Human-Centric Intelligent Systems
https://doi.org/10.1007/s44230-025-00091-9

RESEARCH ARTICLE

Deep Learning for Pneumonia Detection: A Combined CNN and YOLO Approach

Rathnakannan Kailasam[1] · Saranya Balasubramanian[2]

Received: 23 July 2024 / Accepted: 6 February 2025

Abstract

Pneumonia, a prevalent lung infection caused by various pathogens, remains a leading cause of hospitalization globally, particularly in resource-limited regions where expert radiologists are scarce. Chest X-rays (CXRs) are the primary diagnostic tool for pneumonia; however, their manual interpretation is time-consuming and requires significant expertise. This study investigates the potential of deep learning for automated pneumonia detection and localization, addressing challenges of efficiency and accessibility in clinical diagnostics. A Convolutional Neural Network (CNN) was employed for image classification, and the YOLO algorithm was utilized for region-of-interest (ROI) localization. Four models were trained using diverse CXR datasets preprocessed for consistency, incorporating varying combinations of data augmentation and dropout techniques. Model performance was evaluated based on training accuracy, validation accuracy, and F1-scores. The best-performing model achieved a training accuracy of 0.968, a validation accuracy of 0.83, and F1-scores of 0.799 for normal images and 0.819 for pneumonia images. Additionally, the YOLO-based localization approach achieved F1-scores of 0.82 for normal images and 0.54 for pneumonia images, with a weighted average of 0.71 and a macro average of 0.68. This study demonstrates the feasibility of machine learning models for automated pneumonia detection and localization in CXRs, providing a cost-effective and efficient alternative to traditional diagnostic methods. The proposed models significantly reduce diagnostic time while maintaining high accuracy, offering a transformative solution for healthcare systems, particularly in under-resourced settings. These advancements have the potential to alleviate the burden on radiologists, improve patient outcomes, and enhance access to quality healthcare worldwide.

Keywords Pneumonia · X-ray · Diagnosis · Deep learning · CNN · Localisation · YOLO · Automated detection

1 Introduction

Pneumonia, a lung infection affecting individuals of all ages, can have life-threatening consequences if left undiagnosed. Early and accurate diagnosis is crucial, traditionally relying on the interpretation of chest X-rays (CXRs) by radiologists. This process, historically, has been a time-consuming and challenging task for human experts. While researchers have proposed various computer algorithms and tools to aid in decision-making, their effectiveness has been limited. However, the field has witnessed a significant paradigm shift with the emergence of both handcrafted and deep learning techniques, empowering researchers to analyze and classify medical images with remarkable accuracy. Convolutional Neural Networks (CNNs), renowned for their exceptional performance in image-related tasks, have proven to be highly effective in detecting pneumonia from CXRs.

The study by Salido and Ruiz [1] explored the use of color-to-grayscale conversion and CNNs for feature extraction and classification, aiming to improve the accuracy of pneumonia detection. YOLO, a powerful subclass of deep neural networks, has demonstrated significant success in vision-based computations [2, 3], including object detection. However, pre-trained YOLO models, typically built on massive datasets, can be large and resource-intensive, requiring substantial computing power. This can lead to prolonged training times and increased susceptibility to overfitting, particularly when dealing with smaller datasets [4, 5]. To mitigate these challenges, this study constructs the YOLO

✉ Rathnakannan Kailasam
krkannan@annauniv.edu

[1] Department of Electrical and Electronics Engineering, CEG, Anna University, Chennai, India

[2] Toronto, Canada

Published online: 08 March 2025

architecture [2] from the ground up, avoiding the limitations associated with pre-trained models. Datasets from RSNA and Kaggle [6] were utilized to evaluate the performance of the proposed models. Data augmentation, a crucial technique for enhancing the accuracy of image classification with limited data, was implemented. Furthermore, another study [7] introduced a novel data augmentation method that leverages Generative Adversarial Networks (GANs) to generate synthetic chest X-ray images for further analysis.

Manual detection of threat objects in X-ray images is a laborious and time-consuming process, especially when dealing with obscured or rotated objects [8–10]. This study proposes a YOLO-based object detector to automate threat detection in X-ray images, thereby addressing this significant challenge. Several prior studies have contributed valuable insights to this domain. Srivastava et al. [11] introduced the "dropout" technique to prevent overfitting during model training, a strategy strategically employed in this project. Ozturk et al. [12] developed a deep learning model for accurate classification of chest X-rays, achieving impressive accuracy in both binary (COVID-19 vs. No Findings) and multi-class (COVID-19 vs. No Findings vs. Pneumonia) classifications. Notably, their model incorporates the DarkNet framework within the YOLO system.

Yao et al. [13] proposed Pneumonia Yolo (PYolo), an efficient algorithm for pneumonia detection based on a CNN specifically designed for X-ray image data. PYolo utilizes dilated convolutions and an attention mechanism to enhance the detection of pneumonia lesions, achieving a mean Average Precision (mAP) of 46.84 on the RSNA dataset. Nithya et al. [14] proposed an efficient multisampling image filtering technique in conjunction with an enhanced CNN architecture for pneumonia classification. These studies collectively demonstrate the substantial potential of machine learning, particularly CNNs and YOLO, in revolutionizing pneumonia detection from chest X-rays. By automating the detection process and improving accuracy, these techniques can significantly assist medical professionals in timely diagnosis and effective treatment, ultimately saving lives. The core idea presented in the proposed paper is to develop an efficient and low-cost machine-learning model for automatic pneumonia identification in lung X-rays. Achieving this objective holds the potential to revolutionize healthcare by enabling faster, more accurate diagnoses and alleviating the workload on radiologists.

The proposed model aims to address two key challenges associated with traditional methods of pneumonia detection:

(i) Efficiency: Manual analysis of X-rays by radiologists is a time-consuming and resource-intensive process. Automating this task with a well-trained model could significantly reduce diagnosis times and improve patient outcomes.

(ii) Cost-effectiveness: Implementing advanced diagnostic tools can be expensive, particularly for resource-constrained healthcare settings. Developing a low-cost model would make this technology more accessible to a wider range of patients and healthcare providers.

By achieving both efficiency and cost-effectiveness, this research has the potential to make a significant positive impact on global healthcare, particularly in regions with limited access to advanced medical resources.

2 Proposed System

The proposed system's workflow for pneumonia prediction is illustrated in Fig. 1.

The initial stage of our pneumonia detection pipeline involves pre-processing the collected chest X-ray images.

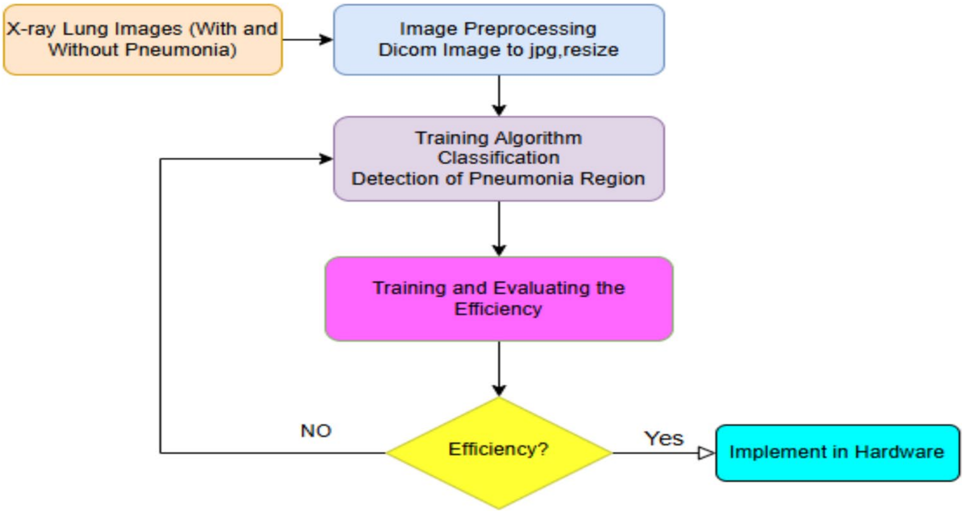

X-ray Lung Images (With and Without Pneumonia)
Image Preprocessing
Dicom Image to jpg,resize
Training Algorithm
Classification
Detection of Pneumonia Region
Training and Evaluating the Efficiency
NO
Efficiency?
Yes
Implement in Hardware

Fig. 1 Block diagram of the proposed system

Given that medical images are typically stored in DICOM format [15], we begin by converting them to the more widely used JPG format, ensuring compatibility with our chosen deep learning algorithms. To standardize input dimensions, all images are resized to 200 × 200 × 3 pixels while maintaining three color channels. Furthermore, we convert the images from RGB to grayscale, reducing dimensionality from three channels to one. This transformation enhances training efficiency and processing speed, as extracting features from image intensity variations proves sufficient for accurate pneumonia detection in our application.

The pre-processed images are subsequently fed into the deep learning model. Within the model, a series of layers collaboratively extract relevant features indicative of pneumonia. These extracted features are then flattened into a one-dimensional vector and passed to a fully connected layer. This final layer utilizes the extracted features to make predictions on new, unseen chest X-ray images. Continuous monitoring and evaluation of the model's performance ensure optimal accuracy and efficiency in pneumonia detection.

2.1 Tools Used and Algorithm for Pneumonia Detection

This research leverages potent cloud resources (Google Colab with GPUs/TPUs) and key software tools (Python, OpenCV, Keras/PyTorch) to analyze chest X-rays and detect pneumonia. The model employs two CNN architectures (with and without dropout) built using ReLU and Sigmoid activation functions [16, 17]. These models were trained on datasets with and without augmentation to effectively capture spatial and temporal relationships within the images for accurate pneumonia diagnosis.

The proposed analysis incorporates two distinct CNN architectures tailored for pneumonia detection. The CNN with Dropout utilizes strategically placed dropout layers after convolutional layers to prevent overfitting and improve generalization. This architecture comprises 23 layers, including 3 convolutional layers with ReLU activation (except for the final Sigmoid layer for binary classification). Each convolutional layer is paired with a pooling and dropout layer. The network processes 200 × 200 × 3 color images using 16 filters with a 3 × 3 kernel size.

Extracted features from the first four convolutional layers, with progressively doubled channels, are flattened into a 1D vector by a flatten layer. This data is then fed into three dense layers (512, 256, and 64 neurons) responsible for image classification. The training process is guided by binary cross-entropy loss, the Adam optimizer, and accuracy metrics [17, 18]. Figure 2 visualizes the network architecture.

The baseline CNN, without dropout as seen in Fig. 3 for comparison, comprises 14 layers. It includes 3 convolutional-pooling pairs for feature extraction, with ReLU activation applied to all but the final layer, which employs Sigmoid for binary classification. The network accepts 200 × 200 × 3 color images as input. Convolutional layers utilize 16 filters with a kernel size of 3 × 3, chosen for optimal performance.

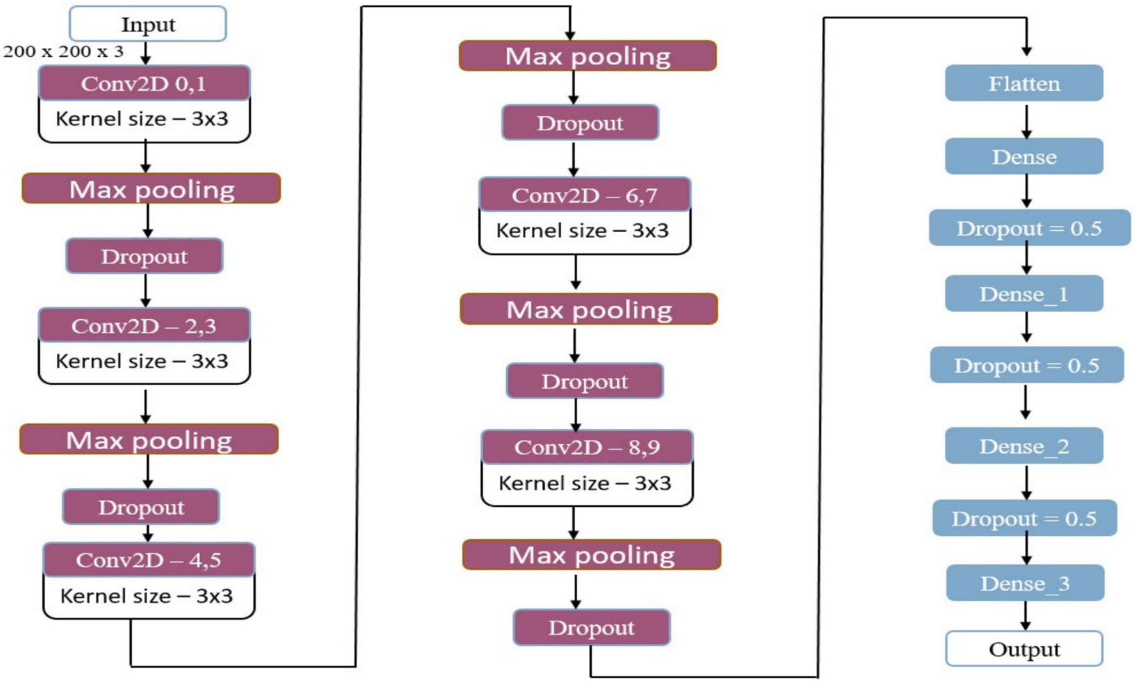


Fig. 2 Architecture of the CNN using dropout

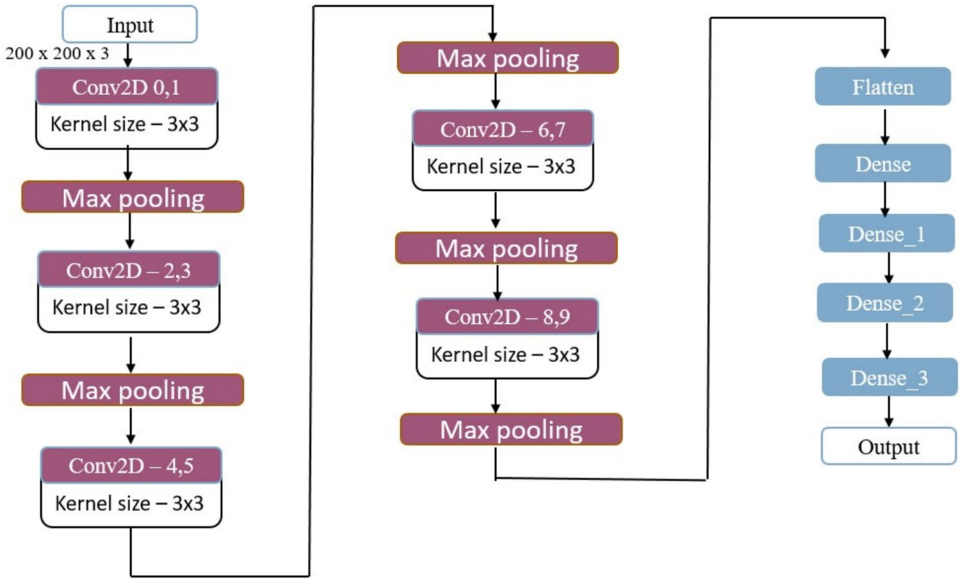


Fig. 3 Architecture of the CNN without dropout

In each of the four subsequent convolution layers, the number of input features is doubled while maintaining a constant kernel size. The remaining parameters of the model remain unchanged.

2.2 YOLO

YOLO has emerged as a cutting-edge solution for real-time object detection, marking a significant advancement in the field. Its unique combination of speed, accuracy, and innovative features, such as adaptive anchor boxes, makes it a powerful tool for diverse real-world applications. Unlike traditional two-stage methods, YOLO processes the entire image in a single pass, achieving a remarkable balance of speed and accuracy.

The YOLO Backbone (CSPDarknet53) network employs a novel approach where layers are divided into processing and bypass branches. This strategy effectively preserves crucial features while simultaneously reducing the number of network parameters. Both the SPP component and PANet play pivotal roles: SPP effectively captures objects of varying scales, while PANet facilitates the flow of high-resolution information to upper layers, enabling precise object localization. Finally, the Head is responsible for generating the final predictions, outputting bounding box coordinates, confidence scores, and class probabilities for each detected object.

2.3 Chest X-Ray for Pneumonia and Data Set

X-rays are a relatively quick and non-invasive imaging modality, allowing for prompt evaluation in patients with suspected pneumonia [19]. The images provide a visual representation of the lungs and can help identify signs of infection, inflammation, and consolidation. In patients with pneumonia, chest X-ray helps determine the extent and location of the infection, which allows healthcare professionals to determine the choice of appropriate treatment. It also provides a baseline for monitoring and assessing the patient's progress during and after treatment. Regular follow-up X-rays may be performed to assess the resolution of the infection and the healing of lung tissue. The dataset used for this study comprises 6289 chest X-ray (CXR) images, carefully curated from diverse and reputable sources, including Google Images, Kaggle repositories, and the RSNA Pneumonia Detection Challenge dataset. This extensive and heterogeneous collection is recognized for its high quality, diversity, and relevance, representing a wide range of demographic groups and pneumonia cases. By leveraging such a dataset, the study aims to ensure robustness and inclusivity in training and evaluating the machine learning models.

To facilitate effective learning and unbiased assessment, the dataset was systematically divided into three subsets: training, validation, and testing. This segmentation ensures a balanced approach to model development and evaluation, enabling the network to learn from a substantial amount of data while also being rigorously tested on unseen cases to assess its generalization capabilities. The distribution of normal and pneumonia-affected images in each subset is detailed in Table 1.

This division provides a robust foundation for training and evaluating the proposed models. The training set, comprising 3875 normal and 1341 pneumonia images, serves as the primary resource for the model to learn distinguishing features. The validation set, with 390 normal and 234 pneumonia images, is used to fine-tune the hyperparameters and prevent overfitting. Finally, the testing set, which includes

Table 1 Dataset information

Dataset	Number of normal images	Number of pneumonia images
Training data set	3875	1341
Validation data set	390	234
Testing data set	170	279

170 normal and 279 pneumonia images, ensures a comprehensive and unbiased evaluation of the model's performance on unseen data.

Figure 4 illustrates a representative chest X-ray without pneumonia, while Fig. 5 showcases X-rays with pneumonia manifesting in various positions. This visual diversity further enriches the dataset, helping the model to generalize

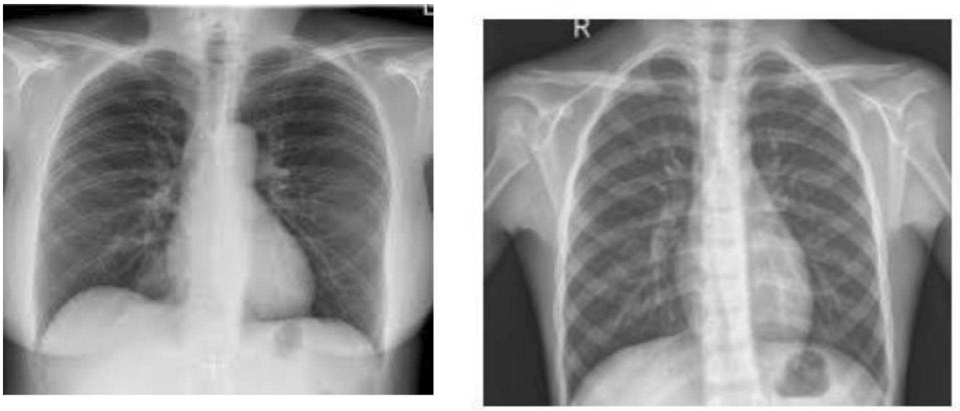
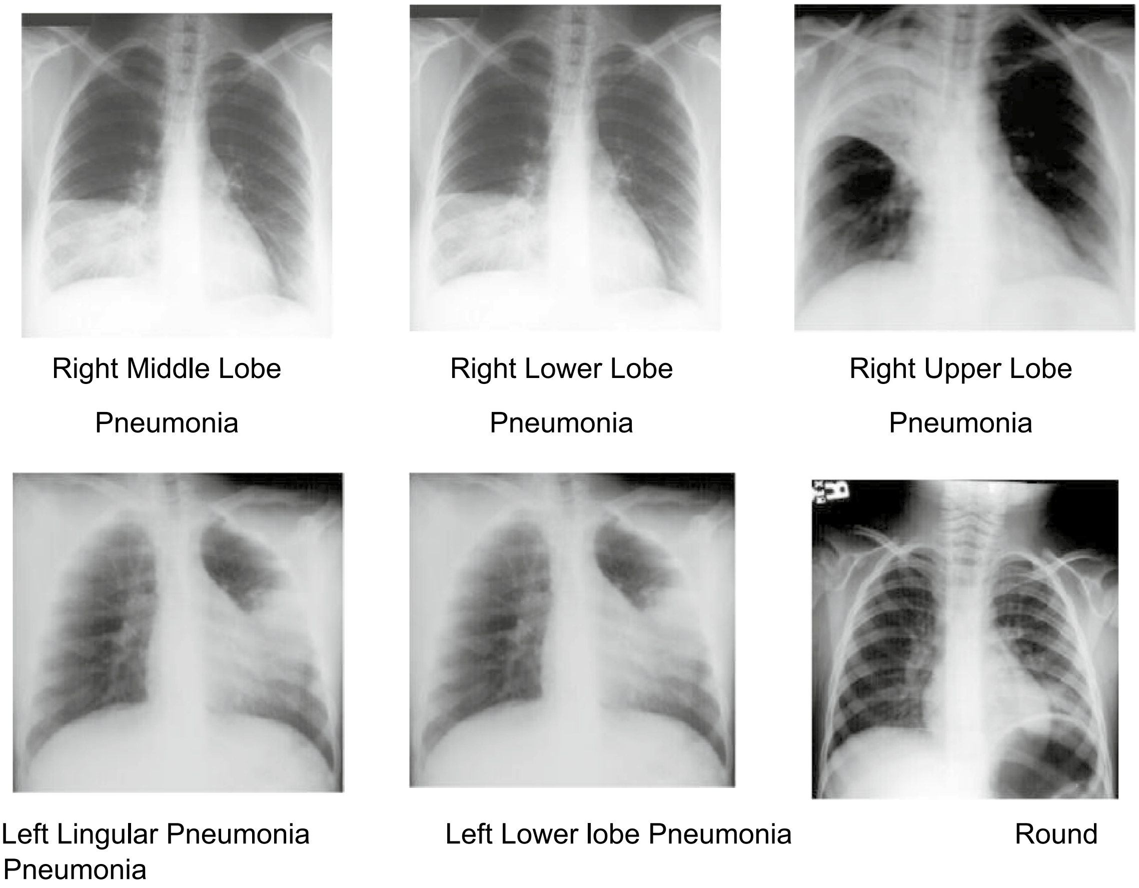


Fig. 4 Chest X-ray without pneumonia


Right Middle Lobe Pneumonia

Right Lower Lobe Pneumonia

Right Upper Lobe Pneumonia

Left Lingular Pneumonia Pneumonia

Left Lower lobe Pneumonia

Round

Fig. 5 Chest X-ray images affected with pneumonia

better across a variety of real-world clinical scenarios. The inclusion of such a representative dataset underpins the study's goal of developing a reliable, efficient, and scalable solution for automated pneumonia detection and localization in clinical practice.

Initially, the collected pneumonia images were in DICOM format, a common format for medical images. To make them suitable for neural network algorithms, we converted them to the JPG format. Additionally, as images were sourced from diverse origins, they varied in size. To standardize the input, we resized all images to a uniform 200 × 200 × 3 pixel format. For feature extraction, we converted the images to grayscale (single-channel) format, which can improve training efficiency and processing time.

During the training phase, the model employs multiple layers to extract salient features from the input images. Once these features are extracted, they are flattened and fed into a fully connected layer. The model's performance is evaluated by making predictions on the testing dataset and monitoring its accuracy. Our models were trained using the Adam optimizer with a learning rate of 0.001. The training process was conducted for 80 epochs, with a batch size of 32. We employed standard data augmentation techniques such as random rotations, flips, and zooming to enhance the robustness of our models. To prevent overfitting, we utilized early stopping and a dropout layer with a probability of 0.5.

3 Analysis and Results of Pneumonia Detection Using CNN

A total of 80 epochs were used for each model with a batch size of 64.

The four models investigated in this work are:

Model 1: Without Augmented dataset, with dropout,
Model 2: With Augmented dataset, with dropout,
Model 3: With Augmented dataset, without dropout,
Model 4: With Augmented dataset, without dropout.

Model 1 Results: Model 1 utilizes traditional Convolutional Neural Network (CNN) layers, including convolution, pooling, and dropout, to extract salient features from the input data. These extracted features, in the form of flattened feature maps, are then fed into a fully connected layer for classification. Before feeding the data into the model, it undergoes preprocessing steps to ensure compatibility with the model's requirements, such as resizing and normalization.

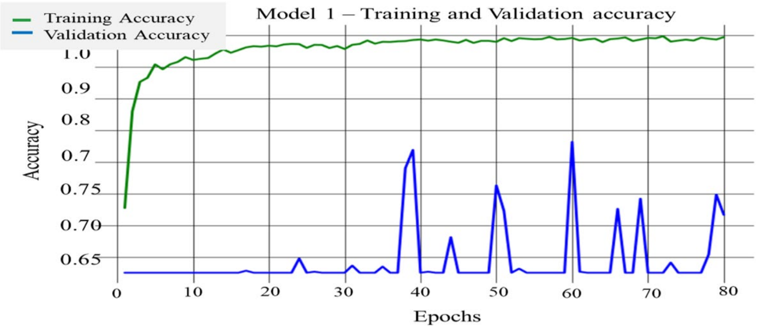
Figure 6 visually depicts the model's training and validation accuracy over epochs, providing insights into its learning progress and generalization performance. Training accuracy remains stable between 0.96 and 0.987 after 30 epochs, while validation accuracy peaks around epoch 15 at approximately 0.8. This suggests that the best model in this set can be obtained after 60 epochs. Utilizing a callback function allows for early stopping at this point, rather than continuing until epoch 100.

Training loss, typically calculated with binary cross-entropy in classification problems due to its effectiveness, is represented by the training dataset. As shown in Fig. 7, the training loss begins decreasing from the first epoch itself. While further increasing epochs might lead to a continued decrease in loss, it also carries the risk of underfitting or overfitting the model. Therefore, stopping training at the best-performing epoch is crucial.

Figure 8 illustrates the loss incurred during validation of the data on the validation set. Typically, this validation loss is lower than the training set loss. This phenomenon arises because the model, during the training phase, learns by encountering novel challenges and undergoes adjustments through backpropagation. Conversely, during validation, the model encounters validation data that it has already learned to handle effectively. At the 60th epoch, a sharp spike is observed in the validation loss, which subsequently tends to fluctuate between 0.2 and 0.6.

Model 2 employs the same underlying architecture as Model 1, comprising traditional convolutional neural network (CNN) layers including convolution, pooling, and dropout. However, a key distinction lies in the strategic implementation of data augmentation techniques within

Fig. 6 Accuracy of Model 1



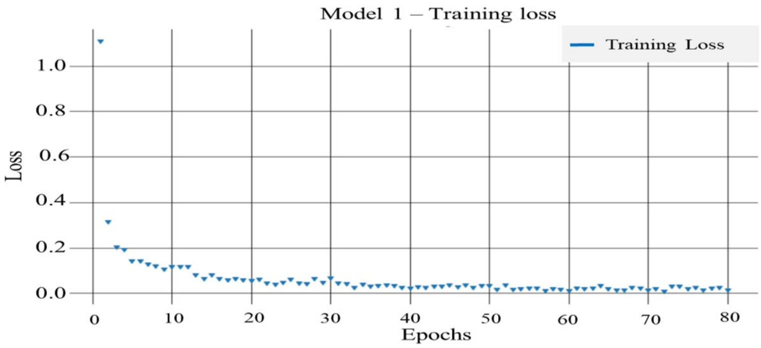


Fig. 7 Training loss of Model 1

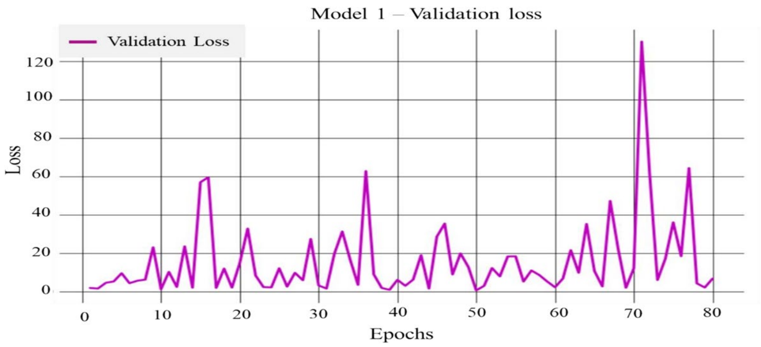


Fig. 8 Validation loss of Model 1

Table 2 Data augmentation specification

Data augmentation techniques	Values
Rescale	1.0/255
Rotation range	10
Zoom range	0.1
Width shift range	0.1
Height shift range	0.1
Horizontal flip	True

Model 2. By applying a diverse range of data augmentation methods, such as random rotations, flips, and scaling, Model 2 effectively generates new training samples from the existing dataset. This expanded and diversified training dataset empowers the model to learn more robust and generalizable features, ultimately enhancing its capacity to accurately classify unseen data. Detailed model specifications are provided in Table 2.

Figure 9 depicts the training and validation accuracy for this model. Training accuracy demonstrates a steady increase from 0.85 to 0.98 within the first 20 epochs and subsequently stabilizes between 0.94 and 0.98. Notably, validation accuracy exhibits a significant improvement compared to Model 1, converging towards training accuracy after 60 epochs (from 0.85 to 0.9). Ideally, the discrepancy between training and validation accuracy should be neither excessively large nor excessively small. Based on this observation, Model 2 appears to strike a favorable balance in terms of accuracy.

As shown in Fig. 10, the training loss commences at 0.35 and remains within the range of 0.15 to 0.5 after 20 epochs. While the losses exhibit some fluctuations, the values remain acceptable and do not indicate overfitting or underfitting.

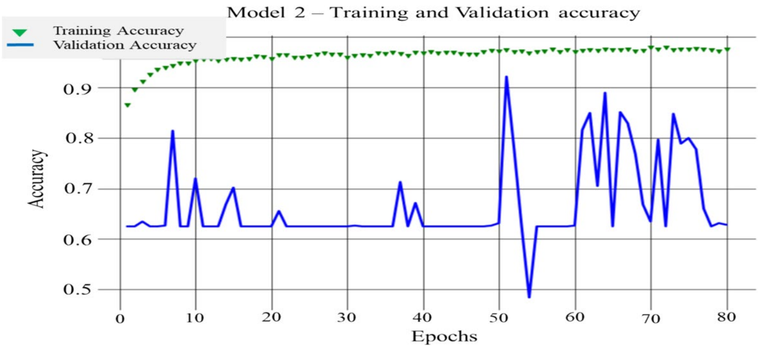


Fig. 9 Training and validation accuracy of Model 2

The validation loss, depicted in Fig. 11, primarily stabilizes around 0.4, with only one outlier epoch reaching 0.40. A comparison of Models 1, 2, and 3 in terms of validation loss reveals similar behavior.

Model 3 Results: Model 3 is a variant of Model 1, distinguished by the absence of dropout layers. While the core architecture, encompassing convolutional, pooling, flattening, and fully connected layers, remains unchanged, the exclusion of dropout layers facilitates a more streamlined training process. This modification can potentially result in faster convergence and potentially enhanced performance, contingent upon the specific dataset and model hyperparameters.

Figure 12 visually depicts the validation accuracy attained by Model 3, offering insights into its generalization capabilities. It demonstrates a steady increase until approximately epoch 30, followed by stabilization at approximately 0.75. Notably, the training accuracy rapidly approaches 1 after only 10 epochs, indicating potential overfitting and suggesting that the model may have exhaustively learned the training data. This translates to a testing accuracy biased towards a single classification, thereby limiting its generalizability.

As shown in Fig. 13, the training loss begins at 0.12 and exhibits a rapid exponential decrease, ultimately reaching zero around epoch 30. This rapid decline suggests that the model has become fully learned, closely fitting the training data. However, this rapid decrease may also indicate overfitting, meaning the model might not generalize well to unseen data. The validation loss, as depicted in Fig. 14, for this

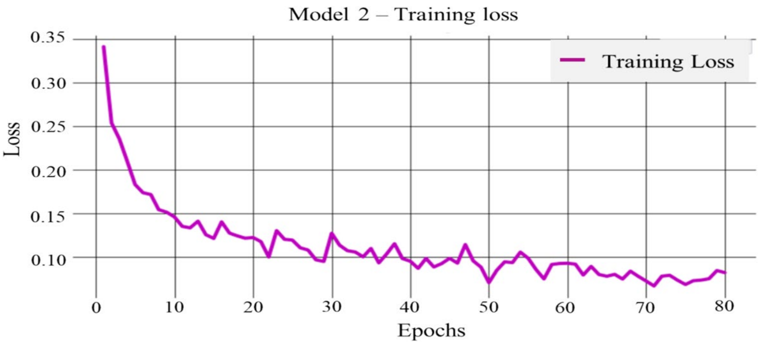


Fig. 10 Training loss of Model 2

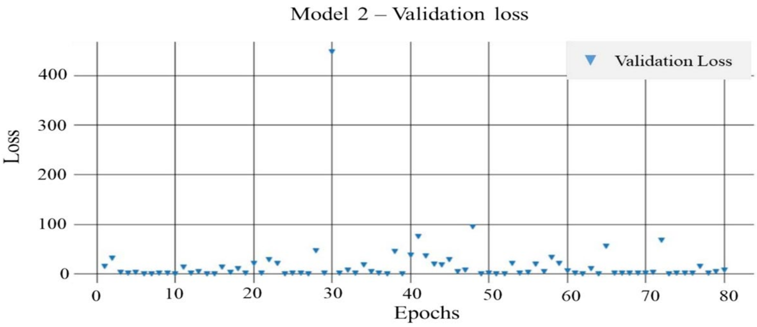


Fig. 11 Validation loss of Model 2

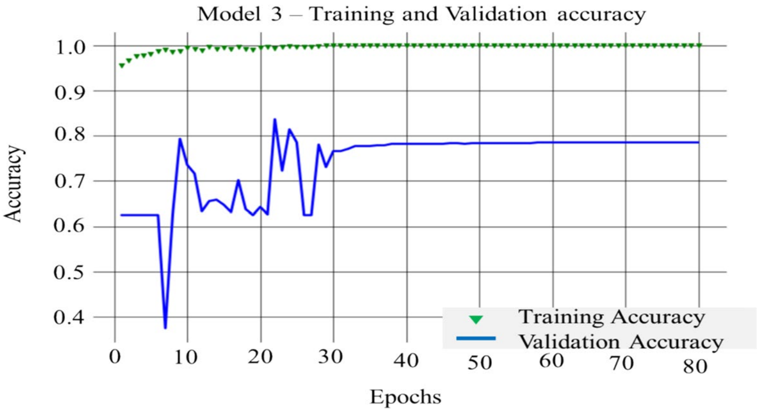


Fig. 12 Training and validation accuracy of Model 3

model displays a wide range of values (0–12.5) until epoch 30, before stabilizing around 3. As mentioned earlier, this behavior is characteristic of an overfitted model, suggesting that it has learned the training data too closely and may not perform well on unseen data. Therefore, due to its overfitting tendencies, this model cannot be considered a viable option.

Model 4 Results: Model 4 maintains the same layer architecture as Model 3, which omits dropout layers to potentially accelerate training and enhance performance. To further augment the model's ability to generalize to unseen data, Model 4 incorporates data augmentation techniques, similar to Model 2. By artificially generating diverse training samples through transformations such as rotations, flips,

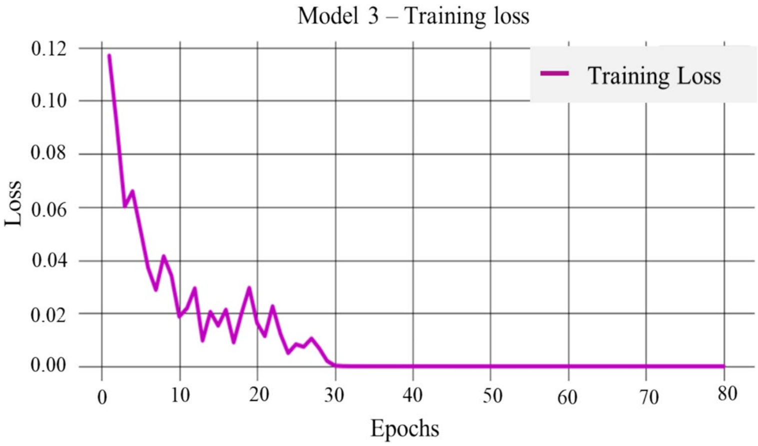


Fig. 13 Training loss of Model 3

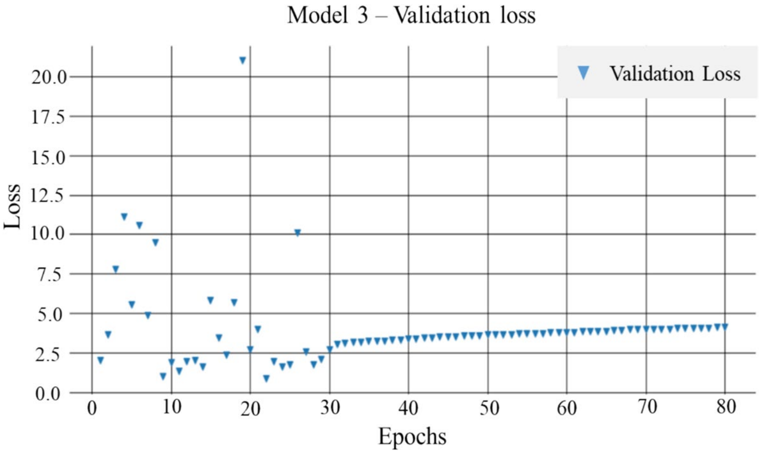


Fig. 14 Validation loss of Model 3

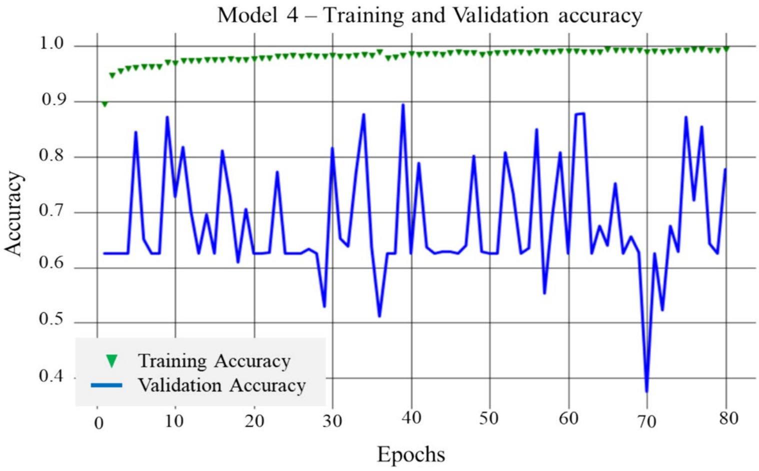


Fig. 15 Training and validation accuracy of Model 4

Fig. 16 Training loss of Model 4

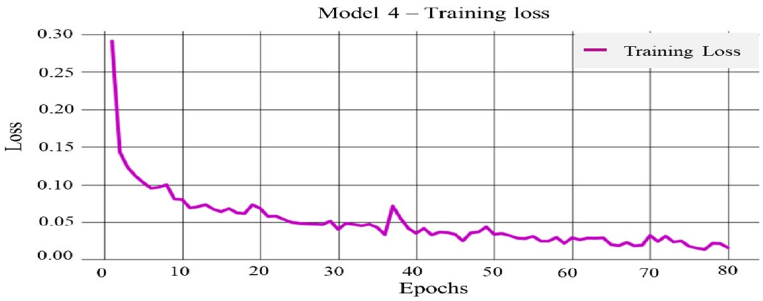


Fig. 17 Validation loss of Model 4

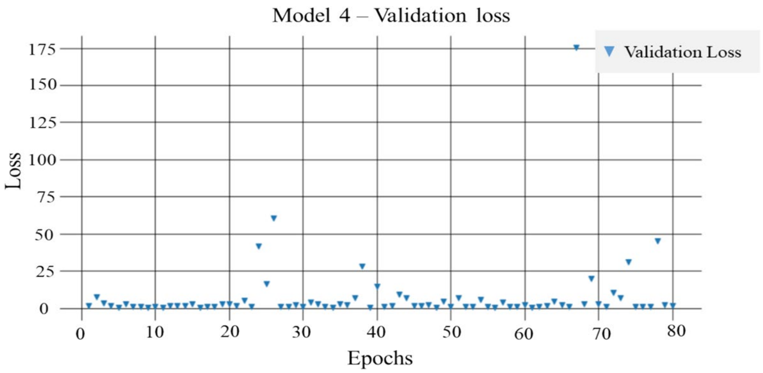


Table 3 Accuracy of training and validation dataset

Models	Training accuracy	Validation accuracy
1—with dropout without augmentation	0.97	0.63
2—with dropout with augmentation	0.968	0.83
3—without dropout without augmentation	1.00	0.78
4—without dropout With augmentation	0.98	0.65

Table 4 Precision, recall and F1-score of test dataset

Metrics		Model 1	Model 2	Model 3	Model 4
Precision	0	1.00	0.78	0.45	0.50
	1	0.67	0.83	0.68	0.74
Recall	0	0.50	0.80	0.30	0.53
	1	1.00	0.81	0.50	0.57
F1-score	0	0.67	0.799	0.36	0.51
	1	0.80	0.819	0.58	0.64

and scaling, Model 4 aims to improve its robustness and accuracy. This strategy enables the model to learn more robust and generalizable features, ultimately enhancing its classification performance (details in Table 2). Figure 15 depicts the training and validation accuracy for this model. While the training accuracy rapidly approaches 1, indicating potential overfitting, the validation accuracy remains within a favorable range of 0.6 to 0.9. Although performing better than Model 3, it isn't necessarily the optimal model due to the persistent high training accuracy, suggesting potential overfitting tendencies.

Figure 16 illustrates the training loss, exhibiting a pronounced exponential decrease from 0.30 to 0.07 within just 10 epochs, followed by a gradual decline towards 0.03. This rapid initial decrease further accentuates the possibility of overfitting. Figure 17 portrays the validation loss, which remains remarkably close to zero throughout the training process. While this might seem desirable, it corroborates the concerns raised by the training accuracy and loss trends, suggesting potential overfitting and limited generalizability.

The best validation and training accuracies from all the 4 models are listed in the Table 3. While all four models demonstrated comparable results, Model 2 (augmented dataset with dropout CNN) emerged as slightly more accurate and computationally efficient. This choice is supported by the following key observations.

Improved Performance: Data augmentation in Model 2 effectively prevented overfitting, a limitation observed in Model 3. Furthermore, compared to Model 1, Model 2 demonstrated significantly better accuracy.

3.1 Loss–Accuracy Correlation

Analyzing the accuracy and loss trends in Model 2 revealed a desirable correlation. As accuracy increased, validation loss decreased, indicating that the model was genuinely learning and not merely memorizing the training data.

Strong Testing Performance: Predictions made on the testing dataset after training further solidified the case for Model 2. As detailed in Table 4, it achieved favorable precision, recall, and F1-score values compared to the other models.

In evaluating classification models, precision assesses the proportion of predicted positives that are truly positive, while recall captures the model's ability to identify all actual positives. The F1-score, as a weighted average of these two metrics, provides a balanced measure that accounts for both false positives and false negatives, offering a deeper understanding of the model's performance across both precision and recall. This is crucial for tasks like pneumonia detection, where minimizing both types of errors is important. Therefore, analyzing the F1-score of all the models allows us to conclude that Model 2 outperforms the other three models.

4 Analysis of Pneumonia Detection Using CNN and YOLO Algorithms

This section delves into the training process and results obtained using both the convolutional neural network (CNN) with dropout and augmented dataset and the YOLO algorithm. Both algorithms were initially trained on the same dataset comprising 4295 normal and 1652 pneumonia-affected chest X-rays using the proposed model. A total of 80 and 90 epochs were used for CNN and YOLO, respectively. For the YOLO algorithm, the same training dataset used for CNN was analyzed. Additionally, this model aimed to predict and localize the pneumonia-affected regions beyond just classification. To enhance overall efficiency, image formatting was performed prior to analysis.

4.1 Quantitative Results

Training and validation results interpretation: In YOLO, we perform both classification and localization of pneumonia-affected regions. During the training and validation process, we monitor precision, recall, and mAP to track improvements in the model's performance. Precision, as shown in Eq. (1), quantifies the proportion of positive class predictions that are actually true positives. To gain insights into the loss function, we monitor the box loss, class loss, and object loss.

$$\text{Precision} = \left(\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}\right) \tag{1}$$

Figure 18 shows the gradual increase in precision for the training dataset as the number of epochs increases. A total of 90 epochs were used for the analysis. Notably, the final precision score reaches 25%.

Recall, as presented in Eq. (2), measures the proportion of true positive predictions among all actual positive examples in the dataset. Figure 19 depicts the recall metrics for both the training and validation datasets throughout the training process. It reveals a gradual change in the values over time.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

Mean Average Precision (mAP) is a performance metric for object detection models. It is calculated by averaging the Average Precision (AP) across all object classes and, optionally, across different Intersection-over-Union (IoU) thresholds. AP itself measures the balance between precision and recall for a given class at a specific IoU threshold, representing the area under the Precision-Recall curve. By averaging AP across classes and/or IoU thresholds, mAP provides a robust and comprehensive evaluation of the

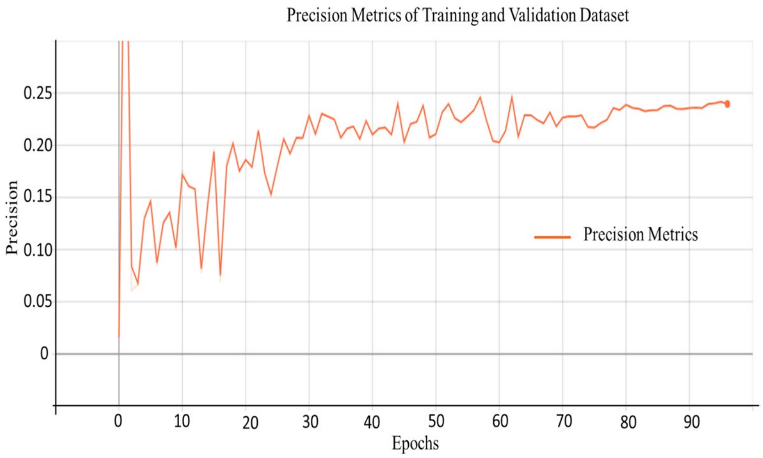


Fig. 18 Precision metrics of training and validation dataset

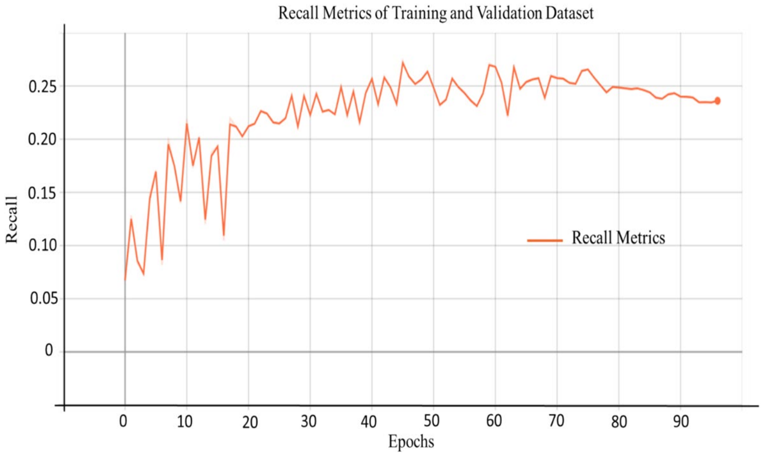


Fig. 19 Recall metrics of training and validation dataset

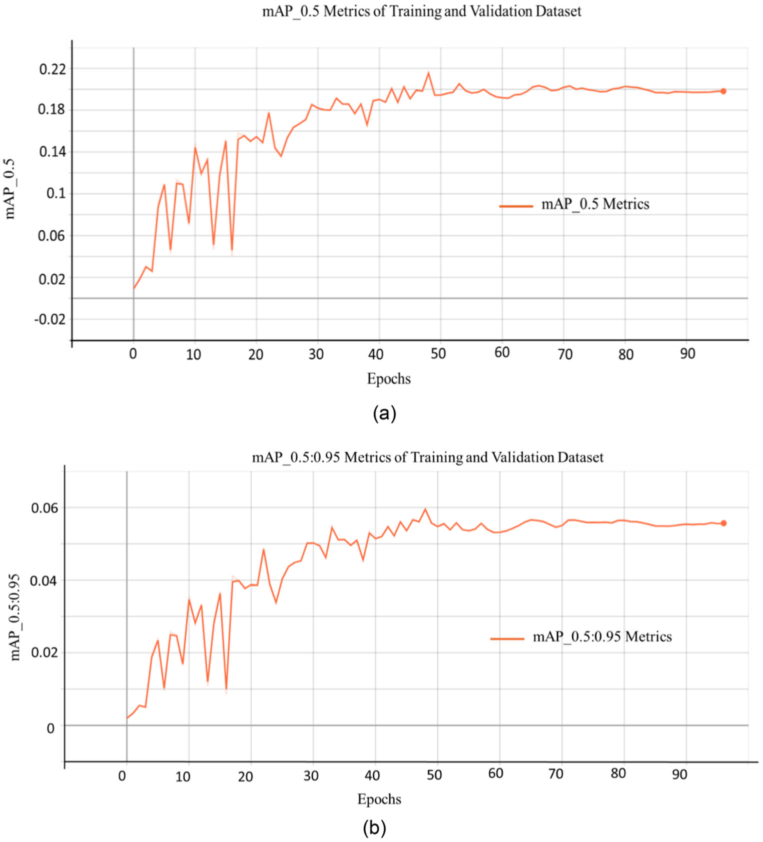


Fig. 20 **a** mAP_0.5 metrics of training and validation dataset. **b** mAP_0.5P0.95 metrics of training and validation dataset

model's detection capabilities. mAP 0.5:0.95 denotes the average mean average precision (mAP) calculated across different Intersection over Union (IoU) thresholds ranging from 0.5 to 0.95 in increments of 0.05 (0.5, 0.55, 0.6, etc.). Figures 20 visualize the evolution of mAP during the training process, demonstrating a gradual increase in these values.

During the training process, the network undergoes self-evaluation to ensure it is learning effectively. This is achieved by utilizing a separate validation dataset. Based on the results obtained from this validation set, the network employs backpropagation, a learning algorithm, to adjust its internal parameters and learn from its mistakes. Throughout this process, the network experiences varying levels of loss. Initially, in the first epoch, the loss is typically high. However, as the network progresses through subsequent epochs, processes more images, and learns from its errors, the loss gradually decreases. Figure 21 visualizes the loss specifically related to the detection of the region of interest (ROI) within the images. As mentioned earlier, the initial loss value is high and subsequently diminishes as the

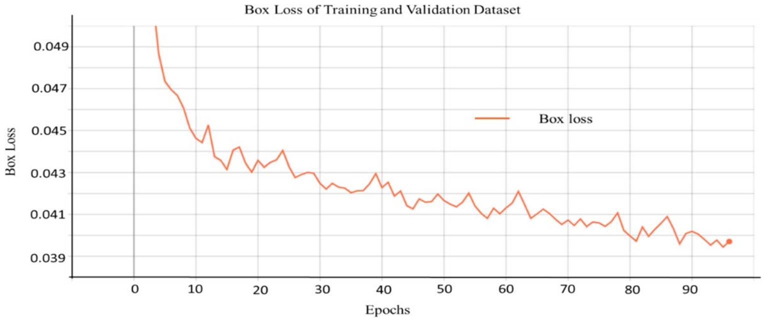


Fig. 21 Box loss of training and validation dataset

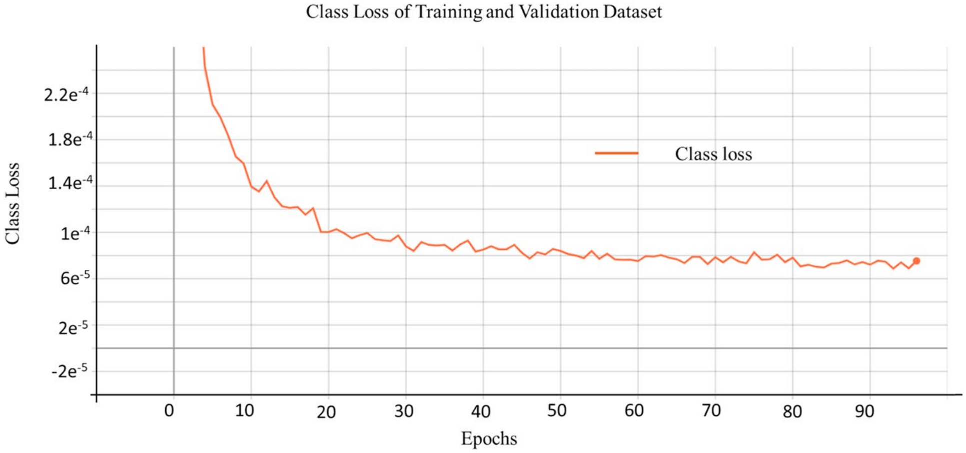


Fig. 22 Class loss of training and validation dataset

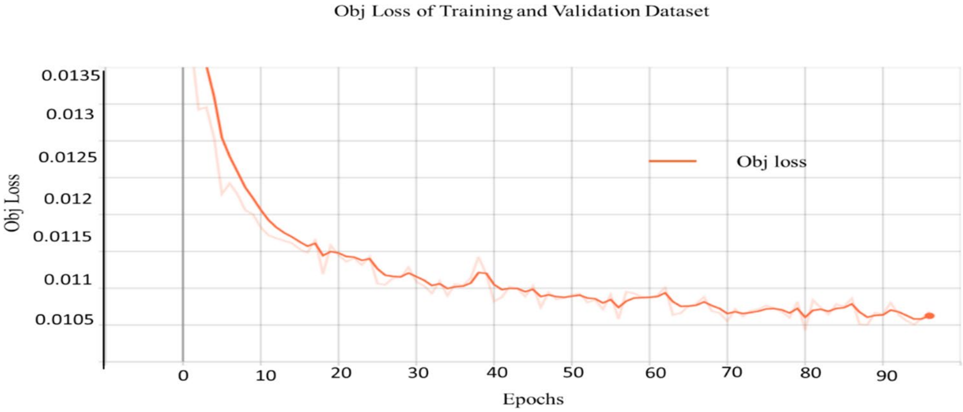


Fig. 23 Obj loss of training and validation dataset

network learns over multiple epochs. Figure 22 follows the same principle, but instead depicts the loss associated with classifying the images as pneumonia or normal.

Figure 23 depicts the object loss incurred during the training and validation processes. It helps us understand the degree of information loss that occurs during the up-sampling and down-sampling of images within the network.

We analyzed the results from four different CNN models to determine the optimal choice. Based on the performance and the advantages observed, we selected Model 2, which utilized an augmented dataset with dropout, as it offered a slight edge in both accuracy and time efficiency. Here are the key factors that led us to choose Model 2 as the best among the four options: Resistance to Overfitting: Data augmentation effectively mitigated overfitting concerns, leading to improved performance. Enhanced Accuracy: Model 2 achieved the highest accuracy compared to the other models. Balanced Learning Dynamics: When analyzing the accuracy and loss curves, we observed a clear decrease in Model 2's validation loss as accuracy increased. This indicates that the model is learning effectively, rather than simply memorizing the training data.

Table 5 CNN accuracy of training and validation dataset

Model	Training accuracy	Validation accuracy
2—with dropout with augmentation	0.968	0.83

The best validation accuracy and training accuracy of the best CNN model are listed in the Table 5.

Figure 24 illustrates the training accuracy of Model 2, which exhibits a smooth increase from 0.85 to 0.98 between epochs 0 and 20. Subsequently, it plateaus and maintains a stable range between 0.94 and 0.98. Notably, the validation accuracy surpasses that of Model 1 and approaches the training accuracy after 60 epochs, reaching a range of 0.85 to 0.9. While a smaller gap between training and validation accuracy is generally preferable for optimal model performance, Model 2 demonstrates a well-balanced trade-off, suggesting strong generalization ability.

Figure 25 depicts the training loss for Model 2. Initially, it starts at 0.35 and gradually converges to a range of 0.15–0.5 after 20 epochs. While the loss exhibits some fluctuations, the values remain within an acceptable range and do not indicate overfitting or underfitting. Figure 26 shows the validation loss for Model 2. It primarily stabilizes around 0.4, with a single outlier epoch where it spikes to 0.400. When compared to Models 1 and 2, the loss behavior appears quite similar in this regard.

4.2 Prediction Result

Following the training of the CNN model using both validation and training datasets, predictions were made on a separate testing dataset. Notably, Model 2 demonstrated the most promising performance among the evaluated models.

Table 6 presents the precision, recall, and F1-score metrics achieved by Model 2 for both normal and pneumonia-affected chest X-rays (CXRs) in the testing dataset.

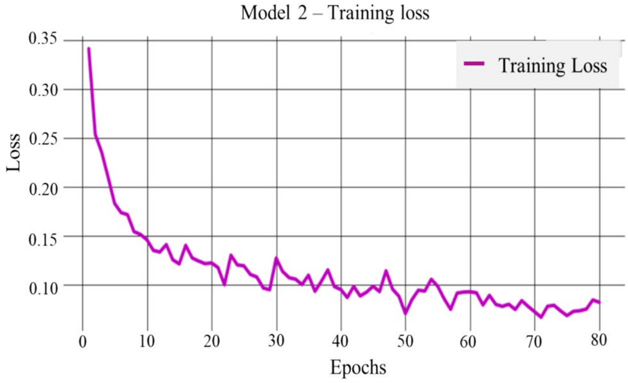


Fig. 25 Training loss of Model 2

By analyzing the F1-scores (Eq. 3) of all models in Table 6, we conclude that Model 2 outperforms the others.

The testing dataset is analyzed using the YOLO algorithm and the result is investigated. Figure 27 displays the printed message of actual and predicted counts for both pneumonia and normal X-rays separately. Based on this data, a classification report is generated, as shown in Fig. 28. Examining this report, we see that the prediction and recall for pneumonia are 0.99 and 0.37, respectively, while for normal X-rays, these values are 0.70 and 1.0. This indicates that the model exhibits high precision for identifying pneumonia but lower recall, meaning it may miss some actual pneumonia cases.

$$\text{F1 score} = 2 \times \frac{\text{PrcisionX Recall}}{\text{Prcision} + \text{Recall}} \tag{3}$$

While accuracy remains a valuable metric, the F1-score (Eq. 3) offers a more in-depth understanding of performance by considering both precision and recall (false positives and false negatives). In complex domains like healthcare, where imbalanced class distributions are common, F1-score often proves more informative than accuracy alone. Our analysis confirms that Model 2 achieves a superior F1-score compared to other models. However, given the critical nature of pneumonia diagnosis

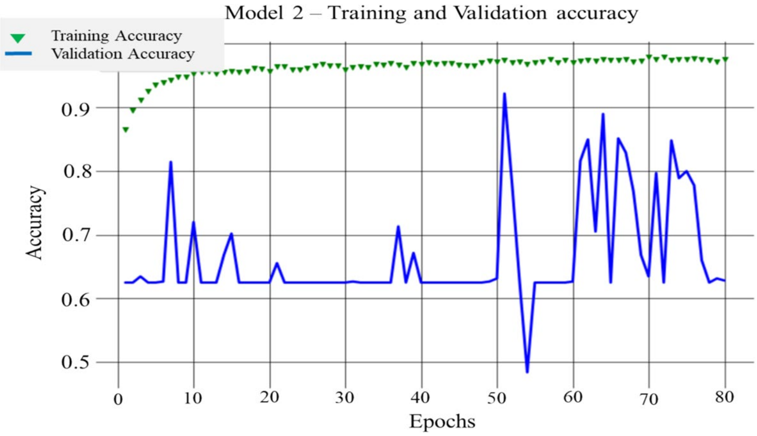


Fig. 24 Training and validation accuracy of Model 2

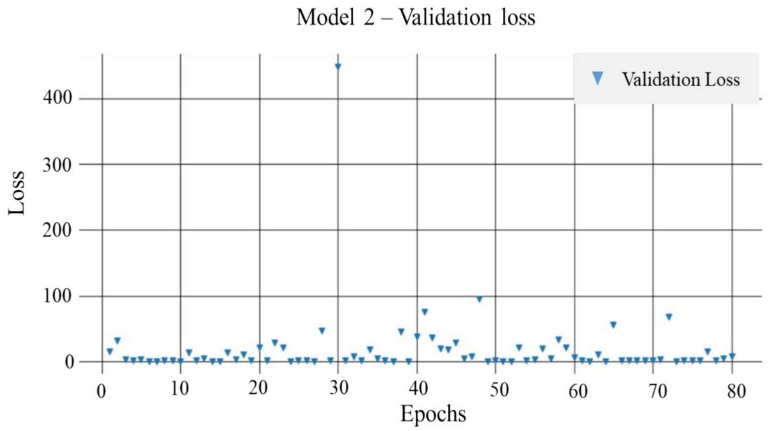


Fig. 26 Validation loss of Model 2

in the medical field, further refinements are necessary to enhance the model's recall specifically for the pneumonia class. This would minimize the risk of missed diagnoses, further improving the model's clinical relevance.

Figure 29 illustrates the inverse relationship between precision and recall. As precision increases, recall tends to decrease, demonstrating a common trade-off in classification models. Figure 30 depicts the ROC curve, which exhibits a high true positive rate (sensitivity) with a low false positive rate. This indicates the model's strong ability to accurately differentiate pneumonia cases from non-pneumonia cases. The curve's proximity to the top-left corner of the plot signifies a low overall error rate in classifying both positive and negative instances. An AUC score of approximately 0.93 further supports the model's high accuracy and discriminative power, confirming its robustness for reliable pneumonia detection in CXR images. The confusion matrix is presented in Fig. 31.

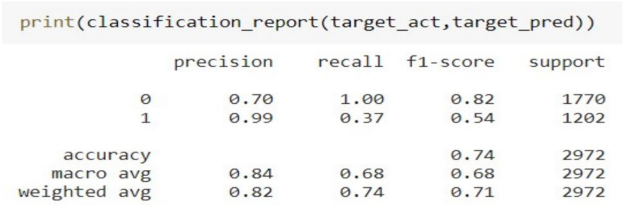
Table 6 CNN precision, recall and F1-score of test dataset

Metrics		Model 2
Precision	0	0.78
	1	0.83
Recall	0	0.80
	1	0.81
F1-score	0	0.799
	1	0.819

4.3 Qualitative Results

4.3.1 Images with Prediction

Following training and validation, the YOLO model generated predictions on the testing dataset. Figure 32 showcases some pneumonia detection results. Each case



```
print(classification_report(target_act,target_pred))

              precision    recall  f1-score   support

           0       0.70      1.00      0.82      1770
           1       0.99      0.37      0.54      1202

    accuracy                           0.74      2972
   macro avg       0.84      0.68      0.68      2972
weighted avg       0.82      0.74      0.71      2972
```

Fig. 28 Classification report of YOLO model

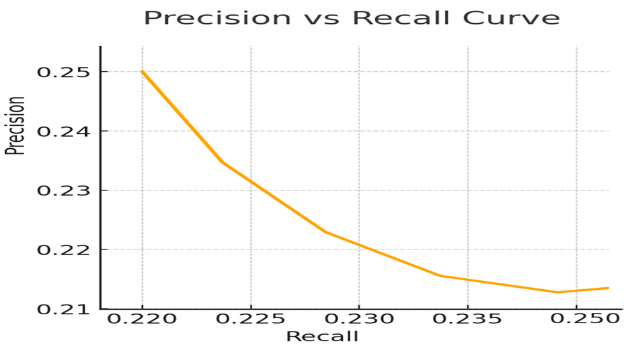
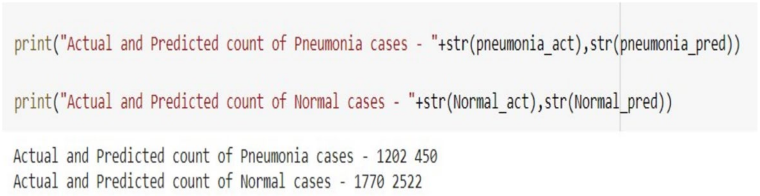


Fig. 29 Precision vs recall curve



```
print("Actual and Predicted count of Pneumonia cases - "+str(pneumonia_act),str(pneumonia_pred))

print("Actual and Predicted count of Normal cases - "+str(Normal_act),str(Normal_pred))
```

```
Actual and Predicted count of Pneumonia cases - 1202 450
Actual and Predicted count of Normal cases - 1770 2522
```

Fig. 27 Actual and predicted count cases

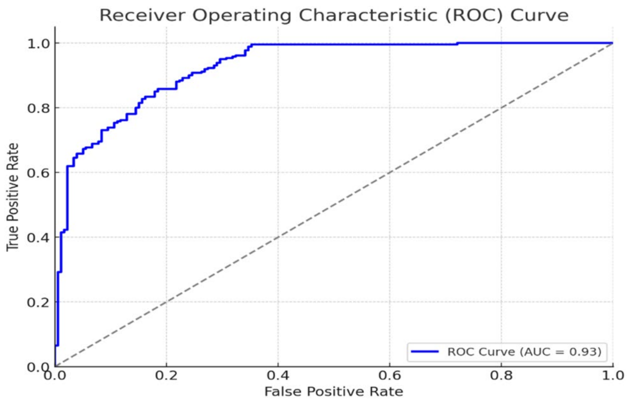


Fig. 30 ROC curve

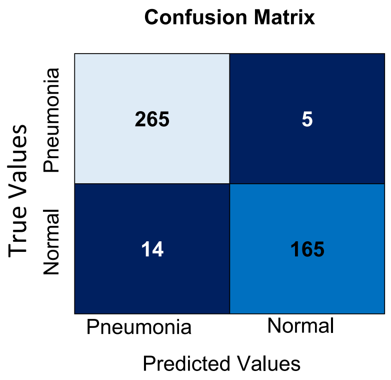


Fig. 31 Confusion matrix

displays the actual image (with known coordinates), the predicted image generated by the model, and the bounding box (red) indicating the predicted region of interest (ROI) for pneumonia. Additionally, the model classifies the image as either pneumonia or normal X-ray. The predicted class and ROI accuracy with the actual images are given in Figs. 33

Figures 33 illustrate actual and predicted results for normal X-rays. Since pneumonia detection is not pertinent here, these figures solely focus on the model's classification output, comparing the predicted "normal" label with the actual image content.

Apart from correctly predicted images, we also encounter error-predicted images, indicating incorrect classifications, changes in the Region of Interest (ROI) area, or inaccuracies in ROI accuracy. These errors highlight areas where the model's performance can be further refined.

Recent studies on pneumonia detection using deep learning have reported comparable yet varied performance metrics, primarily focusing on classification accuracy without extensive localization capabilities. A classic CNN-based architecture for pneumonia detection [19] on a similarly diverse CXR dataset, achieving an F1-score of 0.77 for pneumonia cases and an overall validation accuracy of 0.80. Compared to our approach, which achieved 0.83 validation accuracy and an F1-score of 0.819 for pneumonia, our model offers a modest but meaningful improvement, likely attributable to data augmentation and dropout regularization that reduced overfitting on pneumonia features.

Similar study using a CNN model [13] achieved an accuracy of 0.81 on a dataset with similar preprocessing techniques but lacked specific localization capabilities. Our best CNN model, however, achieved a training accuracy of 0.968 and validation accuracy of 0.83, indicating higher discriminative ability in both training and validation phases. Furthermore, our use of a one-stage object detection model, YOLO, added the benefit of localization with an F1-score of 0.82 for normal cases and 0.54 for pneumonia, offering a weighted average of 0.71. This localization feature allows clinicians to pinpoint pneumonia-affected regions more accurately—a capability not addressed in traditional CNN models.

The combined results from our CNN and YOLO implementations not only validate the robustness of our proposed model in classifying pneumonia with an F1-score of 0.819 but also demonstrate a significant step forward in detecting and localizing affected regions, enhancing both detection reliability and diagnostic applicability in clinical settings.

While our proposed model demonstrates promising performance in pneumonia detection, it is essential to acknowledge its limitations. One significant challenge lies in the difficulty of accurately differentiating pneumonia from other similar lung pathologies, such as tuberculosis or lung cancer. Additionally, variations in image quality across different datasets can impact the model's performance.

Furthermore, inconsistencies in image annotations, particularly regarding the delineation of pneumonia-affected regions, can contribute to lower precision and recall values. To mitigate these issues, future research could explore advanced techniques such as attention mechanisms and ensemble learning. Additionally, the development of larger, more diverse datasets with high-quality annotations

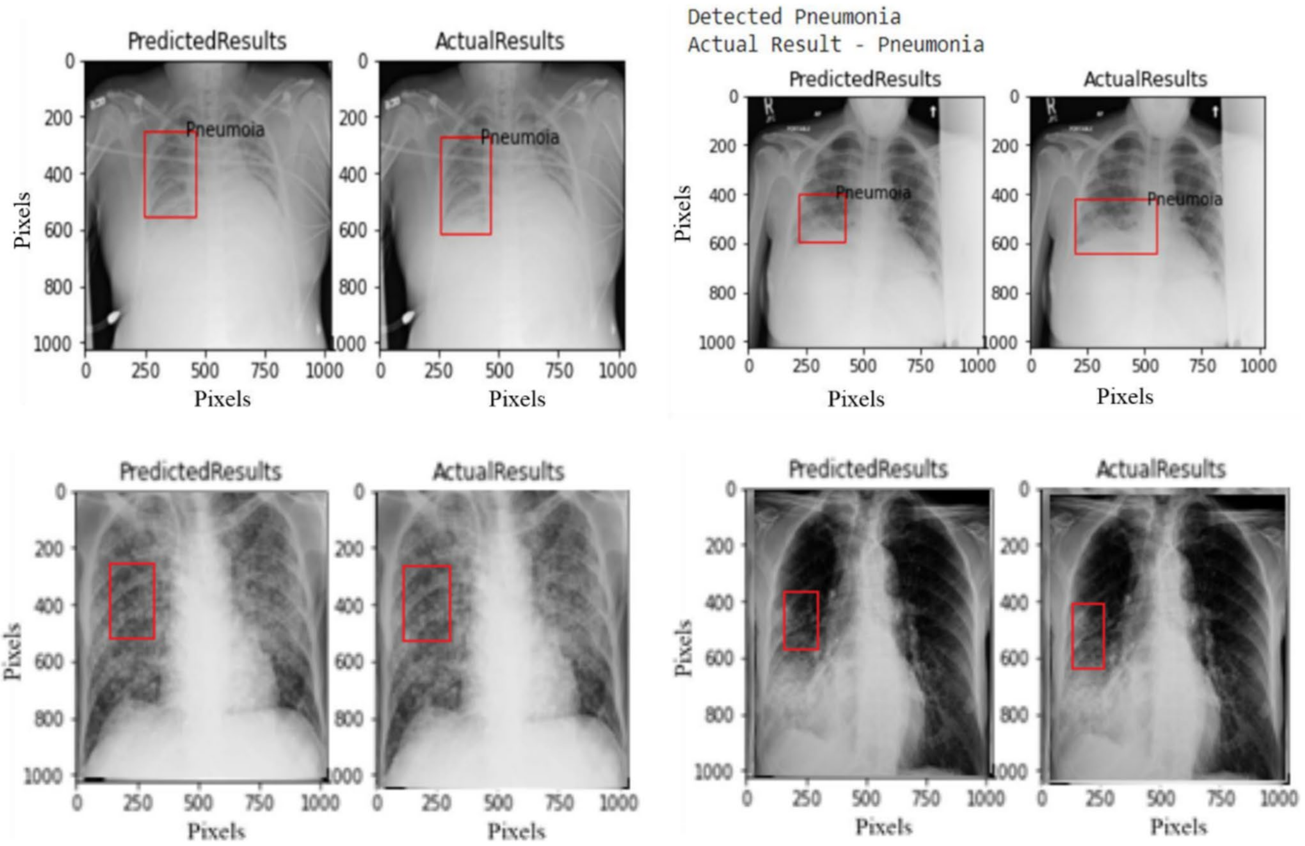


Fig. 32. Correctly detected pneumonia image

is crucial for improving the accuracy and robustness of pneumonia detection models.

The outcomes of this proposal demonstrate significant potential for future applications. By integrating a dedicated embedded edge computing device, we can streamline the automated pneumonia detection process. This device would facilitate rapid and accurate decision-making, enabling physicians to promptly assess patients and administer appropriate medication. Such a system has the potential to revolutionize healthcare delivery, especially in resource-constrained settings, by providing timely and reliable diagnostic support.

5 Conclusion

This work explores the use of machine learning for automated pneumonia detection in chest X-rays, demonstrating its promise for efficient and accurate diagnosis. The proposed approach utilized diverse chest X-ray (CXR) datasets preprocessed for consistency and built upon a baseline convolutional neural network (CNN) model for pneumonia classification. To enhance performance, three additional models incorporating data augmentation and dropout techniques were proposed. The best-performing model achieved high discriminative ability with 0.968 training accuracy and 0.83 validation

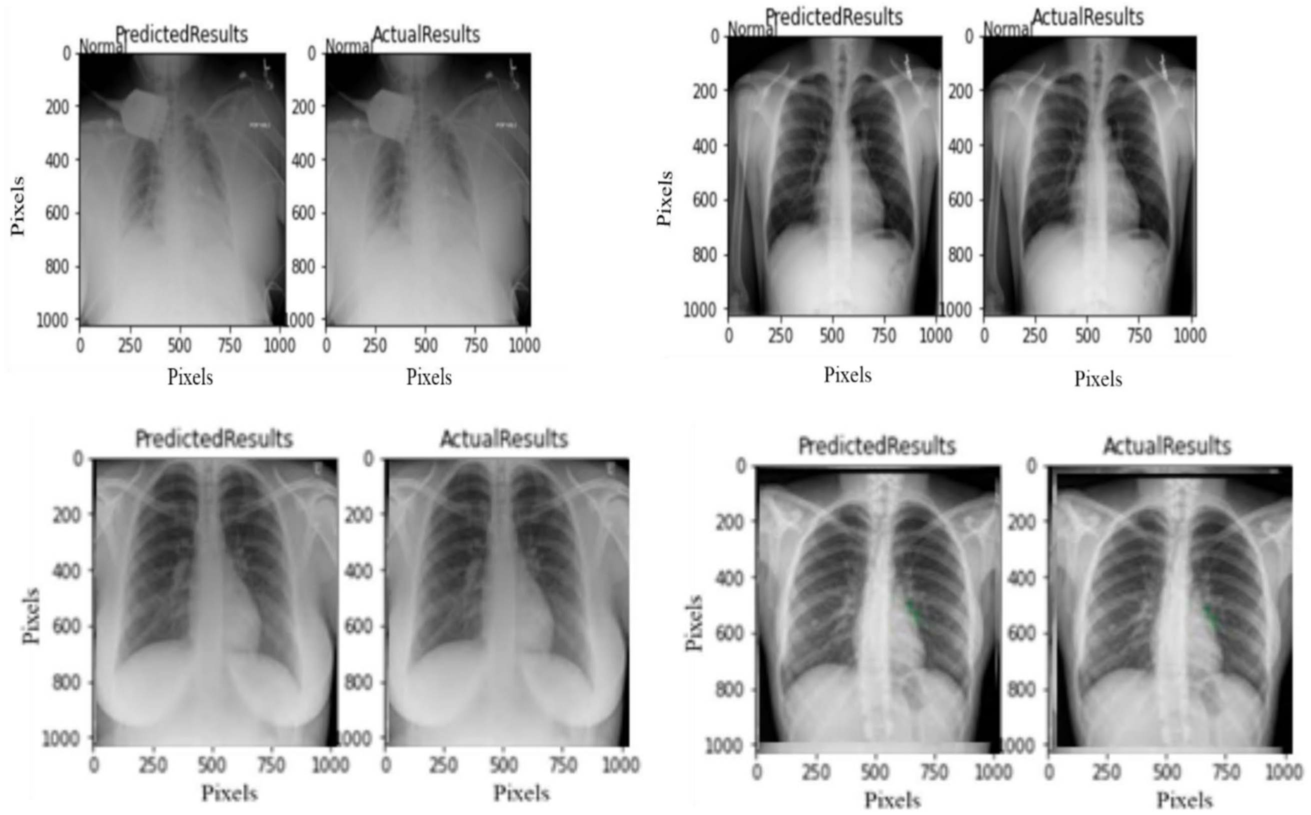


Fig. 33 Correctly detected normal image

accuracy. The predicted F1-score obtained for this model was 0.799 for normal images and 0.819 for pneumonia images. Furthermore, a one-stage object detector, YOLO, was implemented for improved efficiency and localization of affected regions. The F1-score for normal X-ray images was 0.82, and for pneumonia X-ray images, it was 0.54, resulting in a weighted average of 0.71 and a macro average of 0.68.

Acknowledgements This research was supported by the Rashtriya Uchchatar Shiksha Abhiyan (RUSA) 2.0/PO5, Dept. of Electrical and Electronics Engineering. College of Engineering, Anna University, Chennai, Tamil Nadu, India.

Author Contributions RK (RATHNAKANNAN KAILASAM): RK conceived the Idea and concept, developed the algorithms and prepared the manuscript. SB (Saranya Balasubramanian): SB contributed by providing and validating medical insights related to pneumonia, ensuring the accuracy of the study's clinical context. All the authors have read and approved the final manuscript.

Funding Not applicable.

Availability of Data and Materials The data and materials used in this research are available upon request.

Declarations

Conflict of interest Not applicable.

Ethics approval and consent to participate Not applicable.

Consent for publication All authors of this paper consent to its publication.

References

1. Salido JAA, Ruiz C Jr. Using deep learning to detect melanoma in dermoscopy images. Int J Mach Learn Comput. 2018;8(1):18–23. https://doi.org/10.18178/ijmlc.2018.8.1.664.
2. Lu Y, Zhang L, Xie W. YOLO-compact: an efficient YOLO network for single-category real-time object detection. In: 2020 Chinese control and decision conference (CCDC). 2020. pp. 1931–6. https://doi.org/10.1109/CCDC49329.2020.9164580.

3. Pathaka R, Pandeya M, Rautaraya S. Application of deep learning for object detection. Procedia Comput Sci. 2018;132:1706–17. https://doi.org/10.1016/j.procs.2018.05.147.
4. Rajaraman S, Kim I, Antani SK. Detection and visualization of abnormality in chest radiographs using modality-specific convolutional neural network ensembles. PeerJ. 2020;8: e8693. https://doi.org/10.7717/peerj.8693.
5. Jaiswal AK, Tiwari P, Kumar S, Gupta D, Khanna A, Rodrigues JJP. Identifying pneumonia in chest X-rays: a deep learning approach. Measurement. 2019;145:511–8. https://doi.org/10.1016/j.measurement.2019.05.076.
6. Chest X-ray pneumonia dataset. Kaggle. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.
7. Joby R, Raju RJ, Job R, Thomas SM, AK S. PneumoGAN: a GAN-based model for pneumonia detection. In: 2021 2nd international conference on advances in computing, communication, embedded and secure systems (ACCESS). 2021. pp .102–6. https://doi.org/10.1109/ACCESS51619.2021.9563341
8. Lakhani P, Sundaram B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology. 2017;284(2):574–82. https://doi.org/10.1148/radiol.2017162326.
9. Patil SA, Udpi VR. Chest X-ray features extraction for lung cancer classification. J Sci Ind Res. 2010;69(4):271–7.
10. Mohsen H, El-Dahshan EA, El-Horbaty ESM, Salem AM. Classification using deep learning neural networks for brain tumors. Future Comput Inform J. 2018;3(1):68–71. https://doi.org/10.1016/j.fcij.2017.12.001.
11. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
12. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med. 2020;121: 103792. https://doi.org/10.1016/j.compbiomed.2020.103792.
13. Yao S, Chen Y, Tian X, Jiang R, Ma S. An improved algorithm for detecting pneumonia based on YOLOv3. Appl Sci. 2020;10(5):1818. https://doi.org/10.3390/app10051818.
14. Nithya TM, Rajesh Kanna P, Vanithamani S, Santhi P. An efficient PM—multisampling image filtering with enhanced CNN architecture for pneumonia classification. Biomed Signal Process Control. 2023;86: 105296. https://doi.org/10.1016/j.bspc.2023.105296.
15. Chandra TB, Verma K, Singh BK, Jain D, Netam SS. Automatic detection of tuberculosis-related abnormalities in chest X-ray images using hierarchical feature extraction scheme. Expert Syst Appl. 2020;158: 113514. https://doi.org/10.1016/j.eswa.2020.113514.
16. Duong LT, Le NH, Tran TB, Ngo VM, Nguyen PT. Detection of tuberculosis from chest X-ray images: boosting the performance with vision transformer and transfer learning. Expert Syst Appl. 2021;184: 115519. https://doi.org/10.1016/j.eswa.2021.115519.
17. Rajan BP, Shree AJ, Rathnakannan K. Binarized neural network with depth imaging techniques for stock return direction prediction. J Ambient Intell Human Comput. 2023;14:3899–912. https://doi.org/10.1007/s12652-022-04460-1.
18. Liz H, Huertas-Tato J, Sánchez-Montañés M, Del Ser J, Camacho D. Deep learning for understanding multilabel imbalanced Chest X-ray datasets. Future Gener Comput Syst. 2023;144:291–306. https://doi.org/10.1016/j.future.2023.03.005.
19. Szepesi P, Szilágyi L. Detection of pneumonia using convolutional neural networks and deep learning. Biocybern Biomed Eng. 2022;42(3):1012–22. https://doi.org/10.1016/j.bbe.2022.08.001.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.