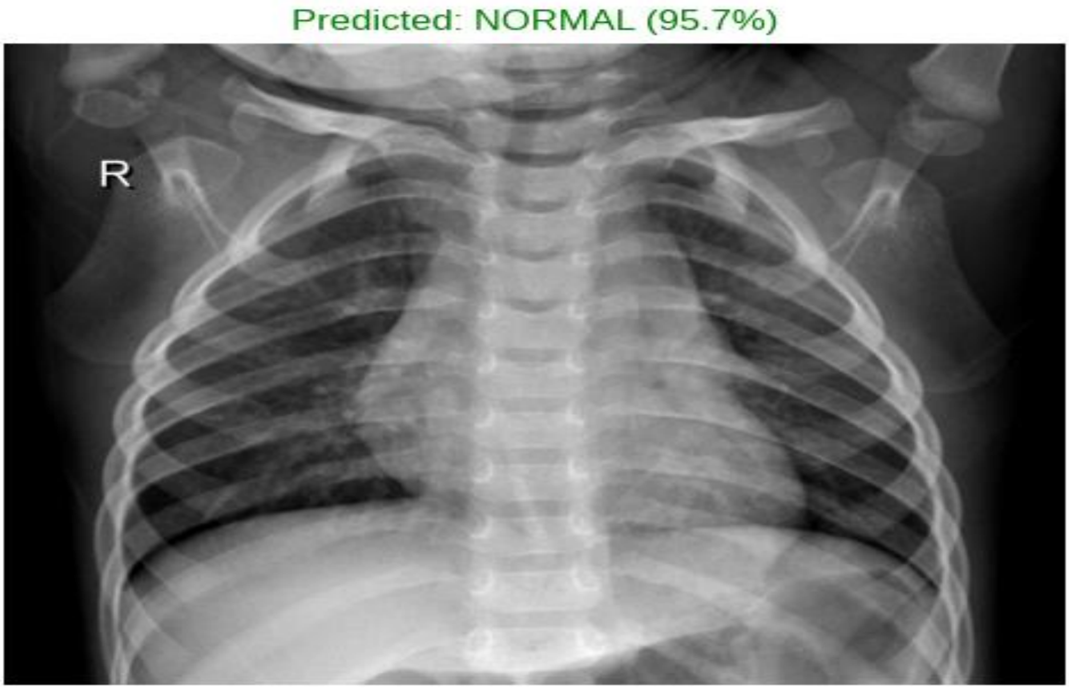


Figure 3.22 : predicted samples on test data

12.3 Prediction on URL image

To demonstrate the practical applicability of the trained model, an image was retrieved from an external URL and used as a test case for prediction. This real-world scenario simulates how the model might be used in deployment to analyze unseen medical images.

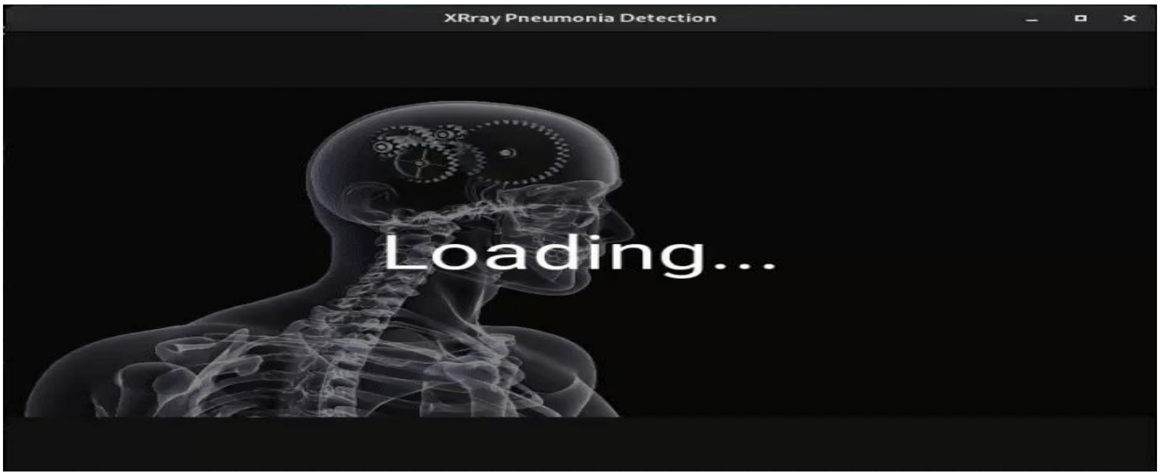


The successful prediction on an image from an external URL indicates that the model can generalize well beyond the training dataset and can be integrated into web-based or mobile health applications for automated diagnosis.

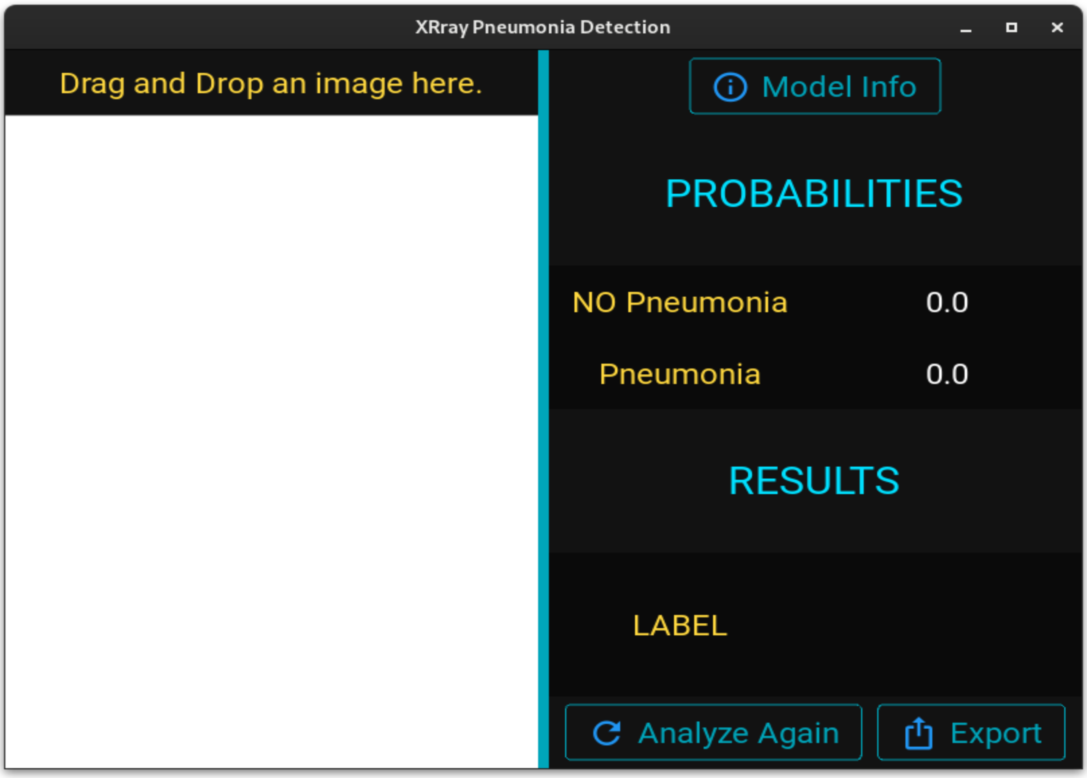
13. Employment

After a thorough evaluation of multiple machine learning and deep learning techniques—including Random Forest, XGBoost, and Convolutional Neural Networks (CNNs)—the model that achieved the best performance in pneumonia detection was EfficientNetB0. To make this model accessible and practical for real-world use, we developed a user-friendly interface.

13.1 Loading Screen



13.2 Menu



The developed interface provides an intuitive and efficient environment for pneumonia detection using chest X-ray images. It is structured into two main sections:

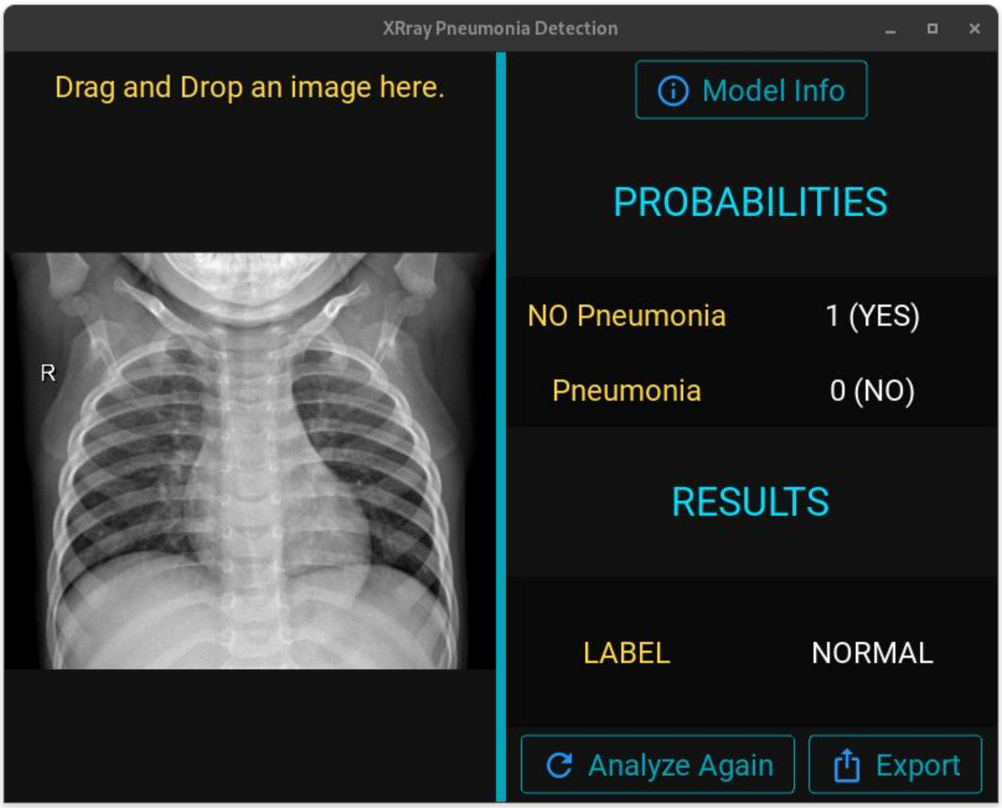
- Image Upload Panel: Located on the left, it allows users to drag and drop chest X-ray images for analysis, simplifying the interaction process.
- Results and Controls Panel: On the right, it displays prediction probabilities for both classes (*Pneumonia* and *No Pneumonia*), the final diagnostic label, and provides control buttons:
  - Model Info to view details about the EfficientNetB0 model.
  - Analyze Again to restart the analysis process.
  - Export to save the prediction results.

The interface employs a dark theme with contrasting color highlights (cyan and yellow) to enhance usability and visual clarity. It is designed for local deployment and provides fast, reliable predictions in a user-friendly format.

13.3 Prediction

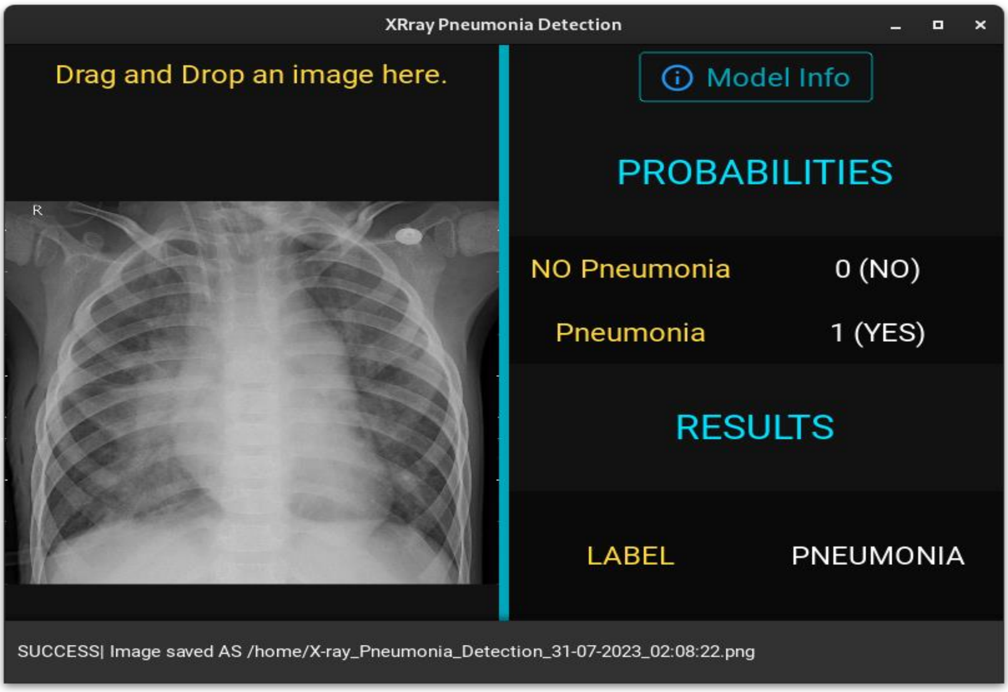
13.3.1 Pneumonia

This image shows the interface of our application designed for pneumonia detection from chest X-rays. It displays the analysis results of a chest radiograph, where the AI model has classified the image as "NORMAL," indicating no signs of pneumonia. The probability scores show full confidence (100%) in the "No Pneumonia" prediction. Such tools are used to support healthcare professionals in the rapid and accurate diagnosis of lung infections.

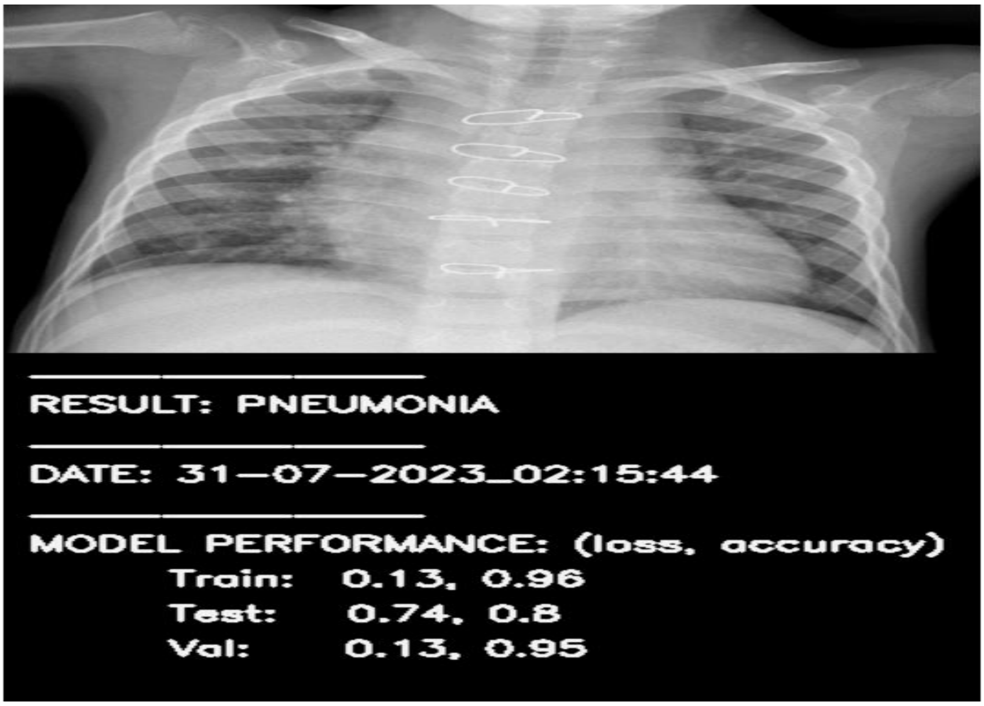


13.3.2 Normal

The interface displays a chest radiograph that has been analyzed by the model, which classified it as showing signs of pneumonia. The probability scores reflect high confidence in the presence of pneumonia, with a value of 1 (YES) for “Pneumonia” and 0 (NO) for “No Pneumonia.” This demonstrates the potential of artificial intelligence in assisting medical professionals with early and accurate diagnosis of respiratory infections.



The exported image should appear as follows :



14.Conclusion

In this chapter, we presented a complete pipeline for automated pneumonia detection using chest X-ray images. The process began with rigorous dataset preprocessing to ensure data quality and consistency. We then evaluated multiple machine learning and deep learning models, including Random Forest, XGBoost, and five CNN architectures. After a thorough performance comparison, EfficientNetB0 emerged as the most accurate and robust model.

To translate these results into a practical tool for clinical use, we developed a user-friendly interface enabling healthcare professionals to upload chest X-rays and receive real-time diagnostic predictions. This interface, powered by the EfficientNetB0 model, offers a reliable, fast, and accessible solution that bridges the gap between AI research and medical application.

General Conclusion

This project aimed to develop an accurate and practical system for the detection of pneumonia from chest X-ray images. We explored multiple approaches—starting with classical machine learning models like Random Forest and XGBoost, and advancing to more sophisticated deep learning techniques using Convolutional Neural Networks.

Five CNN architectures were tested and compared, with EfficientNetB0 emerging as the most effective in terms of accuracy and generalization. This model was ultimately selected for integration into a functional detection platform.

The final result is a comprehensive pipeline: from data preprocessing and model evaluation to the deployment of an intelligent interface. This work not only highlights the potential of AI in medical imaging but also lays a foundation for future enhancements, such as cloud deployment, database integration, or support for multi-disease detection.

References

[1] https://www.geeksforgeeks.org/

[2] https://www.geeksforgeeks.org/decision-tree/

[3] https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

[4] https://en.wikipedia.org/wiki/Main_Page

[5] https://www.datacamp.com/

[6] https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

[7] https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

[8]https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSeq-9h9kBuckdFv-ccMoaRW2FY_Ttd789BsWyfywomKHKat_dK

[9]https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSeq-9h9kBuckdFv-ccMoaRW2FY_Ttd789BsWyfywomKHKat_dK

[10] https://developers.google.com/machine-learning/crash-course?hl=fr

[11] https://developers.google.com/machine-learning/crash-course?hl=fr

[12] https://www.geeksforgeeks.org/

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