AN INTERGRATED DEEP LEARNING AND MACHINE LEARNING APPROACH FOR PREDICTING PNEUMONIA

Komal[1], Harpreet Kaur[1]
[1]DAV Institute of Engineering & Technology, Jalandhar, Punjab, India

Abstract

Healthcare systems worldwide strive for timely and accurate diagnostics to improve patient outcomes. Within this domain, the chest houses critical organs such as the heart and lungs, and maintaining chest health ensures efficient respiration, optimal oxygen exchange, and reduced systemic disease burden, forming a cornerstone of preventive and curative care. However, chest diseases remain a major global health challenge: pneumonia inflames the air sacs causing fluid accumulation with cough, fever, and breathing difficulty. Traditional manual detection by radiologists is error-prone due to inter-observer variability, time-consuming, and costly, limiting accessibility. Artificial intelligence, especially machine learning, deep learning, and computer vision, bridges these gaps by enabling rapid, consistent, and automated analysis of medical images. Yet existing techniques have limitations: standard convolutional neural networks struggle with long-range spatial dependencies and overfit on small datasets; vision transformers and their variants demand extensive pre-training data and high computational resources, hindering generalization in medical imaging. To address these issues, we propose a hybrid EfficientNetB0–XGBoost framework. The model leverages EfficientNetB0 pre-trained on ImageNet as a robust feature extractor, applying selective layer freezing to retain generic features while fine-tuning deeper layers. A lightweight classification head with global average pooling and dropout is appended during training. After fine-tuning, deep features are extracted as 1280-dimensional vectors and used to train an XGBoost classifier, which learns complex decision boundaries through gradient boosting. The final pipeline sequentially applies feature extraction and XGBoost classification for two categories: Pneumonia, and normal. The models demonstrate strong performance, with precision, recall, and F1-score all exceeding 95%, indicating reliable classification capability. Robustness is validated by ROC and precision-recall curves.

Keyword: Pneumonia, deep learning, machine leaning.

1. Introduction

Healthcare is very important for human life, and correct diagnosis at the right time helps in proper treatment [1]. The chest contains important organs like the lungs, which help us in breathing and oxygen supply. If the lungs do not work properly, it can affect the whole body and cause serious health problems [2][3]. So, keeping the lungs healthy is very important for a good life [4].One common and serious lung disease is pneumonia. In this disease, the air sacs in the lungs get infected and may fill with fluid. This causes symptoms like cough, fever, chest pain, and difficulty in breathing [5]. Pneumonia is a major cause of death, especially in children and older people, so early detection is very important [6].Normally, doctors check chest X-ray images to detect pneumonia. But this process is done manually and has some problems. It takes a lot of time, needs expert doctors, and sometimes mistakes can happen [7]. In many areas, there are not enough doctors, which makes diagnosis slower and difficult [8]. So, there is a need for an automatic system that is fast and accurate.Artificial Intelligence (AI), especially Machine Learning (ML) and Deep Learning (DL), is now widely used in medical image analysis [9]. Convolutional Neural Networks (CNNs) are very useful for finding patterns in X-ray images and detecting diseases [10]. However, these models may not perform well when the dataset is small and can sometimes give overfitting problems [11]. Also, advanced models need more data and high computing power, which is not always available [12].To solve these problems, this study uses a hybrid model. In this method, a pre-trained EfficientNetB0 model is used to extract important features from chest X-ray images. The early layers are kept fixed, while the last layers are trained to learn pneumonia-related patterns [13]. A simple classification layer is used during training, and then features are extracted from the model. These features are then given to an XGBoost classifier, which helps in making better and more accurate predictions [14]. This combination improves the overall performance of the model. The proposed system classifies chest X-ray images into two categories: Pneumonia and Normal. The model performance is checked using accuracy, precision, recall, F1-score, and confusion matrix to ensure good and reliable results.

2. Literature Review

In recent years, many researchers have focused on using deep learning techniques for detecting pneumonia from chest X-ray images. These methods help doctors in faster and more accurate diagnosis. Different models like transformers, CNNs, EfficientNet, VGG, and ResNet have been widely used to improve performance.In paper [15], the authors used a **Vision Transformer (ViT)** model for pneumonia detection. Unlike traditional CNNs, transformer models can understand global relationships in images. This means they can look at the whole image and find patterns more effectively. The study showed that transformers can improve accuracy by capturing important features that may not be easily detected by normal CNN models. However, transformer models require large datasets and high computational power, which can be a limitation in real-world medical applications where data is limited.In paper [16], the authors proposed an **EfficientNet-based ensemble model** for lung disease detection. EfficientNet is known for its ability to provide high accuracy while using fewer parameters. In this study, multiple EfficientNet models were combined together in an ensemble approach. This means the predictions from several models were merged to get better results. The ensemble method improved the overall accuracy and made the system more robust. However, combining multiple models increases complexity and training time, which can make the system slower.In paper [17], the authors used **deep convolutional neural networks (CNNs)** for chest X-ray image classification. CNNs are one of the most commonly used models in image processing because they can automatically extract important features from images without manual effort. The study showed that CNNs can achieve high accuracy in detecting diseases from chest X-rays. However, CNNs may suffer from overfitting when the dataset is small. Overfitting means the model performs well on training data but poorly on new unseen data.In paper [18], the authors used the **EfficientNetV2L model** for pneumonia detection. This model is an improved version of EfficientNet and provides better performance and faster training. The study achieved high accuracy, showing that EfficientNetV2L is very effective for medical image classification. The model can capture both simple and complex patterns in X-ray images. However, like other deep learning models, it still requires a good amount of data and computational resources.In paper [19], the authors proposed a **custom CNN model** for pneumonia detection. They focused on solving common problems like overfitting and class imbalance. Class imbalance happens when one class (for example, normal images) has more samples than another class (pneumonia). To solve this, they used techniques like data balancing, dropout, and optimization methods. Dropout helps in reducing overfitting by randomly ignoring some neurons during training. Their model achieved very high training and validation accuracy, showing that simple CNN models can also perform well if proper techniques are used.In paper [20], the authors used a **VGG-16 based deep learning model** for pneumonia detection. VGG-16 is a well-known CNN architecture that uses multiple layers to extract detailed features from images. In this study, the model was combined with neural networks to improve classification performance. The results showed that this approach achieved higher accuracy compared to traditional machine learning methods like Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and Naïve Bayes. This proves that deep learning models are more effective for image-based medical diagnosis.In paper [21], the authors proposed an **enhanced ResNet-50 model** for pneumonia detection. ResNet is a deep learning model that uses skip connections to solve the problem of vanishing gradients in deep networks. The authors improved the model by adding techniques like multi-feature fusion, denoising, and attention mechanisms. These improvements helped the model focus on important regions of the X-ray image and ignore unnecessary noise. As a result, the model achieved better accuracy and robustness compared to standard methods.From all these studies, it is clear that deep learning models play an important role in pneumonia detection from chest X-ray images. Each model has its own advantages. Transformer models are good at capturing global features, CNNs are effective for feature extraction, EfficientNet models provide high accuracy with fewer parameters, and ResNet models improve performance in deeper networks. However, there are still some challenges.Many models require large datasets and high computational power, which may not always be available. Some models are too complex, making them difficult to use in real-world applications. Also, many approaches rely only on deep learning and do not combine it with machine learning techniques, which could further improve performance.Therefore, there is a need for a simple, efficient, and accurate approach that can work well on limited data and provide better results. Combining deep learning for feature extraction and machine learning for classification can help in improving performance. This idea forms the basis of the proposed hybrid model in this study.

3. Research Gap

Many researchers have applied Artificial Intelligence (AI) and deep learning techniques to detect pneumonia from chest X-ray images. These methods have improved the speed and accuracy of diagnosis, but several important challenges still remain.Most existing studies use deep learning models such as Convolutional Neural Networks (CNNs), EfficientNet, and Vision Transformers for pneumonia detection [15]–[17]. These models are very powerful and can automatically extract important features from medical images. They help in identifying disease patterns that may not be easily visible to the human eye. However, these models usually require a large amount of labeled data and high computational resources. In real-world healthcare settings, especially in small hospitals or rural areas, such large datasets and advanced hardware are often not available. Because of this, the practical use of these models becomes limited.Some studies have proposed more advanced architectures like EfficientNetV2 and ResNet to improve the accuracy of pneumonia detection [18], [21]. These models achieve high performance and better feature representation. However, they are complex in design and require more memory and processing time. This increases the cost and makes them difficult to deploy in real-time clinical environments. Therefore, even though these models perform well in research, their real-world application is still challenging.Other researchers have used simpler models such as VGG-16 and basic CNN architectures for pneumonia detection [19], [20]. These models are easier to implement and require less computational power compared to advanced models. However, they often suffer from overfitting, especially when trained on small datasets. Overfitting means the model learns the training data very well but fails to perform accurately on new, unseen data. This reduces the reliability of the model in real clinical use.Some recent works have also explored ensemble and hybrid approaches to improve performance [16]. These methods combine multiple models to achieve better accuracy and robustness. While ensemble methods can improve results, they also increase the complexity of the system and require more training time. This makes the system harder to manage and less efficient for real-time applications. Another important limitation in many existing studies is that they rely only on deep learning models for both feature extraction and classification. Although deep learning is very effective, using it alone may not always give the best results. Machine learning algorithms like XGBoost are known for their strong decision-making ability and can handle complex patterns more efficiently. However, the combination of deep learning for feature extraction and machine learning for classification has not been fully explored in many studies. In addition, many models are designed as end-to-end systems, where feature extraction and classification are done together. This reduces flexibility and makes it difficult to improve individual parts of the system. Separating feature extraction and classification can help in better optimization and improved performance. Therefore, there is a clear need for a simple, efficient, and reliable model that can work well even with limited data, reduce overfitting, and require less computational power. Such a model should also combine the strengths of deep learning and machine learning to improve overall performance. A hybrid approach that uses EfficientNetB0 for feature extraction and XGBoost for classification can help address these gaps and provide better results for pneumonia detection.

3.1 Objective of the study

1. To build a system that combines deep learning and machine learning to classify chest X-ray images into Normal and Pneumonia.
2. To use a pre-trained EfficientNetB0 model to extract important features and patterns from chest X-ray images.
3. To use the XGBoost algorithm to accurately classify images into Normal and Pneumonia classes.
4. To evaluate the model using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

4. Methodology

The complete workflow of proposed model is shown in figure 1.

4.1 Data Collection

The dataset collection step is the foundation of the proposed system, as the performance of the model highly depends on the quality and organization of the data. In this study, a chest X-ray image dataset is used, which contains images belonging to two categories: Pneumonia and nomal. These images are collected and stored in separate folders, where each folder represents a specific class. This structured arrangement helps in easy identification and labeling of images. Each image is associated with its corresponding class label, which is essential for supervised learning. The dataset is then converted into a structured format using a dataframe, where each row contains the file path of the image and its corresponding label. This format makes it easier to process and manage the dataset during training. To ensure that the model learns effectively from all classes, the dataset is split into training, validation, and testing sets using stratified sampling. This ensures that each class is equally represented in all subsets. The training set is used for learning, the validation set is used for monitoring performance, and the test set is used for final evaluation.
Proper dataset collection and organization help in building a reliable and unbiased model.

4.2 Data pre-processing and augmentation

Data preprocessing is an important step that prepares the raw X-ray images for input into the deep learning model. Since the images may vary in size, resolution, and quality, it is necessary to standardize them. In this study, all images are resized to a fixed size of 224x224 pixels to match the input requirements of EfficientNetB0. This ensures uniformity across the dataset and allows the model to process the images efficiently. After resizing, normalization is applied to adjust pixel values into a suitable range. Without normalization, the model may take longer to learn or may produce poor results. Preprocessing also helps remove unnecessary variations in the data, allowing the model to focus only on important features. Data augmentation is used to increase the diversity of the training dataset and improve the model's ability to generalize. In medical imaging, datasets are often limited, which can lead to overfitting. Overfitting occurs when a model performs well on training data but fails on new, unseen data. To overcome this problem, different augmentation techniques are applied to the training images. These include rotation, zooming, and horizontal flipping. Rotation helps the model learn from images taken at different angles, while zooming allows it to handle variations in size and scale. Horizontal flipping helps the model recognize patterns regardless of orientation. These transformations create new variations of existing images without changing their actual meaning. It is important to note that augmentation is applied only to the training dataset and not to validation or test datasets, to ensure fair evaluation. By using data augmentation, the model becomes more robust and can perform better in real-world conditions.

4.3 Base model initialization (Transfer learning)

The base model is initialized by loading EfficientNetB0 pre-trained on ImageNet, with include top is False to discard the original classification head. This establishes a transfer learning framework where knowledge from a large-scale dataset is repurposed for the target task. For an input image $I \in \mathbb{R}^{260\times260\times3}$, the model functions as a parametric feature extractor $\Phi(I; \Theta_{\text{base}})$, mapping the input to a deep feature representation. The architecture consists of a hierarchy of inverted bottleneck convolutions, culminating in a feature map with spatial dimensions downsampled by a factor of 32 relative to the input. By leveraging pre-trained weights Θ^*_{base} learned from over 14 million images, the model encodes rich hierarchical features from edges to complex patterns. This initialization provides a high-quality starting point, significantly reducing the need for target-domain data and training time. Formally, the extracted feature vector for an input image is given in below mentioned equation before global pooling.

$$f = \Phi(I; \Theta^*_{\text{base}}) \in \mathbb{R}^{8\times8\times1280} \tag{1}$$

Consequently, the model serves as a robust feature extractor, capturing semantically meaningful representations that are subsequently refined or directly used by downstream classifiers. This approach ensures that generic visual concepts are retained, enabling efficient adaptation to the specific classification task.

4.4 Layer Freezing (Fine-Tuning Strategy)

In the fine-tuning strategy, selective layer freezing is employed to balance the preservation of generic features with domain-specific adaptation. Let the EfficientNetB0 base model consist of L total layers, indexed from 1 to L . The first $L - 30$ layers (earlier convolutional blocks) capture universal low- and mid-level features such as edges, textures, and simple patterns. These layers are frozen, meaning their parameter Θ_{frozen} remain fixed at the pre-trained ImageNet weights $\Theta_{ImageNet^*}$:

$$\Theta_{\text{frozen}} = \{\theta_i \mid i = 1,2, \ldots , L - 30\},\ \nabla_{\theta_i}\mathcal{L} = 0 \tag{2}$$

Conversely, the last 30 layers (deeper blocks) are set as trainable, allowing them to adapt to the target dataset's high-level semantics. This partition is mathematically expressed as:

$$\Theta_{trainable} = \{\theta_i \mid i = L - 29, L - 28, \ldots , L\} \tag{3}$$

During backpropagation, gradients are only computed for $\Theta_{trainable}$ and the newly appended classification head parameters Θ_{head}. This strategy reduces the risk of overfitting on a limited target dataset while enabling the model to learn task-specific representations. The overall optimization objective becomes:

$$\min_{\Theta_{trainable}\,\Theta_{head}} \mathcal{L}_{CE}(\hat{y}, y), \quad with\ \Theta_{frozen}\ fixed \qquad (4)$$

By freezing the early layers, the model retains robust, generalizable features extracted from ImageNet, ensuring stable and efficient training convergence.

4.5 Custom classification head

A lightweight custom classification head is appended to the frozen base model to enable taskspecific prediction. First, a Global Average Pooling (GAP) layer aggregates the 4D feature map $F \in \mathbb{R}^{\mathbb{H}\times\mathbb{W}\times\mathbb{C}}$ into a 1D vector by averaging across spatial dimensions:

$$g = \frac{1}{H \cdot W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_{i,j,:} \in \mathbb{R}^{\mathbb{C}} \qquad (5)$$

A Dropout layer with rate $p = 0.4$ is then applied to randomly zero out elements, reducing overfitting: $(d = g \odot m)$, where m is a Bernoulli mask with $(\mathbb{P}(m_k = 0)\ becomes\ 0.4)$. Finally, a fully connected layer with softmax activation maps the dropout output to class probabilities:

$$z = Wd + b, \qquad \hat{y}_k = \frac{\exp(z_k)}{\sum_{j=1} \exp(z_j)} \qquad (6)$$

where $W \in \mathbb{R}^{\mathbb{K}\times\mathbb{C}}$ and $b \in \mathbb{R}^{\mathbb{K}}$ are trainable parameters, and K is the number of classes. This head is jointly optimized with the last 30 layers of the base model, enabling the network to learn discriminative representations tailored to the target dataset while maintaining regularization through dropout.

4.6 Model compilation

During model compilation, the classification network is configured for training by specifying the optimizer, loss function, and evaluation metric. The categorical cross-entropy loss is employed to measure the discrepancy between predicted probabilities ($\hat{y}$ and ground-truth onehot encoded labels y:

$$\mathcal{L}_{C} = -\sum_{k=1}^{K} y_k \log(\widehat{y_k}) \qquad (7)$$

where K denotes the number of classes. To minimize this loss, the Adam optimizer is selected with a learning rate $\eta = 1 \times 10^{-4}$. Adam adapts parameter updates by maintaining exponentially decaying averages of past gradients \(\mathbf{m}_t\) and squared gradients v_t:

$$\Theta_{t+1} = \Theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{v_t} + \epsilon} \qquad (7$$

where Θ_t represents the trainable parameters (the last 30 layers of the base model and the custom classification head). Accuracy is set as the primary evaluation metric, measuring the proportion of correctly classified samples. This compilation step ties together the model architecture with the optimization objective, establishing the training dynamics for subsequent fine-tuning. The choice of a low learning rate ensures stable convergence by preserving the pre-trained features while allowing gradual adaptation to the target domain.

4.7 Model training

During model training, the compiled network is optimized over up to \(T=100\) epochs using mini-batch gradient descent. At each iteration, a batch of (B) images is passed through the network to compute the categorical cross-entropy loss:

4.8 Feature extraction

During feature extraction, a separate model \(\Psi\) is constructed to output the penultimate layer's activations, specifically the Global Average Pooling (GAP) features before the dropout and classification head. For an input image \(\mathbf{x}\), the feature vector is obtained as:

$$f = \Psi(x) = \mathrm{GAP}\left(\Phi(x; \Theta^{*}_{base})\right) \in R^{1280} \tag{11}$$

where \Φ represents the frozen or fine-tuned EfficientNetB0 base model. This mapping is applied to every batch in the training and test generators. Formally, for a generator $\mathcal{G}$ with N batches, the complete feature set F_{set} and corresponding labels Y_{set} are aggregated as:

$$F_{set} = \bigoplus^{N}{}_{i=1} \Psi(X_i)\ , \quad Y_{set} = \bigoplus^{N}{}_{i=1} Y_i \tag{12}$$

where $\mathcal{X}_i$ and $\mathcal{Y}_i$ denote the i-th batch of images and one-hot labels, and \(\bigoplus\) denotes vertical stacking. The extracted features X_{train} and X_{test} serve as compact, high-level representations, effectively distilling each image into a 1280-dimensional vector. This process decouples the deep feature learning stage from the subsequent XGBoost classification, enabling efficient training of the gradient-boosted classifier on precomputed features.

4.9 Label transformation

During label transformation, the one-hot encoded label matrices extracted from the data generators are converted into integer class indices required by the XGBoost classifier. For a given batch, the one-hot labels $Y_{one-hot} \in \{0,1\}^{B\times K}$ satisfy $\sum^{K}{}_{k=1} y_{i,k} = 1$ for each sample i.
The transformation applies the argmax operation along the class dimension: $y_i = \underset{k}{\mathrm{argmax}}\ y_{i,k},\ i = 1,2, \ldots, B$ (13)

yielding a vector $y_{int} \in \{0,1, \ldots, K-1\}^{B}$. This is applied to all extracted feature–label pairs, converting the training labels Y_{train} and testing labels Y_{test} from one-hot encoding to a compact integer representation. Mathematically, the process can be expressed as:

$$y_{int} = argmax_k(Y_{one-hot}, axis = 1) \tag{14}$$

This transformation ensures compatibility with the XGBoost API, which expects integer class labels for multi-class classification. It also reduces memory footprint and simplifies subsequent evaluation metrics such as accuracy, where direct comparison between predicted and groundtruth integer labels is performed.

4.10 XGBoost classifier training

During XGBoost classifier training, the extracted deep features $X_{train} \in R^{N\times 1280}$ and corresponding integer labels $y_{train} \in \{0, \ldots, K-1\}^{N}$ are used to learn an ensemble of decision trees. The model predicts the prediction score for binary classification via an additive function:

$$\hat{y}_i = \sum_{m=1}^{M} f_m(x_i), \qquad f_m \in \mathcal{F} \tag{15}$$

where F is the space of regression trees, and $M = 600$ is the number of trees. The objective function combines a convex loss l (binary classiication) with a regularization term Ω to control complexity:

$$L = \sum_{i=1}^{N} \mathrm{l}(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(f_m), \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda |w|2 \tag{16}$$

with T the number of leaves, w leaf weights, γ and λ regularization parameters. Training employs gradient boosting: at each iteration, a new tree is fitted to the negative gradient of the loss, scaled by a learning rate $\eta = 0.03$. The max depth is set to 8 to limit tree complexity, and the model is trained using the XGBoost library's efficient parallel implementation. This yields a robust classifier that captures non-linear relationships among the deep features, achieving high predictive accuracy on the test set.

4.11 Hybrid predict pipeline

During the **hybrid prediction pipeline**, the trained feature extractor Ψ and XGBoost classifier h are combined to generate predictions for the test set. For each test batch

$\mathcal{X}_{\text{test}}{}^{(i)}$ $with(B)$ samples, the deep features are first extracted:

$$F^{(i)} = \Psi(\mathcal{X}_{\text{ts}}{}^{(i)}) \in \mathbb{R}^{\mathbb{B}\times 1280} \tag{17}$$

These features are then passed to the XGBoost classifier, which outputs the predicted integer class for each sample:

$$\hat{y}^{(i)} = h(F^{(i)}) \in \{0,1, \dots, K-1\}^B \tag{18}$$

The predictions $\hat{y}^{(i)}$ from all batches are concatenated to form the complete prediction vector

$\hat{Y}_{\text{test}}$. Simultaneously, the ground-truth one-hot labels from each batch are converted to integer indices and concatenated to yield \(\mathbf{Y}_{\text{test}}\). This process can be summarized as:

$$\hat{Y}_{\text{test}} = \bigoplus_{Mi=1} h\left(\Psi(\mathcal{X}_{\text{ts}(i)})\right), Y_{\text{ts}} = \bigoplus_{Mi=1} \operatorname{argmax}(\mathcal{Y}_{\text{ts}(i)}) \tag{19}$$

where M denotes the number of batches. The pipeline leverages the CNN's ability to extract high-level visual features and XGBoost's capacity to model complex decision boundaries, ensuring efficient and accurate inference. This decoupled architecture allows the two components to operate sequentially without backpropagation through the gradient-boosted classifier, enabling fast predictions on new data.

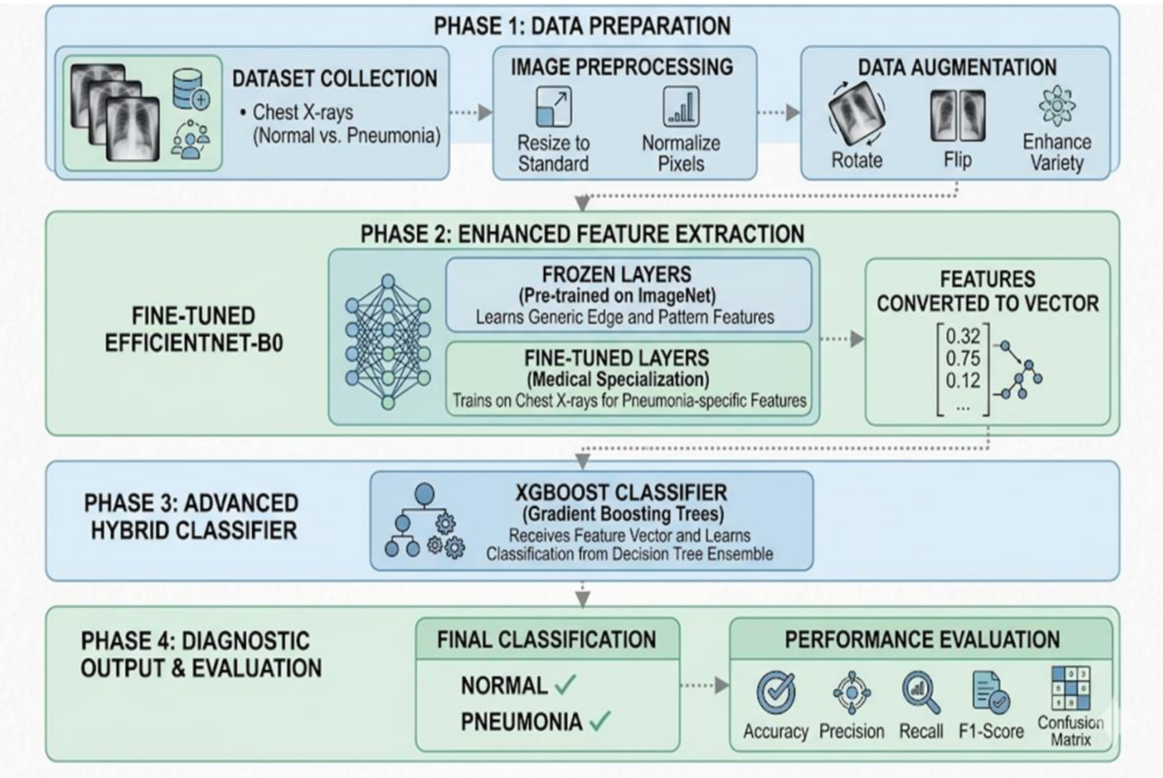

Figure 1: Workflow of the proposed hybrid deep learning framework

4. Results and Discussion

The results and discussion section presents the performance of the proposed model for classifying chest X-ray images into Normal and Pneumonia. The model is evaluated using different metrics such as accuracy, loss, ROC curve, PR curve, and confusion matrix.

4.1 Training and validation Accuracy curves

The blue line shows training accuracy, which slowly increases from about 98.1% to 99.3%, meaning the model is learning very well from the training data, and small ups and downs are normal. The orange line shows validation accuracy, which first drops a little, then increases quickly up to around 98.8% at epoch 3–4, and after that it slightly decreases and becomes stable around 98%. Both lines stay close to each other, which is a good sign, but after epoch 4 the training accuracy keeps improving while validation accuracy stops improving, showing a small sign of overfitting, although the overall performance is still very good.

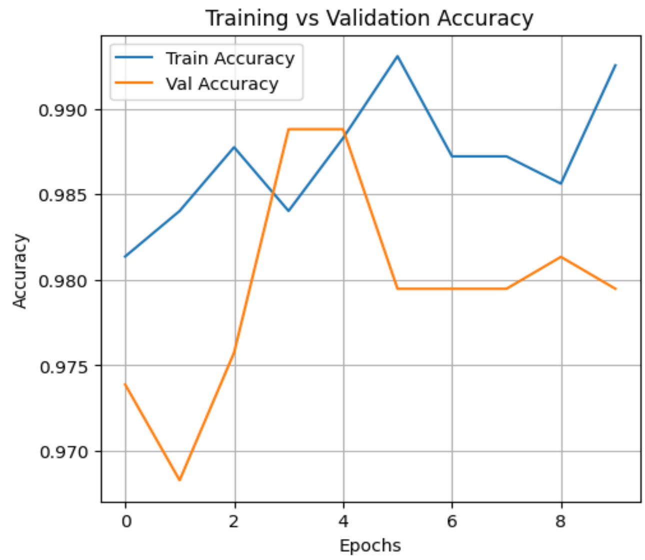

Figure 2: Training and validation accuracy curve of proposed model

4.2 Training and validation Loss curves

The training and validation loss were studied over 10 epochs to see how the model improves. The training loss keeps decreasing from 0.046 to 0.024, which means the model is learning well and making fewer mistakes on training data. The validation loss becomes lowest at around epoch 4, which shows the model performs best on new data at this point. After epoch 4, the validation loss starts increasing slowly while training loss continues to decrease. This means the model is starting to overfit, or memorize the training data too much. So, the best performance is achieved around epoch 4, and it is better to stop training early or use techniques to reduce overfitting.

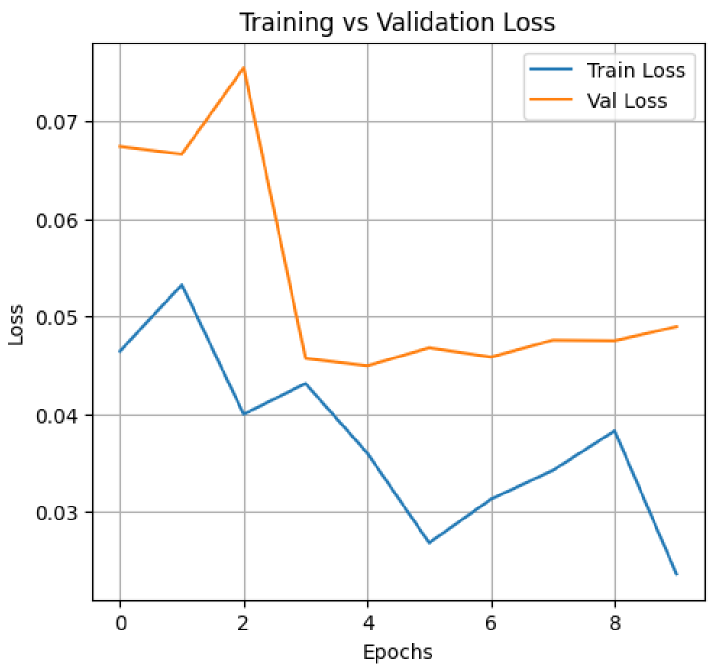

Figure 3: Training and validation loss curve of proposed model .

4.3 ROC Curve

The model's performance was also checked using the ROC curve. The result showed an AUC value of 0.99, which means the model can almost perfectly tell the difference between Normal and Pneumonia cases. The curve shows that the model correctly identifies most positive cases (high true positive rate) while making very few mistakes (low false positives). This high AUC value proves that the model is very reliable and can accurately detect the disease with very little error.

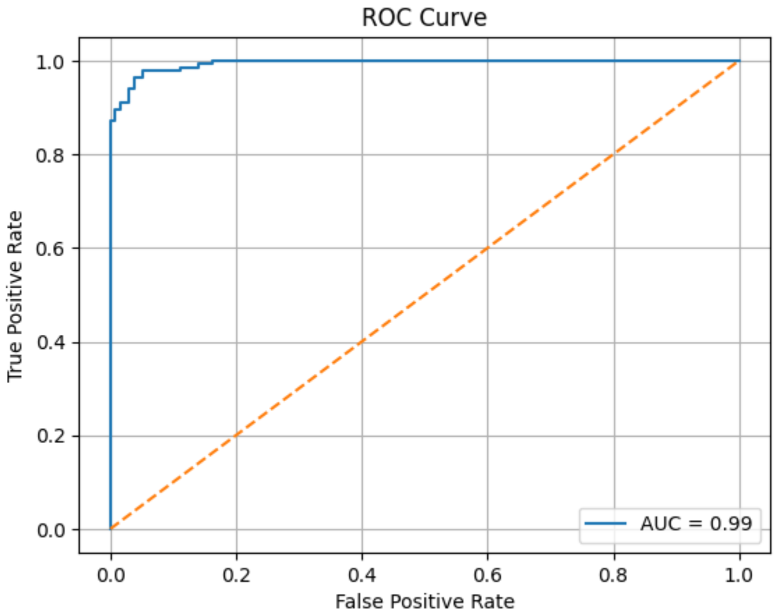

Figure 4: ROC curve of proposed model .

4.4 PR Curve

The model's performance was also tested using the Precision-Recall (PR) curve to better understand how well it detects pneumonia cases. The PR-AUC value is 0.99, which means the model has both high precision and high recall. This shows that the model can correctly identify most pneumonia cases (high recall) while making very few wrong predictions (high precision). The curve also shows that the model stays accurate even when we try to detect more positive cases. Overall, this means the model is very reliable and effective for identifying pneumonia from chest X-rays.

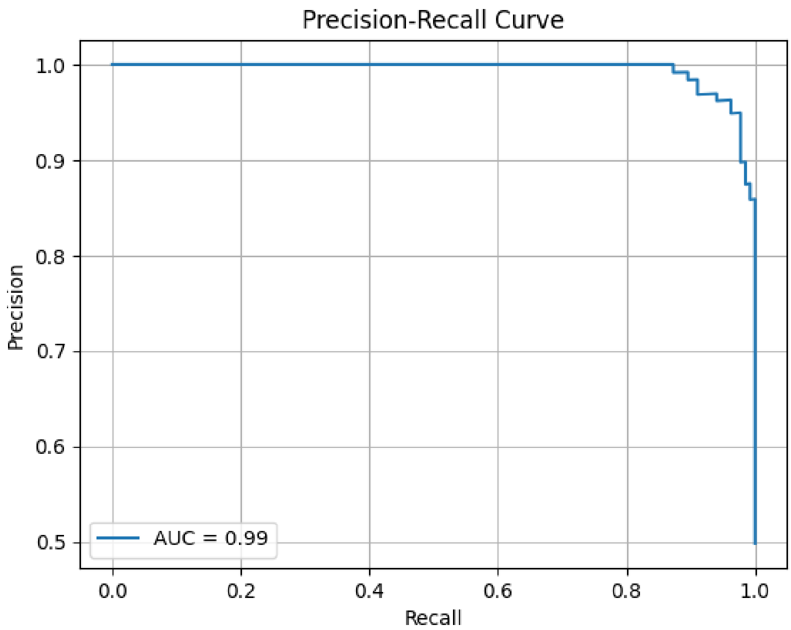

Figure 5: Precision recall curve of proposed model.

4.5 Confusion Matrix

The confusion matrix shows how well the model classifies Normal and Pneumonia cases. Out of all Normal cases, 131 were correctly predicted as Normal, while 4 were wrongly predicted as Pneumonia. Similarly, out of Pneumonia cases, 125 were correctly identified, while 9 were wrongly predicted as Normal. This means the model makes very few mistakes and correctly classifies most of the cases. Overall, the high number of correct predictions compared to errors shows that the model has high accuracy and good performance in detecting both Normal and Pneumonia cases.

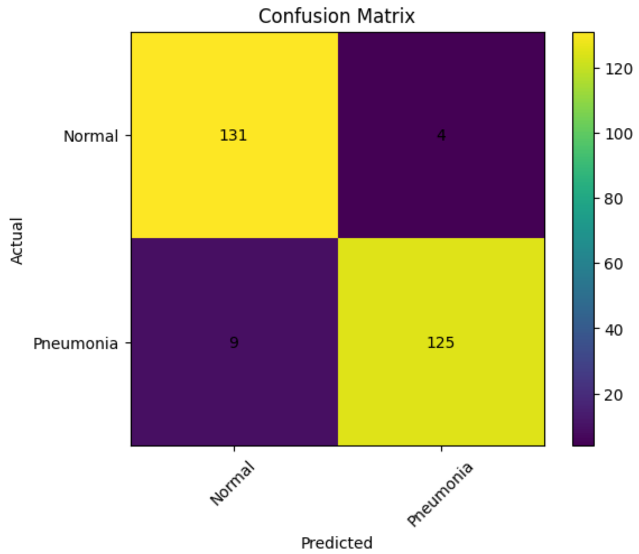

Figure 7: Confusion matrix of proposed model

4.6 Class-wise comparative analysis of proposed model

This table (as shown in Figure 8) explains how well the model works for both Normal and Pneumonia cases using different measures. For Normal cases, the accuracy is 0.97, which means the model is correct 97% of the time. The precision is 0.94, so when the model predicts Normal, it is usually correct. The recall is 0.97, which means it can find most of the actual Normal cases. The F1-score is 0.95, showing a good balance between precision and recall. For Pneumonia cases, the accuracy is also 0.97, showing strong overall performance. The precision is 0.97, meaning the model is very accurate when predicting Pneumonia. The recall is 0.93, which means it detects most Pneumonia cases but misses a few. The F1-score is 0.95, again showing a good balance. Overall, this figure shows that the model is very accurate and reliable with only a small number of errors.

Table 1: Class-wise comparative analysis of proposed model.

Class	**Accuracy**	**Precision**	**Recall**	**F1-Score**
normal	0.97	0.94	0.97	0.95
Pneumonia	0.97	0.97	0.93	0.95

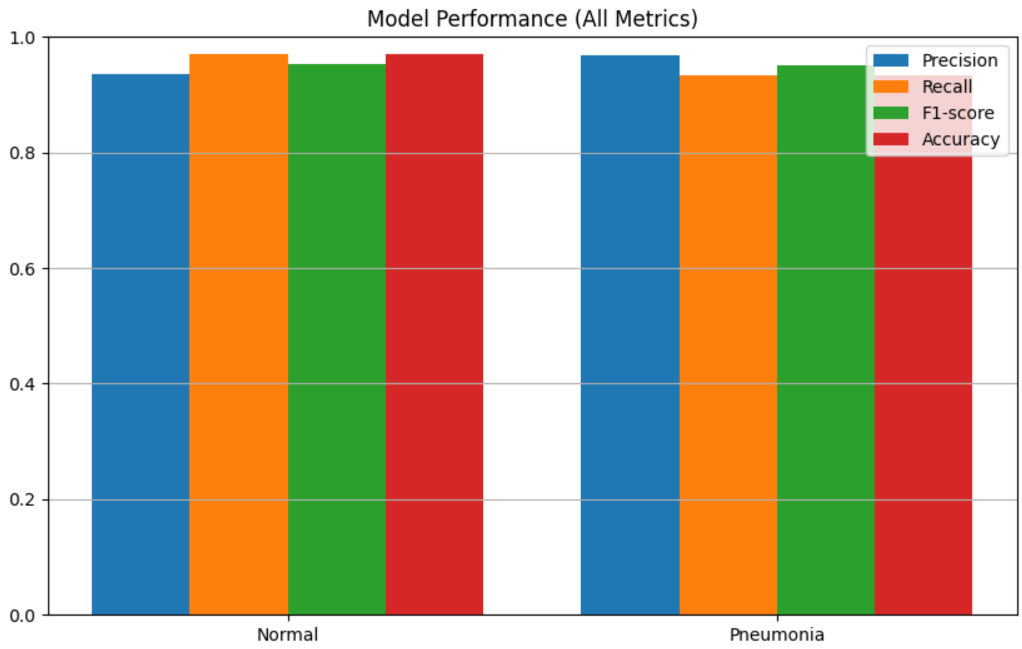

Figure 8: Class-wise comparative analysis of proposed model based on Accuracy, Precision, Recall, and F-1 score.

4.7 Comparative analysis of proposed model with existing techniques

The image 9 presents a comparative analysis of different techniques—EfficientNetV2L, Custom CNN, VGG16, ResNet50, and the Proposed Model—based on accuracy, precision, recall, and F1-score. The Proposed Model outperforms all others, achieving the highest scores (around 95%) across all metrics, indicating superior and consistent performance. EfficientNetV2L and VGG16 show strong intermediate results, while Custom CNN and ResNet50 exhibit comparatively lower performance, highlighting the effectiveness of the proposed approach, as shown in table 2.

Table 2: Comparison of proposed model with existing techniques based on performance metrics such as Accuracy, Precision, Recall and F-1 Score.

Techniques	**Accuracy**	**Precision**	**Recall**	**F1-Score**
EfficientNetV2L [18]	94.02	94.02	94.02	94.02
Custom CNN [19]	88.78	88.78	88.78	88.78
VGG16 [20]	91.66	91.66	91.66	91.66
ResNet50 [21]	87.98	87.98	87.98	87.98
Proposed model	**95.20**	**95.20**	**95.20**	**95.20**

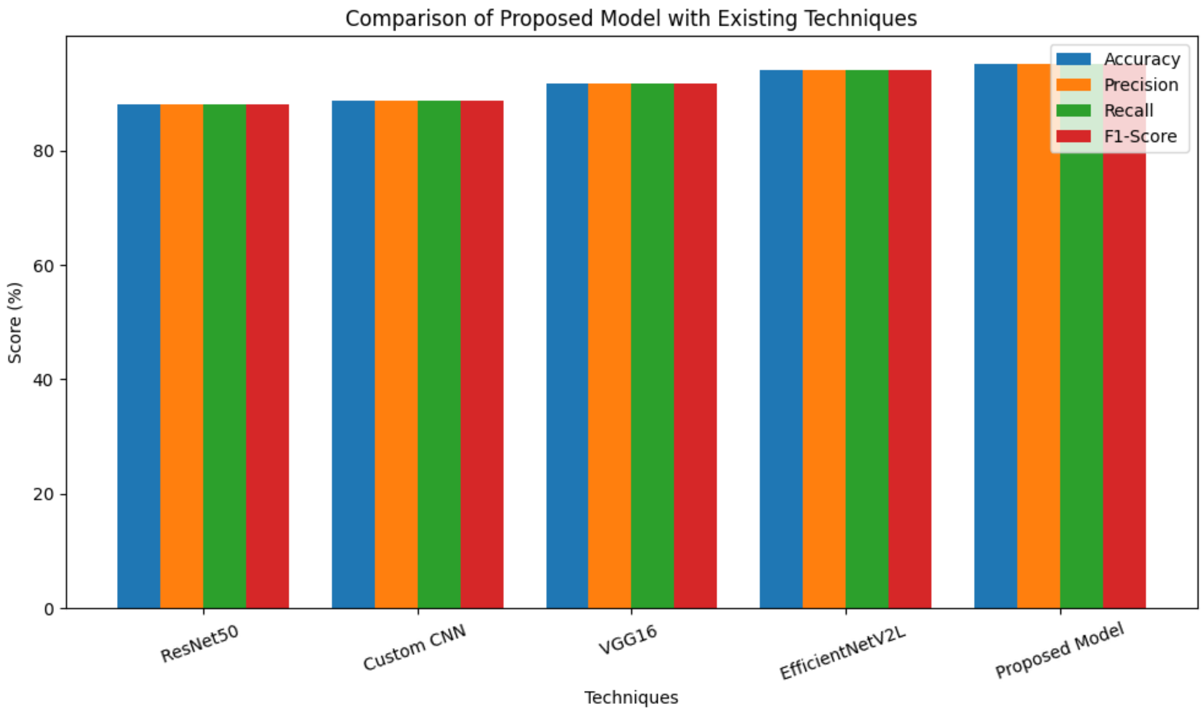

Figure 9: Comparative analysis of proposed model with existing techniques based on performance metrices such as Accuracy, Precision, Recall, and F-1 score.

5. Conclusion and Future Work

This study presented a hybrid framework combining EfficientNetB0, fine-tuned using transfer learning, with an XG Boost classifier for binary classification of chest X-ray images into Normal and Pneumonia classes.. By leveraging pre-trained weights and selective layer freezing, the model effectively extracted discriminative deep features while mitigating overfitting. The XG Boost classifier further enhanced classification by capturing complex non-linear relationships among the extracted features. Experimental results demonstrated the superiority of the proposed approach, achieving an accuracy of 95.20%, precision of 95.20%, recall of 95.20%, and F1-score of 95.20%. Comparative analysis showed that the proposed model outperformed existing techniques such as EfficientNetV2L, Custom CNN, VGG16, and ResNet50, validating its effectiveness for reliable and efficient chest disease diagnosis. For future work, we aim to extend the framework to larger and more diverse multi-center datasets to further assess generalization. Incorporating explainable AI techniques, such as Grad-CAM or SHAP, would enhance interpretability by highlighting regions influencing predictions, thereby increasing clinical trust. Additionally, exploring lightweight vision transformer architectures and ensemble strategies could improve robustness while reducing computational overhead. Deployment of the model as a real-time clinical decision support system, integrated with existing hospital workflows, represents a critical step toward translating this research into practical healthcare impact.

References

[1] J. F. Ludvigsson et al., “The healthcare system in Sweden,” European Journal of Epidemiology, vol. 40, no. 5, pp. 563–579, 2025.
[2] S. D. C. Catapan et al., “A systematic review of consumers’ and healthcare professionals’ trust in digital healthcare,” NPJ Digital Medicine, vol. 8, no. 1, p. 115, 2025.
[3] D. Ma et al., “A fully open AI foundation model applied to chest radiography,” Nature, vol. 643, no. 8071, pp. 488–498, 2025.

[4] A. H. Chehade et al., "Reconstruction-based approach for chest X-ray image segmentation and enhanced multi-label chest disease classification," Artificial Intelligence in Medicine, vol. 165, p. 103135, 2025.
[5] S. K. Satapathy et al., "A federated learning approach for classifying chest diseases from chest X-ray images," Biomedical Signal Processing and Control, vol. 100, p. 107107, 2025.
[6] C. Usman et al., "Pneumonia disease detection using chest X-rays and machine learning," Algorithms, vol. 18, no. 2, p. 82, 2025.
[7] R. Bhuvanya et al., "A hybrid deep learning approach using XceptionNet and Vision Transformer for chest disease detection," Biomedical Signal Processing and Control, vol. 110, p. 108118, 2025.
[8] S. Dardouri, "A deep convolutional neural network model for lung disease detection using chest X-ray imaging," Pulmonary Medicine, vol. 2025, p. 6614016, 2025.
[9] N. R. Subawickrama et al., "Advancements in breathomics and AI for disease diagnosis," ACS Omega, vol. 10, no. 5, pp. 4187–4196, 2025.
[10] M. A. Salam et al., "A hybrid deep learning and machine learning model for lung disease detection," International Journal of Intelligent Engineering & Systems, vol. 18, no. 1, 2025.
[11] H. Sadr et al., "AI in disease diagnosis: A comprehensive review," European Journal of Medical Research, vol. 30, no. 1, p. 418, 2025.
[12] S. Paul et al., "Thorax disease detection using computer vision," in Proc. Int. Conf. Data Science and Applications, 2025, pp. 440–451.
[13] A. Migdady et al., "EfficientNet model for chest X-ray disease detection," International Journal of Business Information Systems, vol. 50, no. 1, pp. 1–21, 2025.
[14] M. A. Al-qaness et al., "Chest X-ray disease detection using deep learning: A survey," Archives of Computational Methods in Engineering, vol. 31, no. 6, pp. 3267–3301, 2024.
[15] S. Singh et al., "Efficient pneumonia detection using vision transformers," Scientific Reports, vol. 14, no. 1, p. 2487, 2024.
[16] V. Ravi et al., "EfficientNet-based ensemble model for lung disease detection," Cluster Computing, vol. 26, no. 2, pp. 1181–1203, 2023.
[17] M. Mann et al., "Deep CNNs for accurate chest X-ray diagnosis," Interdisciplinary Sciences, vol. 15, no. 3, pp. 374–392, 2023.
[18] M. Ali et al., "Pneumonia detection using EfficientNetV2L model," IEEE Access, vol. 12, pp. 34691–34707, 2024.
[19] J. M. Dudala, "Automated pneumonia detection using CNN," in Proc. Int. Conf. Decision Aid Sciences and Applications, 2024, pp. 1–7.
[20] S. Sharma and K. Guleria, "Pneumonia detection using VGG-16 and neural networks," Procedia Computer Science, vol. 218, pp. 357–366, 2023.
[21] N. Sebastian and B. Ankayarkanni, "Enhanced ResNet-50 for pneumonia detection," Diagnostics, vol. 15, no. 16, p. 2041, 2025.