Pertanika J. Sci. & Technol. 34 (5): 2569 - 2596 (2026)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Pneumonia Screening and Prediction on Images Using a Convolutional Neural Network Deep Learning Model

Shuria Saaidin[1], Murizah Kassim[1,2]*, I Gede Pasek Suta Wijaya[3], and Nur Amirah Kamaluddin[1]

[1]Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[2]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Kompleks Al-Khawarizmi, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[3]Department of Informatics Engineering, Faculty of Engineering, University of Mataram, Jalan Majapahit No. 62, Mataram 83125, West Nusa Tenggara, Indonesia

ABSTRACT

Pneumonia remains a leading cause of morbidity and mortality worldwide, and its radiographic diagnosis depends on expert readers who are unavailable in many settings. This study reports a comparative evaluation of two ImageNet-pretrained convolutional architectures, VGG16 and Inception-v3, for binary pneumonia classification on chest radiographs. The publicly available Kaggle chest radiograph collection was used, restricted to the Normal and Pneumonia classes: 5,216 training images (1,341 normal, 3,875 pneumonia) and an independent test partition of 624 images (234 normal, 390 pneumonia). Each architecture was evaluated under two regimes: feature extraction with a frozen convolutional base and fine-tuning of the upper convolutional blocks. All performance figures are reported on the held-out test partition with 95% Wilson confidence intervals and are accompanied by specificity, balanced accuracy, and the Matthews correlation coefficient, as the test partition is imbalanced. The fine-tuned VGG16 model achieved an accuracy of 0.905 (0.880-0.926), sensitivity of 0.956, specificity of 0.821, F1-score of 0.927, and MCC of 0.797. The fine-tuned Inception-v3 model achieved an accuracy of 0.936 (0.914-0.953), sensitivity 0.951, specificity 0.910, F1-score 0.949 and MCC 0.863, at approximately 60% of the training time per epoch required by VGG16. Fine-tuning improved VGG16 by raising specificity at the cost of sensitivity but produced no measurable change in Inception-v3. These held-out figures are substantially below the near-perfect accuracies commonly reported

ARTICLE INFO

Article history:
Received: 26 May 2025
Accepted: 17 August 2026
Published: 30 October 2026

DOI: https://doi.org/10.47836/pjst.34.5.02

E-mail addresses:
shuria6809@uitm.edu.my (Shuria Saaidin)
murizah@uitm.edu.my (Murizah Kassim)
gpsutawijaya@unram.ac.id (I Gede Pasek Suta Wijaya)
nuramirahkamaluddin97@gmail.com (Nur Amirah Kamaluddin)
* Corresponding author

e-ISSN: 2231-8526

© Universiti Putra Malaysia Press

for these architectures on this benchmark, a discrepancy attributable to the reporting of training-phase rather than test-phase performance.

Keywords: Chest radiography, convolutional neural network, model evaluation, pneumonia screening, transfer learning

INTRODUCTION

Pneumonia continues to be a significant health challenge worldwide, particularly in developing countries (WHO, 2025). In 2021 alone, it was responsible for approximately 2.5 million fatalities, including about 700,000 children under five years old (UNICEF, 2024). Prompt and precise diagnosis of pneumonia is crucial to improve patient outcomes and reduce mortality (Lim, 2022). Chest X-ray imaging is a standard diagnostic tool for pneumonia; however, interpreting these radiographs relies on expert radiologists who may not be readily available, especially in resource-limited settings. The global shortfall in healthcare workers, projected to reach 18 million by 2030, further exacerbates the challenge of timely and reliable pneumonia diagnosis (Boniol et al., 2022). Recently, deep learning techniques, especially CNNs, have been applied to automate medical image analysis tasks. CNN-based models have demonstrated strong performance in detecting pneumonia from chest radiographs (Aljuaid et al., 2026). For example, a CNN using the VGG16 architecture achieved high diagnostic accuracy on a paediatric chest X-ray dataset, outperforming earlier methods (Kermany et al., 2018). Many researchers employ transfer learning by fine-tuning CNN architectures pre-trained on large datasets like ImageNet using pneumonia-specific datasets, such as the Kaggle chest X-ray collection, which contains thousands of images labelled as either normal or pneumonia, to enhance classification performance (Mahalle et al., 2024).

Despite this volume of work, three problems recur across the literature on this benchmark. First, most studies evaluate a single CNN architecture, so architectural differences are confounded with differences in preprocessing, optimiser settings and training schedule, which limits the generalisability of their conclusions. Second, the validation partition distributed with the dataset contains only sixteen images; studies that adopt it unmodified perform early stopping and model selection against a signal in which a single misclassification shifts the reported figure by 6.25 percentage points. Third, and therefore, reported accuracies are frequently training- or validation-phase values presented as held-out performance, and evaluation is often restricted to accuracy alone, with specificity, balanced accuracy, and the Matthews correlation coefficient omitted despite the class imbalance in the test partition. When confusion matrices are published alongside such claims, the stated accuracy is, in some cases, not recoverable from them.

The gap addressed by this study is therefore methodological rather than architectural. The aim is to establish a well-reported benchmark for two widely used CNN architectures,

VGG16 and Inception-v3, under an identical experimental protocol, and to quantify the margin by which held-out performance falls short of the figures commonly claimed. Both models are trained on the same Kaggle chest X-ray dataset with consistent preprocessing and augmentation, and each is examined under two training regimes: an initial phase with the convolutional base frozen, and a subsequent fine-tuning phase in which the upper blocks are unfrozen for domain-specific retraining. The primary focus is the held-out test partition, evaluated for accuracy, sensitivity, specificity, F1-score, balanced accuracy, the Matthews correlation coefficient, ROC-AUC, and average precision, each reported with a 95% confidence interval, together with training and inference times. The contributions are as follows. (i) A comparative evaluation of the two architectures reported entirely on held-out data with confidence intervals and statistical testing of the pairwise differences. (ii) Evidence that fine-tuning does not improve both operating characteristics uniformly but redistributes error between sensitivity and specificity, which has direct consequences for a screening application. (iii) An explicit account of the reporting practices by which accuracy on this benchmark becomes overstated. No claim is made to architectural or theoretical novelty.

RELATED WORK

This section reviews prior work on CNN-based pneumonia screening, organised by architecture, training strategy, frequency-domain methods, and evaluation practices, and concludes by identifying the limitations of the existing literature that motivate the present study.

CNN Architectures for Chest Radiography

Several studies have been reviewed in this research to gain insight into the standard deep learning architectures used in settings like ours. It was observed that most of them use CNNs and combine the method with a variety of deep learning architectures for pneumonia detection. For example, models such as VGG16, ResNet, DenseNet, and Inception-v3 have been shown to identify pneumonia in chest X-ray images, achieving high accuracy (Shin et al., 2016; Tan & Le, 2019). However, these studies agree that the common challenge they face is the scarcity of large, annotated datasets. Most studies have used transfer learning as a powerful technique to mitigate this issue, and this approach is also applied in this research. By training CNNs on large-scale general image datasets and then fine-tuning them for the target task, such as pneumonia X-ray classification, researchers can improve model accuracy while reducing training time (Abd Rahman et al., 2022; Raghu et al., 2019; Yamashita et al., 2018). In addition, ensemble methods, which combine predictions from multiple models, have been shown to further improve diagnostic accuracy by leveraging the complementary strengths of different architectures, resulting in more robust and reliable predictions. (Azis et al., 2022; Gulshan et al., 2019; Hu et al., 2024).

Data Augmentation and Preprocessing

Similar to transfer learning, data augmentation techniques are often employed to boost the performance of deep learning models, particularly when training data is limited. Common augmentations include geometric transformations such as rotations, translations, zooms, and flips, which artificially expand the diversity of the training set. (Sharma et al., 2024). Proper preprocessing of chest X-ray images, like intensity normalisation and contrast enhancement, is also important for improving model performance. Together, these augmentation and preprocessing steps help models generalise better when used with new, unseen data. The effectiveness of deep learning models is typically evaluated using various metrics; for instance, accuracy quantifies the overall correctness of predictions. (El Houby, 2024). The following section describes the performance metrics typically used to evaluate deep learning performance.

Frequency-Domain and Hybrid Architectures

A parallel line of work departs from purely spatial-domain feature extraction by embedding frequency-domain representations directly within the network. Fourier Analysis Network formulations replace conventional dense activations with sine and cosine components, so that a quasi-periodic structure is modelled explicitly rather than approximated by successive spatial convolutions. This principle has been applied to medical image classification in the CONV-FAN-POX architecture, which combines an EfficientNetV2 backbone with Fourier analysis blocks and reports a mean accuracy of 0.9881 and an F1-score of 0.9856 across five-fold cross-validation on the six-class augmented MSLD2.0 dataset, using approximately seven million trainable parameters, substantially fewer than comparable spatial-domain networks (Tülü & Kaya, 2026). Similarly, hybrid attention-based architectures have integrated hierarchical Swin Transformer feature extraction with convolutional neural networks, leveraging self-attention and cross-attention mechanisms to capture both local and global representations for pneumonia classification from chest X-ray images, achieving an accuracy of 98.56% on the publicly available Chest X-Ray dataset and demonstrating improved robustness over earlier deep learning methods (Kunjumon et al., 2026). Notably, the same architecture attains a mean accuracy of 0.8176 on the original, unaugmented partition of that dataset, and the authors report both figures explicitly rather than the higher one alone. Such approaches are relevant to chest radiography because pulmonary texture carries diagnostic information across multiple spatial frequencies that a fixed convolutional receptive field captures only indirectly. They are not evaluated here, since the present study is deliberately confined to the two architectures whose reported performance on this benchmark it seeks to correct. Still, they represent a more promising direction for architectural contribution than further replication of standard transfer-learning pipelines.

Performance Metrics

There are several performance metrics to assess pneumonia detection models, including accuracy, which represents the overall proportion of correct predictions. The second metric is sensitivity or recall, which is the proportion of actual positives, such as pneumonia cases, that were correctly identified as true positives. Lastly, there is specificity, which is the ability to correctly identify normal or non-pneumonia cases among all actual normal cases, such as the true negative rate. The AUC-ROC is a single statistic that summarises, over all classification thresholds, sensitivity as a function of 1-specificity (Müller et al., 2022; Namdar et al., 2021). In the present study, all these indices were calculated simultaneously to provide a comprehensive and detailed assessment of the model's diagnostic performance. With the development of deep learning techniques, their application in clinical practice has great potential to change the pattern of pneumonia screening by enabling fast, accurate diagnosis. There is a need for continued research and collaboration between AI and the medical community to address the remaining challenges and realise the technology's full potential (Abdulahi et al., 2024; Basubrin, 2025). As the field develops, integrating these models into clinical practice could significantly improve patient outcomes and alleviate the strain on the healthcare system.

Limitations of Existing Work

Four (4) weaknesses recur across the studies surveyed above. First, the validation partition distributed with this benchmark contains sixteen images and is frequently adopted unmodified, so that model selection is performed against a signal with no useful resolution, and a reported validation accuracy of 100% carries almost no information. Second, several studies report training-phase or validation-phase accuracy in place of held-out test accuracy; where confusion matrices are published, the accuracy stated in the text is, in some cases, not recoverable from them. Third, the test partition of this collection is imbalanced at approximately 1.7:1 in favour of pneumonia, yet specificity, balanced accuracy, and the Matthews correlation coefficient are frequently omitted, allowing a model that performs poorly on the minority class to still present a high headline accuracy. Fourth, interpretability evidence is rarely provided, leaving unresolved whether classification is driven by pulmonary findings or by acquisition and positioning artefacts that correlate with class membership in this collection. The present study is directly opposed to these four points.

METHODOLOGY

Figure 1 illustrates the research tasks, which were divided into seven phases. The phases included dataset preparation, model development, training, evaluation, and prediction time analysis.

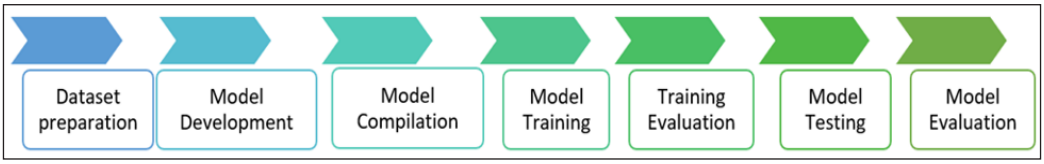
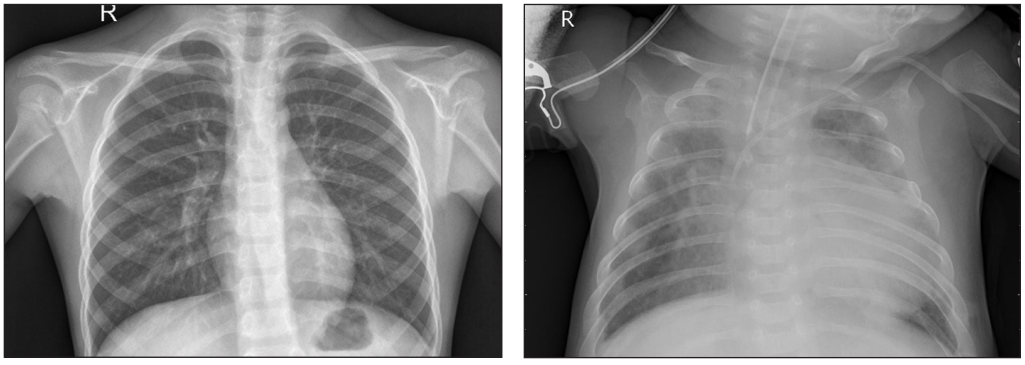


Figure 1. Research activity

Data Collection

In this study, data from the Chest X-ray dataset, including Pneumonia, COVID-19, and Tuberculosis, were utilised. The dataset is available on Kaggle, curated by (JTIPTJ, 2021). Chest X-ray images were divided into three folders: train, test, and validation. Each folder contains subfolders for the image categories: Normal, Pneumonia, COVID-19, and Tuberculosis. This dataset contains 7,135 X-ray images, providing a comprehensive set of labelled images for analysis. However, in this study, only Normal and Pneumonia images were utilised. A sample of datasets is shown in Figures 2(a) and 2(b).



(a) Normal chest X-ray (b) Chest x-ray with pneumonia

Figure 2. Sample *d*ata (a) Normal chest X-ray; (b) Chest X-ray with pneumonia

Dataset Preparation

The data were partitioned into three subsets: a training set, a validation set, and a test set. The training set contained 5,216 images, of which 1,341 were normal, and 3,875 were pneumonia, resulting in an imbalance in favour of pneumonia cases. The validation partition distributed with the dataset consists of only 16 images, eight normal and eight pneumonia, and the test partition comprises 624 images, 234 normal and 390 pneumonia. The test partition is therefore imbalanced at approximately 1.7:1 in favour of pneumonia, and accuracy alone is not an adequate summary of performance on it; specificity, balanced accuracy and the Matthews correlation coefficient are consequently reported alongside

accuracy throughout Section 4. The distributed validation partition is acknowledged to be too small to support reliable model selection or early stopping, since a single misclassification can alter the reported validation accuracy by 6.25 percentage points; therefore, validation-phase figures are reported in this work for completeness only, and no conclusion is drawn from them. The test partition was held out entirely and used once, after all training and model-selection decisions were final. For data preprocessing, Keras was used to normalise pixel values by rescaling image intensities to the range [0, 1] and to perform on-the-fly data augmentation. The augmentations applied were random rotations, shear transformations, zooming, horizontal flips, and brightness adjustments, as described in Shorten and Khoshgoftaar (2019). These transformations were applied during training to overcome data sparsity. The augmentation will also improve the models' robustness by exposing them to a wider range of images. Such normalisation and augmentation strategies are known to facilitate faster convergence and better generalisation in CNNs (Srivastava et al., 2014).

Experimental Environment

All experiments were conducted on a single workstation laptop equipped with an Intel Core i7 processor and 16 GB of system RAM. Models were implemented in TensorFlow using the Keras 3.12 API under Python 3.12, with NumPy 2.2.6. The convolutional backbones were obtained from tf.keras.applications with ImageNet weights, and each network's corresponding preprocessing input transformation was applied as an input layer to ensure preprocessing matched the convention used to learn the pretrained weights. Input images were resized to 224 × 224 × 3. The timing measurements reported in Section 4.5 were obtained on this configuration.

Model Development

Four (4) models were created to analyse the effect of initial training versus fine-tuning: VGG16 Initial Training, VGG16 Fine-Tuning, Inception v3 Initial Training, and Inception v3 Fine-Tuning. All the models were trained on top of a pre-trained base model (ImageNet weights), as in Deng et al. (2009). The architecture for VGG16 and Inception v3 consists of the Base Model, which is a pre-trained VGG16 or Inception v3 without the top classification layer (Simonyan & Zisserman, 2014; Szegedy et al., 2016). A Flatten layer was utilised to transform the 2D feature maps to 1D feature vectors. The flattened vector was passed to a fully connected dense layer, which learned non-linear combinations of the extracted convolutional features. A dropout layer was added to avoid overfitting by randomly setting a fraction of the units to zero at each update. (Srivastava et al., 2014). Finally, the output layer was a single neuron with a sigmoid as the activation for binary classification.

Training Process

The models were optimised using Adam with a learning rate of 1×10^{-4} during initial training, reduced to 1×10^{-5} during fine-tuning, and binary cross-entropy as the loss function. (Kingma & Ba, 2014). The training was completed by compiling a model. Models were built using the Adam optimiser and the binary cross-entropy loss function Equation 1.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad [1]$$

Where:
y_i is the actual label
p_i is the predicted probability
N is the number of samples.

Callbacks were used with the model checkpoint to write model weights after each epoch if the validation loss decreased. CSV Logger was extended to log training and validation scores per epoch. We used early stopping to halt training if the validation loss had not decreased in 5 epochs, and restored the best weights (Prechelt, 1998). Training metrics, including loss and accuracy, were plotted over epochs in TensorBoard. The models were configured to be trained for 20 epochs. Nevertheless, early stopping was used to terminate the training if the validation loss did not decrease for five consecutive epochs, thereby avoiding overtraining and limiting resource expenditure.

Performance Evaluation

The models were evaluated during the training and testing phases to comprehensively assess their performance.

Training Phase

The training phase was done to check training accuracy. The ratio of correctly predicted instances to the total instances in the training set. Accuracy was calculated using Equation 2. The loss value calculated on the training set indicates how well the model is learning. The ratio of correctly predicted instances to the total instances in the validation set. The loss value calculated on the validation set is used to monitor overfitting.

$$Accuracy = \frac{Number\ \ of\ correct\ \ predictions}{Total\ \ number\ \ of\ predictions} \quad [2]$$

Testing Phase

The testing phase is done to identify the ROC AUC Curve. The Receiver Operating Characteristic curve and the area under it indicate the model's ability to distinguish between classes (Bradley, 1997) as Equation 3.

$$AUC = \int_0^1 TPR(FPR)d(FPR) \quad [3]$$

Where:
TPR is the true positive rate
FPR is the false positive rate

A Precision-Recall Curve was plotted to compare precision versus recall at various threshold settings, providing insights into the trade-off between precision and recall for different decision thresholds (Davis & Goadrich, 2006).

$$Precision = \frac{True\ positives}{True\ positives + False\ positives} \quad [4]$$

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives} \quad [5]$$

Hyperparameters were identified, including the Learning Rate. The learning rate was set to 1×10^{-4} while training the custom top layers, with the convolutional base frozen, and was reduced by an order of magnitude to 1×10^{-5} during the fine-tuning phase to avoid disrupting the pretrained representations with large gradient updates. A batch size of 32 was used for both initial training and fine-tuning. Initially, the epoch was set to 20. However, each training phase consisted of multiple epochs, with early stopping implemented to terminate training if validation performance did not improve. The prediction time for each model was measured to assess its efficiency in real-time applications. The inference times for each model on a batch of images were measured and compared. In the present work, two existing CNN architectures, VGG16 and Inception, were compared in their ability to detect pneumonia in chest X-ray images. The evaluation metrics included training accuracy, validation accuracy, training loss, validation loss, precision, recall, F1 Score, and AUC-ROC. Training and validation accuracy evaluate the performance of the models in classifying pneumonia cases from the training data and from previously unseen data points, respectively. On the other hand, training and validation loss signal the error rates during these periods. Precision evaluates true-positive predictions among all positive predictions but reduces the overall number of false positives. Recall assesses the model's performance in detecting positive cases, which reduces false negatives. The F1-score (the harmonic mean of precision and recall) is a balanced measure for evaluating the model's

performance. AUC-ROC curve describes how well the models differentiate between positive and negative cases at various thresholds. The overall evaluation values indicate the predictive ability, reliability, and potential effectiveness of the models in clinical practice, which can enhance pneumonia detection.

Formal Specification and Additional Evaluation Metrics

The classification pipeline is specified formally as follows. Let x denote an input radiograph after resizing, and let r(x) = x/255 denote the rescaling operator mapping intensities to the interval [0, 1]. During training, a stochastic augmentation operator, A (·), is applied, composed of random rotation, shear, zoom, horizontal flip and brightness adjustment; A is the identity at validation and test time. Let f(·) denote the pretrained convolutional feature extractor with parameters θ, and g (·) the classification head with parameters φ, comprising a flattening operation, a dense layer, a dropout layer and a single sigmoid unit. The predicted probability of the pneumonia class is given by Equation 6.

$$p = g_{\varphi}(f_{\theta}(A(r(x)))),\ p \in (0, 1) \quad [6]$$

In the initial training phase, the feature-extractor parameters θ are held fixed at their ImageNet-pretrained values and only φ is updated, as in Equation 7. In the fine-tuning phase, the upper convolutional blocks θ_top □ θ are unfrozen and updated jointly with φ at a reduced learning rate, as in Equation 8. In both phases, the objective L is the binary cross-entropy given in Equation 1.

$$\varphi^* = \arg\min_{\varphi} L (\varphi; \theta \text{ fixed}) \quad [7]$$

$$(\theta_{top}^*, \varphi^*) = \arg\min_{(\theta_{top}, \varphi)} L(\theta_{top}, \varphi) \quad [8]$$

In inference, the decision rule assigns the pneumonia class when the predicted probability reaches a threshold τ, fixed at 0.5 throughout this study, as in Equation 9.

$$\hat{y} = 1 \text{ if } p \geq \tau, \text{ and } \hat{y} = 0 \text{ otherwise, with } \tau = 0.5 \quad [9]$$

Because the test partition is imbalanced, three additional measures are reported alongside the accuracy, precision, and recall defined in Equations 2, 4, and 5. Specificity, given in Equation 10, quantifies the proportion of normal cases correctly identified. The F1-score, given in Equation 11, is the harmonic mean of precision and recall. Balanced accuracy, given in Equation 12, is the unweighted mean of sensitivity and specificity and is insensitive to class prevalence. The Matthews correlation coefficient, given in Equation 13, summarises all four cells of the confusion matrix in a single value bounded by −1

and +1 and is the most conservative of the summary measures used here. In Equations 10 to 13, TP, TN, FP, and FN denote true positives, true negatives, false positives and false negatives, respectively.

$$\text{Specificity} = TN / (TN + FP) \quad [10]$$

$$F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad [11]$$

$$\text{Balanced Accuracy} = (\text{Sensitivity} + \text{Specificity}) / 2 \quad [12]$$

$$MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{[\ (TP+FP)(TP+FN)(TN+FP)(TN+FN)\ ]} \quad [13]$$

RESULTS AND DISCUSSIONS

Model Accuracy

Figure 3 visualises the training and validation accuracy trends for the four trained models. Evaluating the VGG16 and Inception-v3 models before and after fine-tuning reveals notable insights into their classification capabilities. The VGG16 model exhibited a promising learning trajectory in its initial training phase. Starting with a training accuracy of 86.73%, it improved steadily to about 97.48% by the final epoch. The validation accuracy for VGG16 showed greater fluctuation, starting at around 75% in the first epoch, dipping to 68.75% in the second, and reaching 100% by the seventh epoch. Because the distributed validation partition contains only 16 images, a validation accuracy of 100% corresponds to sixteen correct classifications and carries very little information; these validation figures are reported to describe the course of training and are not estimates of generalisation.

This variability suggests that while the model was learning effectively on the training data, it struggled to consistently generalise to the validation data early on, possibly due to the limited size of the validation set. Upon fine-tuning, the performance of VGG16 improved significantly on both the training and validation sets. The fine-tuned VGG16 model's training accuracy started at around 86.61% when we resumed training with the pre-trained weights and quickly climbed to 99.59%. More importantly, its validation accuracy reached 100% by the fifth epoch of fine-tuning and remained perfect across several subsequent epochs. This reflects performance on the 16-image validation partition and does not indicate generalisation; the corresponding held-out test accuracy is 0.9054, as reported in Table 5.

The Inception-v3 model demonstrated robust performance from the outset of initial training. Its training accuracy began at 92.67% and improved to 99.11% throughout the training epochs. The validation accuracy for Inception was also high initially, around 96.36%, with a slight dip in some later epochs, but generally remaining strong and stable. The peak validation accuracy during initial training was 97.31%, which was observed around the 11th epoch, indicating that Inception-v3 was able to generalise well early on.

Fine-tuning further slightly boosted Inception's performance. The fine-tuned Inception model's training accuracy improved from an already high 93.72% to 99.59%. Its validation accuracy also increased, peaking at 99.52% by about the 17th epoch. This suggests that fine-tuning helped Inception to solidify its already high accuracy, achieving near-perfect validation accuracy. Overall, both models demonstrated substantial accuracy improvements through training, with fine-tuning providing an extra performance boost. VGG16 benefited more dramatically from the fine-tuning phase (achieving perfect validation accuracy only after fine-tuning), whereas Inception-v3 showed strong performance throughout and modest gains from fine-tuning. Table 1 compares the initial and final training and validation accuracies for all models, summarising these observations.

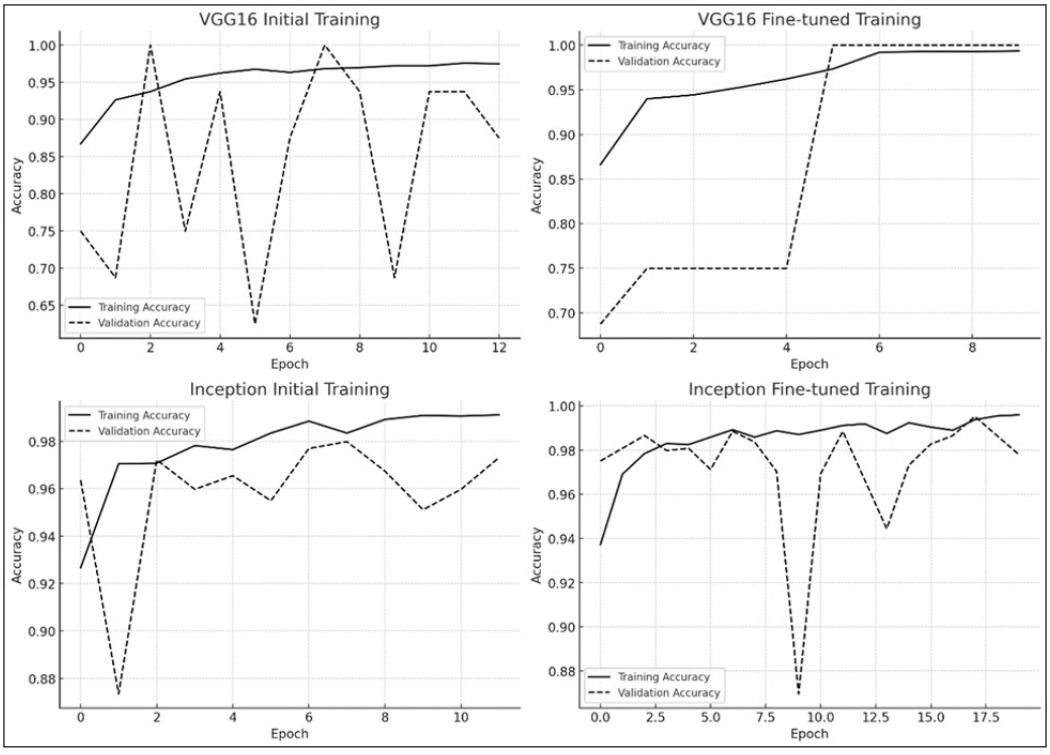


Figure 3. The evaluation of VGG16 and inception models

Table 1 shows that both VGG16 and Inception models demonstrated significant improvements through the training phases, with fine-tuning substantially enhancing their performance. While VGG16 showed considerable fluctuations in validation accuracy during initial training, fine-tuning enabled it to consistently achieve perfect validation accuracy. Conversely, Inception displayed strong performance throughout, with minimal volatility and substantial improvements post-finetuning. The overall analysis highlights the robustness

of Inception and the significant enhancement of VGG16 through fine-tuning, making both models highly effective for classification tasks.

Table 1
Comparison of training and validation accuracies for all models

Model	Training Phase	Initial Training Accuracy	Final Training Accuracy	Initial Validation Accuracy	Final Validation Accuracy
VGG16	Initial	86.73%	97.48%	75.00%	87.50%
VGG16	Fine-tuned	86.61%	99.59%	68.75%	100.00%
Inception	Initial	92.67%	99.11%	96.36%	97.31%
Inception	Fine-tuned	93.72%	99.59%	97.51%	99.52%

Loss Performance for All Models

Figure 4 visualises the training loss and validation loss for all models. An Initial Training was identified for the VGG16 model, and the loss performance exhibited a distinctive pattern. The training loss decreased significantly, from 0.30 to 0.0659, indicating effective learning. However, the validation loss started high at 0.7976, peaked at 0.9322 in the second epoch, and then dropped sharply to 0.1235 by the third epoch. These fluctuations indicate periods of overfitting and recovery. The lowest validation loss was recorded at 0.0835 in epoch 7, suggesting moments of improved generalisation despite initial challenges. Fine-tuning the VGG16 model resulted in a substantial decrease in training loss, from 0.2863 to 0.0114, demonstrating improved learning efficiency. The validation loss showed a general downward trend, starting at 0.4904, with minor fluctuations. Notable peaks in validation loss were observed early in the training, indicating initial overfitting. However, these were followed by a significant drop, with the lowest validation loss of 0.0495 recorded in epoch 7. This reflects that fine-tuning improved the model's generalisation ability, effectively reducing errors on unseen data.

The initial training of the Inception model displayed a different pattern. The training loss decreased significantly from 0.1651 to 0.0297, highlighting efficient learning. The validation loss started low at 0.0907, then increased to 0.3605 in the first epoch, and fluctuated throughout training. These fluctuations suggest intermittent overfitting. Despite these variations, the lowest validation loss was 0.0361 at epoch 11, demonstrating strong generalisation during specific periods. In the fine-tuned training phase, the Inception model's training loss decreased from 0.1636 to 0.0114, indicating continued improvement in learning efficiency. The validation loss showed a general downward trend, starting at 0.0744, but experienced more significant fluctuations. Notable peaks in validation loss were observed at epochs 5 (0.0794), 9 (0.6852), and 13 (0.3019), indicating periods of

overfitting or model adjustments. The lowest validation loss was 0.0205 in epoch 17, reflecting substantial improvements in minimising errors on unseen data.

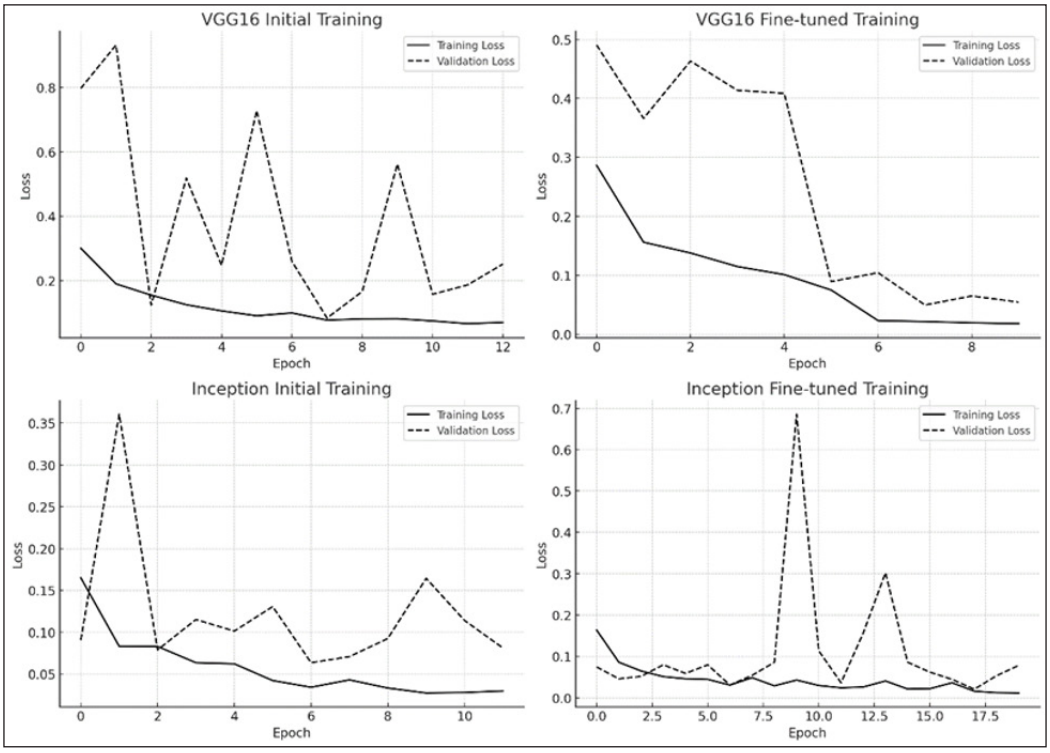


Figure 4. Training loss and validation loss for 4 models

Table 2
Comparison of training and validation loss for all models

Model	Training Phase	Initial Training Loss	Final Training Loss	Initial Validation Loss	Final Validation Loss
VGG16	Initial	0.3000	0.0659	0.7976	0.0835
VGG16	Fine-tuned	0.2863	0.0114	0.4904	0.0495
Inception	Initial	0.1651	0.0297	0.0907	0.0361
Inception	Fine-tuned	0.1636	0.0114	0.0744	0.0205

Table 2 presents the analysis of loss performance for both VGG16 and Inception models, highlighting their learning and generalisation dynamics. VGG16 exhibited substantial improvement post-finetuning, significantly decreasing training and validation losses, indicating enhanced learning and generalisation capabilities. Inception demonstrated robust learning with consistently low training and improved validation losses, particularly after fine-tuning. Despite occasional fluctuations, both models exhibited strong performance

in minimising errors and improving accuracy, underscoring their effectiveness in classification tasks.

Model Performance Evaluation

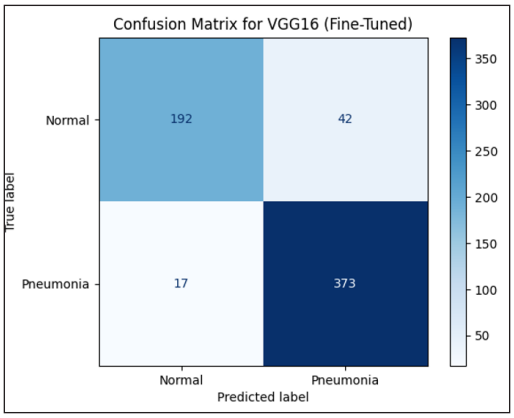
In this study, we trained and evaluated two deep learning models, VGG16 and Inception, to classify chest X-ray images as normal or pneumonia. The models were trained on a large dataset, ensuring comprehensive learning and robust performance. An independent test partition of 624 images was used for evaluation, comprising 234 normal and 390 pneumonia cases. Because this partition is imbalanced at approximately 1.7:1 in favour of pneumonia, accuracy alone would be biased towards the majority class; therefore, specificity, balanced accuracy, and the Matthews correlation coefficient are reported alongside it in Table 5. The provided figures present the confusion matrices for two deep learning models, VGG16 and Inception, evaluated before and after fine-tuning. A confusion matrix is a crucial tool for analysing classification models, as it provides a detailed breakdown of the model's performance in terms of true positives, true negatives, false positives, and false negatives. These matrices offer insights into the models' efficacy in distinguishing between normal and pneumonia cases. Figure 5 (a) to (d) illustrates the confusion matrices for all four models trained.

VGG16 Model Performance

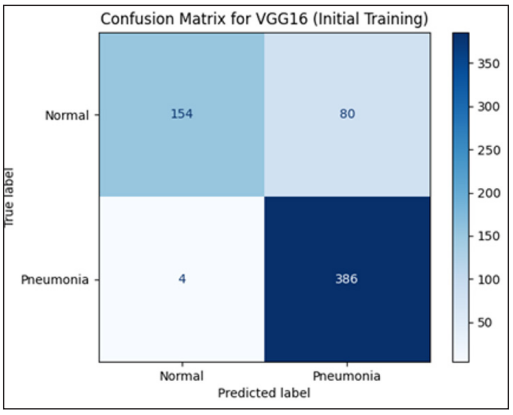
Table 3 presents the confusion matrix for the VGG-16 model at both the initial and fine-tuned stages. The initial training reveals a strong performance in identifying pneumonia cases, with a very high number of true positives and a low number of false negatives. This indicates that the model is susceptible to pneumonia and correctly identifies most pneumonia cases. However, the model struggles more with correctly classifying normal cases, as the higher number of false positives is 80, and the lower number of true negatives is 154. This suggests that while VGG16 is highly sensitive to pneumonia cases, it tends to misclassify normal cases as pneumonia. This performance imbalance suggests an opportunity to improve by increasing the model's specificity without compromising its sensitivity. After fine-tuning, the VGG16 model shows a notable improvement in correctly classifying normal cases, with a significant increase in true negatives from 154 to 192 and a decrease in false positives from 80 to 42. This improvement suggests that the fine-tuning process effectively enhanced the model's ability to identify

Table 3
Comparison of VGG16 at initial training and VGG16 at fine-tuning

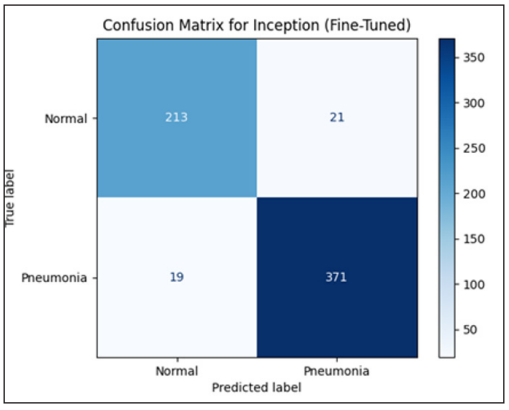
VGG16	VGG16 at Initial Training	VGG16 at Fine-Tuned
True Positives (TP)	386	373
True Negatives (TN)	154	192
False Positives (FP)	80	42
False Negatives (FN)	4	17



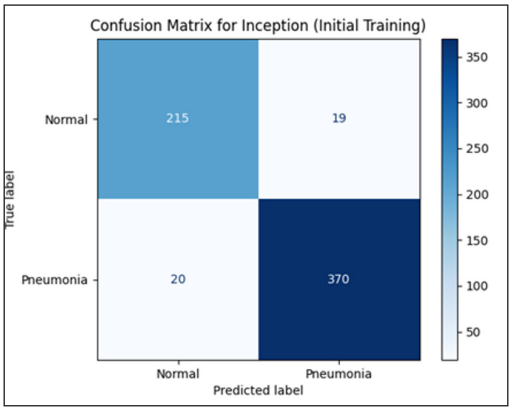
(a) VGG16 Fine-Tuned Model



(b) VGG16 Initial Training Model



(c) Inception Fine-Tuned Model



(d) Inception Initial Training Model

Figure 5. (a) to (d) Confusion matrices for four models

normal cases, thereby increasing its specificity correctly. However, this improvement in specificity comes at a cost: the number of true positives decreases from 386 to 373, while the number of false negatives increases from 4 to 17. This indicates that while the fine-tuning process improved the model's ability to classify normal cases correctly, it slightly compromised its ability to accurately identify pneumonia cases. The increase in false negatives is particularly concerning in medical diagnosis contexts, where missing a positive case can have serious implications. Comparing the two models, it is evident that the Inception model generally performs better than VGG16 in terms of balanced accuracy for both normal and pneumonia cases. The initial Inception training shows fewer misclassifications than VGG16, with fewer false positives and false negatives. Fine-tuning minimally impacts Inception's performance, maintaining its high accuracy, whereas VGG16

shows a significant shift in performance metrics post-finetuning, with improved specificity but reduced sensitivity.

ROC Analysis

Figure 6 displays the ROC curves for four different models evaluated in the study: VGG-16 at both initial and fine-tuned stages, and Inception (both initial and fine-tuned). The ROC curve is a graphical representation of a model's diagnostic performance, plotting the true positive rate against the false positive rate at a fixed specificity.

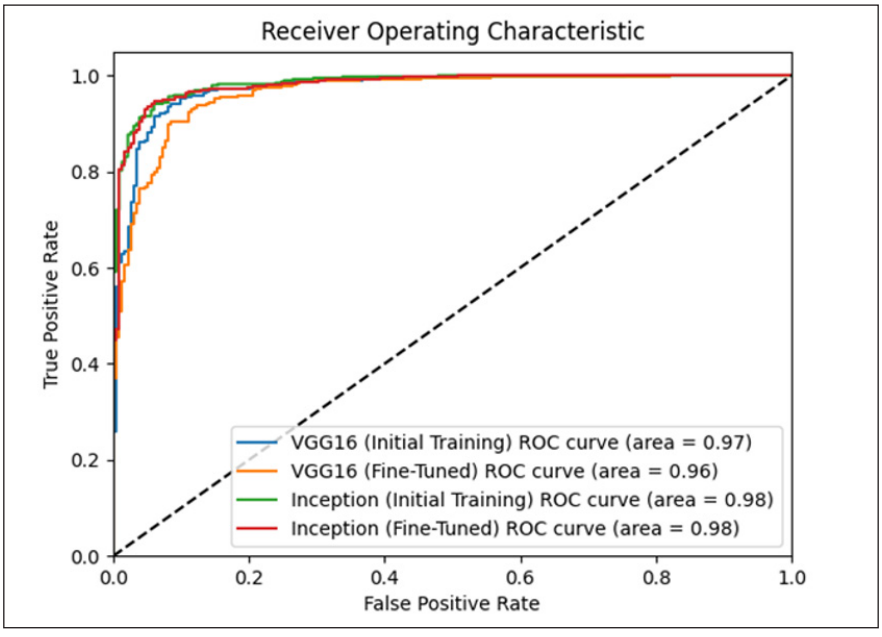


Figure 6. ROC curves for all models

The receiver operating characteristic (ROC) curves for two advanced deep learning models, VGG16 and Inception, are illustrated above after fine-tuning and regularisation. The ROC curve is a key tool for examining the performance of classification models as a function of the trade-off between true positive rate (sensitivity) and false positive rate. The analysis of these curves provides a deep insight into the model's efficiency in class discrimination. ROC curve was plotted with the x-axis representing the false positive rate and the y-axis representing the true positive rate. The false positive rate (FPR) is the ratio of false positives to the total number of positives, N/(false positives + true negatives). On the other hand, the true positive rate (TPR), also known as sensitivity or recall, computes the ratio of the true positives to the sum of true positives and false negatives. The diagonal line represents the performance of a random classifier, with an AUC of 0.50. For any

model that's better than random classification, its curve will be above this line. Analysis of the ROC curves provides valuable insights into the efficacy of the VGG16 and Inception architectures.

The initial training achieves an AUC of 0.97, with high sensitivity and specificity for VGG16. However, with fine-tuning, the AUC drops slightly to 0.96. This slight drop was due to overfitting, and there is a hint that fine-tuning on top of augmentation made the model aware of some repeated images. However, VGG16 remains an excellent model, with an AUC significantly higher than the baseline. On the contrary, the Inception model shows remarkable stability and robustness. With VGG16 as an example, Inception can initially obtain an AUC of 0.98. This noteworthy result demonstrates that the Inception architecture is well-suited for classification. It is also notable that, even after fine-tuning, Inception still achieves an AUC of 0.98, indicating that the model's performance remains very high and is not significantly affected by the fine-tuning. This stable result indicates that Inception is robust and suitable for the data with few performance degradations.

The inception, which is pretrained and fine-tuned, has higher AUC values, which is an obvious indication that it has an a priori advantage in the model architecture better suited for this classification. Although VGG16 performed well, the slight performance drop after fine-tuning suggests it may be more prone to overfitting or other fine-tuning issues than Inception. In summary, the ROC curves of VGG16 and Inception-based models demonstrate that they are effective classifiers with excellent performance. Being passable in terms of AUC, and equally withstanding fine-tuning, Inception would likely be a marginally better option than VGG16. These observations are useful for applications that require a robust classifier, and for which Inception’s robustness and slightly better accuracy may make it an attractive choice. Yet, VGG16 achieves good performance as well, indicating that both architectures have high potential in many classification tasks.

Average Precision

Figure 7 shows the Precision–Recall (PR) curves for the models. PR curves are especially useful for evaluating models on imbalanced datasets, as they focus on performance for the positive class (pneumonia) and can be more informative than ROC curves when one class is rare. The test partition used here is imbalanced at approximately 1.7:1 in favour of pneumonia, so PR analysis is directly applicable and complements the ROC results. Both models exhibit high precision across most of the recall range. For both VGG16 and Inception, precision remains near 1.0 when recall is moderate, and then precision declines as recall approaches 1.0. This is a typical pattern: to capture the very last few per cent of pneumonia cases, which pushes recall toward 100%, the models must tolerate more false positives, which lowers precision. The drop in precision at the extremely high-recall end indicates that some predictions made to capture the remaining pneumonia cases were normal

cases, like false positives. We computed the Average Precision (AP) for each PR curve, which summarises the precision-recall trade-off as a single number, which is essentially the area under the PR curve.

The VGG16 model achieved an AP of approximately 0.98 in its initial state, and this AP remained around 0.98 after fine-tuning. This consistency suggests that fine-tuning did not significantly change VGG16's precision-recall characteristics – the model was already performing near-optimally in terms of balancing precision and recall. The Inception-v3 model achieved a slightly higher AP, about 0.99, both before and after fine-tuning. This indicates that Inception had a marginal advantage in precision-recall performance, maintaining extremely high precision across all recall levels. The fact that both models' AP scores are very high and that neither model's score decreased after fine-tuning underscores that both the initial and fine-tuned models make very few mistakes in a precision-recall sense. For VGG16, an AP of 0.98 indicates that, on average, it achieves 98% precision across all recall levels, which translates to a strong result. Inceptions around 0.99 AP means it performed slightly better, which aligns with its higher precision and the lower false-positive rate observed in the confusion matrix. In summary, the precision–recall analysis reinforces the earlier findings: both models are highly effective at identifying pneumonia, with Inception being a touch better, and the fine-tuning stage preserved the models' excellent precision, recall balance which improves VGG16's recall slightly at the expense of some precision, as seen in the confusion matrix discussion, but overall AP remained very high.

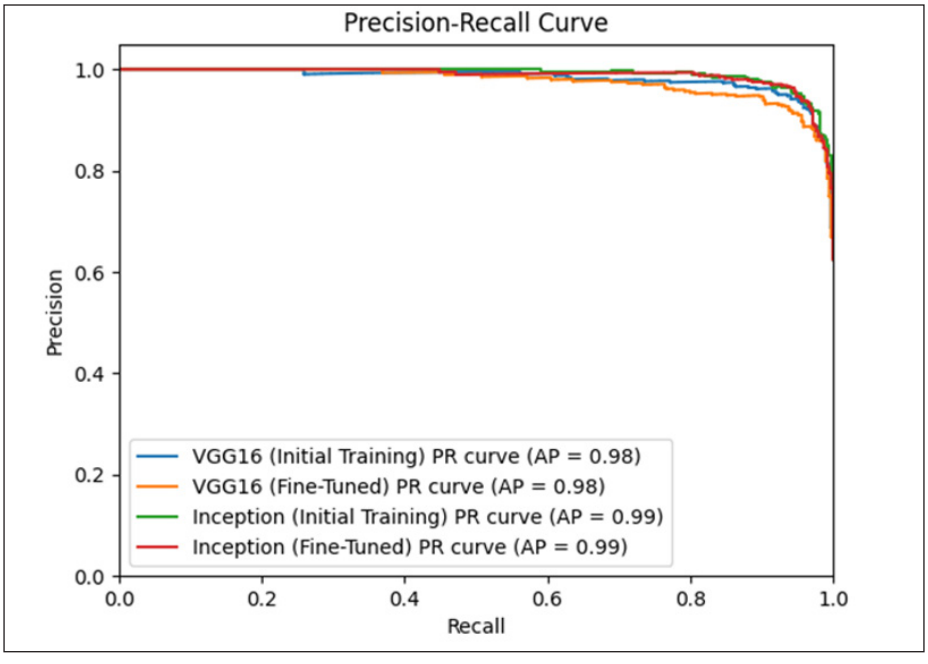


Figure 7. Precision-Recall (PR) curves for VGG16 and Inception deep learning models

F1 Score Analysis

We examined the F1-score for each model to further quantify the balance between precision and recall. The F1-score is the harmonic mean of precision and recall, providing a single metric that balances the two (with an F1 of 1 indicating perfect precision and recall, and 0 indicating the worst). For the VGG16 model, the initial training phase yielded an F1-score of approximately 0.90 on the test set. This indicates that VGG16 initially achieved a strong recall of almost 99% but a somewhat lower precision of around 83%, as reflected in its relatively low F1 score. After fine-tuning, VGG16's F1-score improved to about 0.93. This improvement is consistent with what we observed earlier: fine-tuning significantly increased VGG16's precision (fewer false positives) while only slightly reducing recall, resulting in a better overall balance and thus a higher F1-score.

The Inception-v3 model had a higher F1-score to start with, roughly 0.95 in the initial phase. Both high precision and high recall, over 95% for each metric. After fine-tuning, Inception's F1-score remained essentially the same, at about 0.95. The slight changes in precision and recall for Inception due to fine-tuning were negligible and cancelled out in terms of F1, which is one might expect at most a 0.001 drop, as seen, which is insignificant at this scale. This reiterates that fine-tuning did not have a notable impact on Inception's already well-optimised performance. Comparing the two models: initially, Inception's F1-score is around 0.95, which is higher than VGG16's, which is around 0.90, meaning Inception offered a better precision-recall trade-off without any adjustments. After fine-tuning, VGG16's F1-score rose to 0.93, closing much of the gap, while Inception stayed around 0.95. These values indicate that both models achieve an excellent balance between avoiding false positives and false negatives on the test set, with Inception maintaining a slight edge. Table 4 summarises the precision, recall, and F1-score for both models under initial and fine-tuned conditions, based on test-set outcomes. It provides a quick comparison of how fine-tuning shifted these key performance metrics.

Table 4
Comparative performance metrics for VGG16 and inception models: Initial vs. Fine-Tuned Training

	VGG16Initial Training	VGG16 Fine-Tuned Training	Inception Model Initial Training	Inception Model Fine-Tuned Training
Precision	0.8283	0.8988	0.9512	0.9464
Recall	0.9897	0.9564	0.9487	0.9513
F1 Score	0.9019	0.9267	0.9499	0.9488

Test-Set Performance and Statistical Analysis

The accuracy previously discussed in Figure 4 is for the training and validation phases and must not be read as an estimate of held-out performance. Table 5, therefore, reports

performance computed directly from the test-set confusion matrices of Figure 5, on the independent test partition of 624 images (234 normal, 390 pneumonia). Confidence intervals are Wilson score intervals at the 95% level, which are preferred to the normal approximation for proportions close to unity.

Table 5
Test-Set Performance for All Models (n = 624; 234 normals, 390 pneumonia).

Model	Accuracy	95% CI	Sensitivity	Specificity	F1	MCC
VGG16 (initial)	0.8654	0.8364-0.8899	0.9897	0.6581	0.9019	0.7213
VGG16 (fine-tuned)	0.9054	0.8799-0.9260	0.9564	0.8205	0.9267	0.7969
Inception-v3 (initial)	0.9375	0.9157-0.9539	0.9487	0.9188	0.9499	0.8668
Inception-v3 (fine-tuned)	0.9359	0.9139-0.9526	0.9513	0.9103	0.9488	0.8630

Fine-tuning altered VGG16 substantially, raising test accuracy from 0.8654 to 0.9054. The mechanism is a redistribution of error rather than a uniform improvement: specificity rose from 0.6581 to 0.8205 while sensitivity fell from 0.9897 to 0.9564, and the number of false negatives increased from 4 to 17. In a screening application, where a missed case carries a greater cost than a false alarm, this is a consequential trade rather than an unambiguous gain. For Inception-v3, the effect of fine-tuning was negligible: accuracy moved from 0.9375 to 0.9359 and the Matthews correlation coefficient from 0.8668 to 0.8630, both differences lying well within the confidence intervals. Inception-v3 is therefore best characterised as already converged after the feature-extraction phase alone, and the earlier statement that fine-tuning boosted its performance is not supported by the test-set data.

Comparing the two fine-tuned architectures, Inception-v3 classified 584 of 624 test images correctly, compared with 565 for VGG16, a difference of 19 images, or 3.0 percentage points. Under a two-proportion z-test, this corresponds to $z = 1.99$ and $p = 0.047$. Because both models were evaluated on the same images, their errors are positively correlated, so this unpaired test overestimates the variance of the difference. Consequently, the reported p-value is a conservative upper bound, and a paired analysis of the per-image predictions would be expected to yield a smaller value. The magnitude of the difference is nonetheless small: the 95% confidence intervals of the two fine-tuned models overlap across the range 0.914 to 0.926, and their Matthews correlation coefficients differ by 0.066. The advantage of Inception-v3 is accordingly described here as modest rather than decisive, and the case for preferring it rests as much on its lower computational cost, reported in Section 4.5, as on its classification performance.

Prediction Duration

Apart from accuracy, an important consideration for deploying deep learning models is their computational efficiency, particularly the time required to make predictions or to train, which indirectly affects how quickly updates can be made. Figure 8 presents the time required by each model to make predictions on 624 X-ray images, which represent the size of our test set, as well as the models' training times per epoch. The VGG16 model, known for its deeper architecture and larger number of parameters, exhibited higher computational demands. In the initial training phase, VGG16 required approximately 94 seconds to complete 20 epochs, averaging roughly 4.7 seconds per epoch. Upon fine-tuning, VGG16's computational load increased: it took around 110 seconds to complete 20 epochs, which takes about 5.5 seconds per epoch. This increase is expected because fine-tuning involves unfreezing more layers of the network, resulting in more computations per epoch. In terms of inference time, processing the entire test set would scale similarly with VGG16, which is relatively slow due to its complexity. In contrast, the Inception-v3 model is architecturally optimised for efficiency. During initial training, Inception completed 20 epochs in approximately 60 seconds, which is about 3.0 seconds per epoch, substantially faster than VGG16. With fine-tuning, Inception's time per epoch increased only slightly, taking about 66 seconds for 20 epochs, which takes about 3.3 seconds per epoch. This modest increase reflects the additional computations required for fine-tuning some layers; however, overall, Inception remained significantly faster than VGG16. Consequently, Inception can process a batch of 624 test images in significantly less time than VGG16. The comparison clearly shows that Inception-v3 is more computationally efficient than VGG16. VGG16's deeper network and larger fully connected layers contribute to its longer training and inference times. In a real-world deployment scenario, this means that a system using Inception could handle more images per unit time or provide results to clinicians more quickly than one using VGG16, especially on hardware with limited processing power. On the other hand, VGG16's improved accuracy after fine-tuning might be beneficial in scenarios where maximum accuracy is needed, and the infrastructure can support the heavier model. In summary, if computational resources or time are constrained, for instance, deploying an automated pneumonia detection system in a busy clinic or on edge devices such as mobile X-ray units, the Inception-v3 model is the preferable choice due to its shorter prediction times. VGG16 remains a viable model for scenarios where slightly longer processing times are acceptable in exchange for high accuracy (especially after fine-tuning). Figure 8 summarises the observed prediction durations for each model.

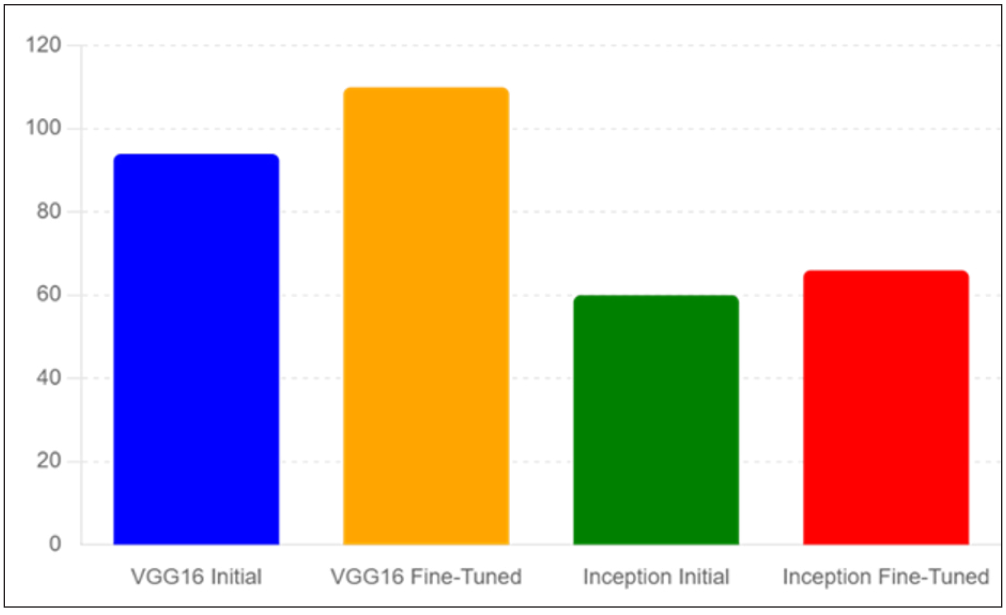


Figure 8. Prediction time duration for each model

DISCUSSION

Interpretation of the Corrected Results

The corrected test-set figures are approximately six to nine percentage points below the accuracies commonly reported for these architectures on this benchmark. The discrepancy is not attributable to the training procedure, which follows established practice, but to the reporting convention. Final training-phase accuracy exceeds 0.99 for both fine-tuned models, whereas held-out accuracy does not exceed 0.936; where a study reports the former without distinguishing it from the latter, the resulting figure overstates deployable performance by roughly the margin observed here. The same mechanism affects validation-phase reporting even more severely: a validation accuracy of 100% on the distributed 16-image partition is consistent with a true held-out accuracy of 0.905, because the partition is too small to resolve a difference that large. This is not a defect specific to the present experiments but rather a property of the benchmark distribution that is inherited by any study that adopts it without modification.

Comparison with Prior Work

Against the hybrid CNN–XGBoost model of Hedhoud et al. (2023), which reported 87% accuracy with fast inference, the fine-tuned Inception-v3 model evaluated here achieves 93.6% accuracy on a held-out partition of comparable size; however, the two studies use

different partitions, so the comparison is indicative only. More generally, the corrected figures sit at the lower end of the published range for this dataset, which is the expected consequence of reporting held-out rather than training-phase performance. Direct numerical comparison with studies that do not state which partition their headline accuracy was computed on is not meaningful, and this irreproducibility is itself one of the findings of the present work.

The Limits of Discrimination Metrics as Evidence of Clinical Validity

A high AUC is not by itself sufficient evidence of clinical validity. A classifier may achieve strong discrimination using image features that correlate with the class label but carry no diagnostic meaning, such as differences in patient positioning, exposure settings, annotation markers, or scapular and diaphragmatic borders that systematically differ between the two classes in a single-source collection. Establishing that a model attends to pulmonary findings requires spatial evidence, most obtained through class-activation mapping; recent work in medical imaging uses Grad-CAM for precisely this purpose, demonstrating that activation concentrates on lesion-relevant rather than background regions (Tülü & Kaya, 2026). No such analysis was performed in the present study; therefore, the results reported here establish discrimination performance without demonstrating that the discrimination is anatomically grounded. This is a material constraint on the conclusions that can be drawn: the accuracy figures in Table 5 quantify how often each model is correct on this test partition, but they do not license any claim about why it is correct, nor about how the models would behave on radiographs acquired under different conditions. Both architectures are accordingly presented as benchmark results rather than as candidates for clinical deployment, and interpretability analysis is identified in Section 5.5 as the first requirement for any subsequent work.

Six limitations constrain the interpretation of these results. First, the validation partition distributed with this benchmark contains only sixteen images and cannot support reliable model selection or early stopping; validation-phase as in Figure 4 should accordingly be read as descriptive of the training process rather than as estimates of generalisation. Second, the data derive from a single institution and a single paediatric population, so performance on adult radiographs, on different acquisition hardware, or under different exposure protocols is unknown, and no external validation was performed. Third, no patient-level metadata accompany the images, which prevents subgroup analysis by age, sex or comorbidity and makes it impossible to verify that images are independent at the patient level. Fourth, the decision threshold was fixed at 0.5 throughout rather than optimised against a clinically specified minimum sensitivity, which is the operating point a screening application would require. Fifth, comparisons between models were assessed using unpaired two-proportion tests and non-overlapping confidence intervals rather than paired tests on

the per-image predictions, such as McNemar's test or DeLong's test for correlated ROC curves. The unpaired approach is conservative in this setting, so the reported differences are not overstated, but a paired analysis would give a more precise estimate of the difference between architectures. Sixth, no interpretability or class-activation analysis was performed, so there is no evidence that either model bases its predictions on pulmonary findings rather than on incidental image characteristics; this is the single largest obstacle to interpreting the reported accuracies as clinically meaningful.

Future Work

Four directions follow directly from these limitations. The first and most immediate is class-activation analysis of the trained models using Grad-CAM or a comparable method, applied to a stratified sample of images correctly and incorrectly classified for both classes, to establish whether the decision evidence is anatomically plausible. Until this is done, the accuracy figures reported here cannot support any claim of diagnostic validity. Second, threshold selection should be driven by a specified minimum sensitivity, with specificity and the resulting review burden reported at that operating point, rather than by the default value of 0.5. Third, external validation in an independent cohort is a precondition for any claim of clinical utility, since the present results characterise generalisation only within a single collection. Fourth, constraining the classifier to anatomically relevant regions via lung-field segmentation as a preprocessing step would directly test whether the performance reported here depends on extrapulmonary image content and could be evaluated by re-running the present protocol on masked inputs. Ensemble combination of the two architectures, model compression for deployment on lower-end hardware, and extension to the multi-class setting distinguishing bacterial from viral pneumonia or screening for tuberculosis and COVID-19 remain worthwhile but secondary to these four.

CONCLUSION

This study reports a comparative evaluation of VGG16 and Inception-v3 for pneumonia classification on chest radiographs, conducted and reported entirely on held-out data. The fine-tuned Inception-v3 model achieved a test accuracy of 0.936 (95% CI 0.914–0.953), compared with 0.905 (0.880–0.926) for the fine-tuned VGG16 model, and did so at approximately 60% of the training time per epoch, making it the preferable choice when computational resources or response time are constrained. Fine-tuning benefited VGG16 by raising specificity from 0.658 to 0.821 at the cost of a reduction in sensitivity from 0.990 to 0.956 and produced no measurable change in Inception-v3. These held-out figures are materially below the near-perfect accuracies frequently reported for the same architectures on the same benchmark; the difference is attributable to the reporting of training-phase rather than test-phase performance, compounded by a distributed validation partition too

small to support model selection. Neither architecture is therefore ready for deployment as a screening aid without external validation, calibration of the decision threshold against a clinical sensitivity requirement, and verification that classification is driven by pulmonary findings. The contribution of this work lies in establishing a correctly reported reference point on a benchmark where published performance has drifted above what the underlying data support.

ACKNOWLEDGEMENT

Authors acknowledge the Universiti Teknologi MARA in collaboration with Universitas Mataram, Fakultas Teknik, Program Studi Teknik Informatika, Indonesia, under direct grant no 600-TNCPI/PBT 5/3 INT (064/2024).

REFERENCES

Abd Rahman, N. N., Yusoff, M., & Kassim, M. (2022). HOG and Haralick feature extractions with machine learning methods for COVID-19 X-ray image classification. *International Journal of Engineering Trends and Technology, 70*(10), 8-17. https://doi.org/10.14445/22315381/IJETT-V70I10P202

Abdulahi, A. T., Ogundokun, R. O., Adenike, A. R., Shah, M. A., & Ahmed, Y. K. (2024). PulmoNet: A novel deep learning based pulmonary diseases detection model. *BMC Medical Imaging, 24*(1), Article 51. https://doi.org/10.1186/s12880-024-01227-2

Aljuaid, H., Adlan, H., Alkebsi, B., Alfurhood, B. S., Liotta, A., & Cavallaro, L. (2026). An experimental comparison of deep learning models for pneumonia classification from chest X-ray images. *Biomedical Signal Processing and Control, 112*, 108742. https://doi.org/10.1016/j.bspc.2025.108742

Azis, S. F. S., Shahbudin, S., Kassim, M., Mohamad, R., & Rahman, F. Y. A. (2022). Photovoltaic module defects classification analysis using DenseNet architecture. In *2022 IEEE Symposium on Industrial Electronics & Applications (ISIEA)* (pp. 1-5). IEEE. https://doi.org/10.1109/ISIEA54517.2022.9873681

Basubrin, O. (2025). Current status and future of artificial intelligence in medicine. *Cureus, 17*(1), Article e77561. https://doi.org/10.7759/cureus.77561

Boniol, M., Kunjumen, T., Nair, T. S., Siyam, A., Campbell, J., & Diallo, K. (2022). The global health workforce stock and distribution in 2020 and 2030: A threat to equity and 'universal' health coverage? *BMJ Global Health, 7*(6), e009316. https://doi.org/10.1136/bmjgh-2022-009316

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145-1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Davis, J., & Goadrich, M. (2006). The relationship between precision–recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233-240). Association for Computing Machinery. https://doi.org/10.1145/1143844.1143874

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

El Houby, E. M. F. (2024). COVID-19 detection from chest X-ray images using transfer learning. *Scientific Reports, 14*(1), Article 11639. https://doi.org/10.1038/s41598-024-61693-0

Gulshan, V., Rajan, R. P., Widner, K., Wu, D., Wubbels, P., Rhodes, T., Whitehouse, K., Coram, M., Corrado, G., Ramasamy, K., Raman, R., Peng, L., & Webster, D. R. (2019). Performance of a deep-learning algorithm vs manual grading for detecting diabetic retinopathy in India. *JAMA Ophthalmology, 137*(9), 987-993. https://doi.org/10.1001/jamaophthalmol.2019.2004

Hedhoud, Y., Mekhaznia, T., & Amroune, M. (2023). An improvement of the CNN-XGboost model for pneumonia disease classification. *Polish Journal of Radiology, 88*, e483-e493. https://doi.org/10.5114/pjr.2023.132533

Hu, Q., Jiang, J., Xu, H., & Kassim, M. (2024). IMNE: Maximizing influence through deep learning-based node embedding in social network. *Swarm and Evolutionary Computation, 88*, Article 101609. https://doi.org/10.1016/j.swevo.2024.101609

JTIPTJ. (2021). *Chest X-ray (pneumonia, COVID-19, tuberculosis)* [Data set]. Kaggle. https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid19tuberculosis

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., . . . Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell, 172*(5), 1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1412.6980

Kunjumon, A., V, S., & Nair, J. J. (2026). A hybrid deep learning framework for improved pneumonia detection in chest X-ray images. *Procedia Computer Science, 282*, 1043-1055. https://doi.org/10.1016/j.procs.2026.05.146

Lim, W. S. (2022). Pneumonia—overview. In S. M. Janes (Ed.), *Encyclopedia of respiratory medicine* (2nd ed., pp. 185–197). Academic Press. https://doi.org/10.1016/B978-0-12-801238-3.11636-8

Mahalle, V. S., Kandoi, N. M., & Patil, S. B. (2024). Transfer learning by fine-tuning pre-trained convolutional neural network architectures for image recognition. In D. Mishra, X. S. Yang, A. Unal, & D. S. Jat (Eds.), *Data science and big data analytics* (pp. 273-287). Springer. https://doi.org/10.1007/978-981-99-9179-2_21

Müller, D., Soto-Rey, I., & Kramer, F. (2022). Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes, 15*(1), Article 210. https://doi.org/10.1186/s13104-022-06096-y

Namdar, K., Haider, M. A., & Khalvati, F. (2021). A modified AUC for training convolutional neural networks: Taking confidence into account. *Frontiers in Artificial Intelligence, 4*, Article 582928. https://doi.org/10.3389/frai.2021.582928

Prechelt, L. (1998). Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks, 11*(4), 761–767. https://doi.org/10.1016/S0893-6080(98)00010-0

Raghu, M., Zhang, C., Kleinberg, J., & Bengio, S. (2019). *Transfusion: Understanding transfer learning with applications to medical imaging* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1902.07208

Sharma, V., Sandhu, R., Kaur, J., Kaur, H., Chouhan, S., & Rakhra, M. (2024). A novel CNN-based approach for pneumonia detection using Keras and TensorFlow. In 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT). IEEE. https://doi.org/10.1109/ICEECT61758.2024.10739031

Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging, 35*(5), 1285-1298. https://doi.org/10.1109/TMI.2016.2528162

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), Article 60. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1409.1556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(56), 1929–1958. https://jmlr.org/papers/v15/srivastava14a.html

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2818-2826). IEEE. https://doi.org/10.1109/CVPR.2016.308

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning* (Vol. 97, pp. 6105-6114). PMLR. https://proceedings.mlr.press/v97/tan19a.html

Tülü, Ç. N., & Kaya, Y. (2026). Convolutional Fourier analysis network (CONV-FAN-POX): A novel time–frequency approach for medical image analysis. *Biomedical Signal Processing and Control, 117*, 109698. https://doi.org/10.1016/j.bspc.2026.109698

UNICEF, United Nations Children's Fund. (2024). *Pneumonia in children statistics.* https://data.unicef.org/topic/child-health/pneumonia/

WHO, World Health Organization. (2025). *Pneumonia.* https://www.who.int/health-topics/pneumonia/

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging, 9*(4), 611-629. https://doi.org/10.1007/s13244-018-0639-9