# Hyperparameter Optimized DenseNet201 for Chest X-Ray Pneumonia and Tuberculosis Classification with Grad-CAM Visualization

**Damianus Christopher Samosir[1*], Ernawati[2], Desi Andreswari[3]**

[1,2,3]Informatics Engineering, Faculty of Engineering, Universitas Bengkulu, Bengkulu, Indonesia
E-mail: [1]*christophersamosir@gmail.com, [2]ernawati@unib.ac.id, [3]desi.andreswari@unib.ac.id

(Received: 28 Jul 2026, revised: 18 Aug 2026, accepted: 26 Aug 2026)

**Abstract**

Pneumonia and tuberculosis (TB) remain major global health challenges, particularly in developing countries. Chest X-ray imaging is the most commonly used initial diagnostic modality, yet visual similarity between these diseases and normal lung conditions frequently causes diagnostic ambiguity. Deep learning, particularly through transfer learning with DenseNet201, offers promising potential for automated multi-class classification; however, numerically high accuracy alone does not guarantee training stability, generalization robustness, or clinical reliability. This study investigates hyperparameter optimization of DenseNet201 transfer learning for three-class chest X-ray classification (Normal, Pneumonia, and Tuberculosis), systematically evaluating the interaction between learning rates (LR) ($1 \times 10^{-5}$, $5 \times 10^{-6}$, $1 \times 10^{-6}$), batch sizes (BS) (8, 16, 32), and fine-tuning strategies (full fine-tuning, partial freeze 10%, and partial freeze 20%). Training was conducted across two phases, namely feature extraction and fine-tuning, yielding 27 experimental scenarios. The optimal configuration, consisting of full fine-tuning, batch size 8, and learning rate $1 \times 10^{-5}$, achieved a test accuracy of 98.06%, a macro F1-score of 0.981, and a macro ROC-AUC of 0.999. Despite these results, an error analysis revealed that 27 of 1,392 test images were misclassified, with 19 disease cases incorrectly predicted as Normal, highlighting that distinguishing subtle pathological patterns from Normal chest X-rays remains a key challenge. Furthermore, Grad-CAM visualization showed class-specific activation patterns, including central activation for Normal, diffuse bilateral activation for Pneumonia, and apical activation for Tuberculosis, providing visual insights into the image regions contributing to the model's classification decisions.

**Keywords:** DenseNet201, Transfer Learning, Pneumonia, Tuberculosis, Grad-CAM

## I. INTRODUCTION

Pneumonia and tuberculosis (TB) are major global health challenges, particularly in developing countries, with high morbidity and mortality rates [1]. Pneumonia causes approximately 2.5 million deaths worldwide each year, while the World Health Organization reported 10.8 million new TB cases and 1.25 million TB-related deaths globally in 2023 [2], [3]. Indonesia is among the countries with the highest TB burden, with an estimated 1.09 million cases and 125,000 deaths annually [4]. In TB-endemic developing countries, radiographic findings may resemble other pulmonary conditions, making diagnosis based on imaging alone challenging [1], [5], [6].

Chest X-ray (CXR) is widely used for the initial evaluation of pulmonary diseases, but its interpretation depends on radiologists and suffers from up to 22% inter-observer disagreement [7]. AI-assisted approaches have shown potential to reduce radiograph reading time [8]. Moreover, the visual overlap among pneumonia, tuberculosis, and normal lung patterns—including infiltrates, consolidations, and opacities complicates consistent classification [1], [9]. These challenges highlight the need for automated approaches capable of supporting consistent pulmonary disease classification from CXR images.

Deep learning, particularly convolutional neural networks (CNNs), has demonstrated strong performance in CXR classification. Dardouri et al.[10] achieved 97.24% accuracy in classifying Normal, COVID-19, and Pneumonia. Furthermore, Jha et al.[11] demonstrated the potential of pretrained representations, while Davila et al.[12] showed that fine-tuning effectiveness varies across architectures and medical imaging domains. These findings highlight the influence of pretrained representations and adaptation strategies on model performance.

DenseNet201 was selected because its densely connected architecture promotes feature reuse and efficient information propagation, facilitating detailed pattern extraction from

DOI: 10.34148/teknika.v15i3.1533

medical images [13], [14]. Previous studies have demonstrated the potential of deep learning for multi-class CXR classification; for example, Malik et al.[15] achieved 99.39% accuracy using CDC_Net. Despite this strong potential, differing architectures, datasets, and training strategies complicate direct comparisons [12]. More importantly, the combined effects of learning rate, batch size, and fine-tuning strategy on DenseNet201 for three-class CXR classification remain insufficiently evaluated.

A systematic evaluation of these training factors is therefore needed to determine whether high classification performance is accompanied by stable training, consistent convergence, and reliable generalization [16], [17]. Evaluating these factors jointly provides an empirical basis for selecting an optimal DenseNet201 configuration beyond conventional training settings [12], [13].

Furthermore, the black-box nature of deep learning can limit the interpretability of model predictions [18], [19]. Grad-CAM and other XAI (Explainable AI) methods provide visual explanations by highlighting image regions that contribute to predictions [19], [20]. Here, Grad-CAM is applied to visualize regions influencing model predictions to assess their consistency with clinically relevant pulmonary [9], [19].

Accordingly, this study makes three contributions: (1) systematic evaluation of 27 combinations of learning rate, batch size, and fine-tuning strategy for DenseNet201; (2) joint analysis of classification performance, training stability, convergence behavior, and generalization to identify an empirically supported optimal configuration; and (3) Grad-CAM-based visualization of the selected model to assess the regions contributing to its classification decisions.

## II. METHODS

### A. Research Framework

This study follows the five-stage research framework shown in Figure 1: problem identification and data collection, data distribution analysis, preprocessing, two-phase model training (feature extraction and fine-tuning), and evaluation with Grad-CAM interpretability.

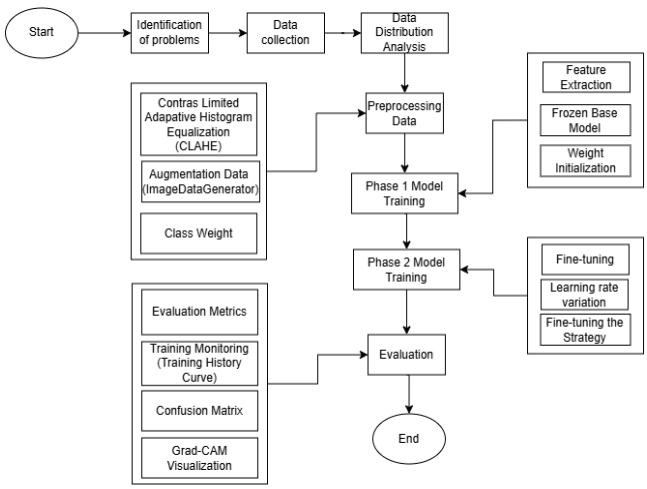


Figure 1. Research Process

### B. Dataset

The dataset used was the Combined Unknown Pneumonia and Tuberculosis dataset from Kaggle. Three classes were selected: Normal, Pneumonia, and Tuberculosis, while the Unknown class was excluded. The dataset was pre-split into training, validation, and testing subsets, as shown in Table 1.

Table 1. Dataset Distribution by Class and Subset

| Class | Train | Validation | Testing |
|---|---|---|---|
| Normal | 4,667 | 274 | 548 |
| Pneumonia | 3,633 | 213 | 427 |
| Tuberculosis | 3,573 | 207 | 417 |
| Total | 11,873 | 694 | 1,392 |

### C. Preprocessing

#### 1. CLAHE (Contrast Limited Adaptive Histogram Equalization)

Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to enhance local contrast in the chest X-ray images using a clip limit of 1.5 and a 6×6 tile grid. The same configuration was applied consistently to the training, validation, and test sets, as summarized in Table 2.

Table 2. CLAHE Configuration

| Parameter | Value |
|---|---|
| Clip Limit | 1.5 |
| Tile Grid Size | 6 × 6 |
| Applied To | Training, Validation and Test Set |

#### 2. Data Augmentation

Data augmentation was applied exclusively to the training set using rotation (±15°), width and height shifts (10%), zoom (15%), and reflect fill mode to increase sample variation while preserving realistic chest X-ray characteristics. No augmentation was applied to the validation and test sets, as summarized in Table 3.

Table 3. Data Augmentation Configuration

| Parameter | Value |
|---|---|
| Rotation Range | ±15° |
| Width Shift Range | 10% |
| Height Shift Range | 10% |
| Zoom Range | 15% |
| Fill Mode | Reflect |
| Applied To | Training set only |

#### 3. Class Weight

Class weighting was incorporated to mitigate the class imbalance observed in the training set. Balanced weights were calculated using compute_class_weight, with the resulting values presented in Table 4.

Table 4. Class Weight Comparison

| Class | Training Samples | Class Weight |
|---|---|---|
| Normal | 4,667 | 0.848 |
| Pneumonia | 3,633 | 1.089 |
| Tuberculosis | 3,573 | 1.108 |

### D. Model Architecture

The model used in this study is DenseNet201, employing ImageNet pretrained weights as the backbone feature extractor. The original chest X-ray images had varying resolutions because they were compiled from multiple sources. To match the input specification of DenseNet201, all images were resized to 224×224×3 pixels. Although this standardization is required for the pretrained architecture, resizing may reduce fine spatial information and potentially affect the representation of small, localized pathological features. DenseNet201 employs a dense connectivity mechanism in which each layer receives inputs from all preceding layers, enhancing gradient flow and feature reuse, which is beneficial for detecting subtle pathological patterns [21]. To adapt the model to the three-class task, a custom classification head was appended, consisting of Global Average Pooling, Batch Normalization, a Dense layer of 512 neurons, Dropout of 0.3, and L2 regularization of 0.005, forming feature representations relevant to the classification task while reducing overfitting risk. The output layer uses softmax to produce class probabilities. The configuration is summarized in Table 5.

Table 5. Model Architecture Configuration

| Component | Configuration |
|---|---|
| Backbone Model | DenseNet201 |
| Pretrained Weight | ImageNet |
| Input Size | 224×224×3 |
| Pooling Layer | Global Average Pooling |
| Dense Layer | 512 neurons |
| Batch Normalization | Yes |
| Regularization | Dropout (0.3) and L2 Regularization (0.005) |
| Output Activation | Softmax |
| Number of Classes | 3 |

### E. Training Strategy

Training was carried out in two main phases, feature extraction and fine-tuning, to maintain learning stability while improving the model's ability to recognize the specific characteristics of the chest X-ray images. The Phase 2 experimental configuration is described below. The fine-tuning phase was controlled through three hyperparameters: fine-tuning strategy, learning rate (LR), and batch size (BS) combined into 27 experimental scenarios (Table 6). Fine-tuning (FT) strategy (Full Fine-Tuning, Partial Fine-tuning Freeze 10% Early Layers, and Partial Fine-Tuning Freeze Early Layers 20%) was varied to evaluate how the degree of backbone adaptation affects classification performance. Learning rate ($1\times10^{-5}$, $5\times10^{-6}$, $1\times10^{-6}$) was kept within a narrow range because the backbone already carries pretrained weights: too large a rate risks catastrophic forgetting of previously learned representations, while too small a rate slows adaptation; $1\times10^{-5}$ served as the common upper bound for fine-tuning medical-image transfer learning models, $1\times10^{-6}$ as a conservative lower bound, and $5\times10^{-6}$ as the intermediate value. Batch size (8, 16, 32) represented small, medium, and relatively large batches, reflecting the trade-off between gradient-estimate quality and training stability: smaller batches often generalize better but train less stably, while larger batches train faster and more stably but may generalize less well.

Table 6. Experimental Configuration Phase 2 Fine-Tuning

| Scenario | LR | Strategy | BS |
|---|---|---|---|
| S1 | $1\times10^{-5}$ | Full Fine-Tuning | 8, 16, 32 |
| S2 | $1\times10^{-5}$ | Partial Fine-Tuning 10% | 8, 16, 32 |
| S3 | $1\times10^{-5}$ | Partial Fine-Tuning 20% | 8, 16, 32 |
| S4 | $5\times10^{-6}$ | Full Fine-tuning | 8, 16, 32 |
| S5 | $5\times10^{-6}$ | Partial Fine-Tuning 10% | 8, 16, 32 |
| S6 | $5\times10^{-6}$ | Partial Fine-Tuning 20% | 8, 16, 32 |
| S7 | $1\times10^{-6}$ | Full Fine-Tuning | 8, 16, 32 |
| S8 | $1\times10^{-6}$ | Partial Fine-Tuning 10% | 8, 16, 32 |
| S9 | $1\times10^{-6}$ | Partial Fine-Tuning 20% | 8, 16, 32 |

Beyond these experimental variables, training relied on a common mechanism applied across all scenarios. The Adam optimizer was chosen for its ability to adjust the learning rate adaptively, while categorical crossentropy was used as the loss function since this is a multi-class classification problem. Phase 1 (feature extraction) trained only the classification head for 15 epochs at a fixed learning rate of $1\times10^{-3}$, while Phase 2 (fine-tuning) continued training for up to 70 epochs at the scenario-specific learning rate listed in Table 6. Three callbacks kept training stable: EarlyStopping (patience=7, monitor=val_loss) halts training once validation performance stops improving; ModelCheckpoint saves the best weights; and ReduceLROnPlateau (factor=0.3, patience=5) lowers the learning rate when optimization stagnates. This configuration is summarized in Table 7.

Table 7. General Training Configuration

| Parameter | Configuration |
|---|---|
| Optimizer | Adam |
| Loss Function | Categorical Crossentropy |
| Epoch (Phase 1) | 15 |
| Epoch (Phase 2) | Up to 70 (early stopping) |
| Learning rate (Phase 1 – Feature Extraction) | $1 \times 10^{-3}$ |
| Learning rate (Phase 2 – Fine-Tuning) | variable per scenario ($1\times10^{-5}$; $5\times10^{-6}$; $1\times10^{-6}$) |
| Callbacks | EarlyStopping (patience=7, monitor=val_loss); ModelCheckpoint (best val_loss); ReduceLROnPlateau (factor=0.3, patience=5, min_LR=$1\times10^{-7}$) |

### F. Evaluation Metrics

Model performance was evaluated using accuracy, precision, recall, F1-score, confusion matrix, ROC-AUC (one-vs-rest), and Precision-Recall curves. Training stability was assessed using training and validation loss and accuracy curves. Grad-CAM was applied to the final convolutional

layer, conv5_block32_concat, to visualize class-discriminative activation patterns in relation to pulmonary anatomical regions.

## III. RESULTS AND DISCUSSION

### 1. Effect of Fine-Tuning Strategy

### A. Quantitative Performance Across Fine-Tuning Strategies

Table 8. Quantitative Performance Comparison of Fine-Tuning Strategies (LR=1×10⁻⁵, BS=16)

| Strategy | Val. Acc | Val. Loss | Test Acc | Test Loss | Recall | Precision | F1-Score |
|---|---|---|---|---|---|---|---|
| Full Fine-Tuning (FT) | 0.9856 | 0.1353 | 0.9727 | 0.1605 | 0.9717 | 0.9757 | 0.9735 |
| Freeze 10% Early Layers | 0.9784 | 0.1489 | 0.9691 | 0.1867 | 0.9670 | 0.9739 | 0.9699 |
| Freeze 20% Early Layers | 0.9784 | 0.1434 | 0.9727 | 0.1687 | 0.9708 | 0.9766 | 0.9733 |

As shown in Table 8, Full Fine-Tuning achieved the best validation performance, with a validation accuracy of 98.56% and a validation loss of 0.1353. On the test set, Full Fine-Tuning and Freeze 20% achieved the same accuracy of 97.27%; however, Full Fine-Tuning produced a lower test loss (0.1605 vs. 0.1687) and the highest F1-score (97.35%). Meanwhile, Freeze 10% yielded the lowest performance across the evaluated metrics. The superior performance of Full Fine-Tuning can be attributed to its greater flexibility to update the pretrained DenseNet201 parameters, allowing the learned feature representations to adapt more extensively to the target chest X-ray domain. This is consistent with previous research showing that excessive freezing of pretrained layers may underutilize the representational capacity of the backbone, particularly for higher-level features that require adaptation to disease-specific patterns in chest X-ray images. Thus, in this study, updating the entire backbone provided greater capacity for learning task-specific representations, contributing to the superior validation performance and F1-score of Full Fine-Tuning [22].

### B. Training Stability Analysis

Training stability was assessed through training history curves (training loss, validation loss, training accuracy, validation accuracy) for each epoch. Each strategy comprised two training phases: feature extraction (epochs 1–15) and fine-tuning (epoch 16 onward).

For the three fine-tuning strategies (LR = 1 × 10⁻⁵, BS = 16), the resulting training histories are presented in Figures 2–4. All strategies showed consistent convergence without sustained increases in validation loss. Freeze 20% exhibited the smallest validation fluctuations, indicating the most stable convergence, whereas Full Fine-Tuning showed greater adaptation after the Phase 2 transition. This observation is consistent with previous research indicating that freezing parameters can constrain optimization and promote more stable convergence [22]. However, greater stability did not necessarily yield better classification performance, as the results in Table 8 demonstrate [23].

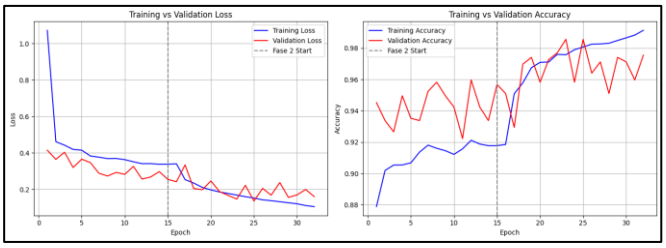


Figure 2. Training History Curve Full Fine-Tuning, LR=1×10⁻⁵

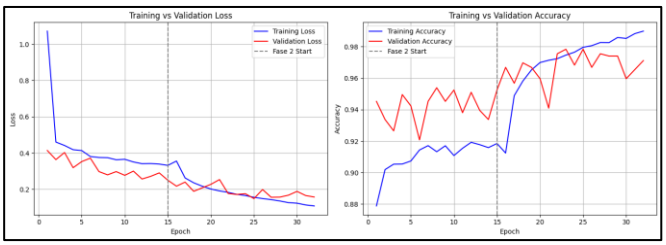


Figure 3. Training History Curve Freeze 10% Early Layers, LR=1×10⁻⁵

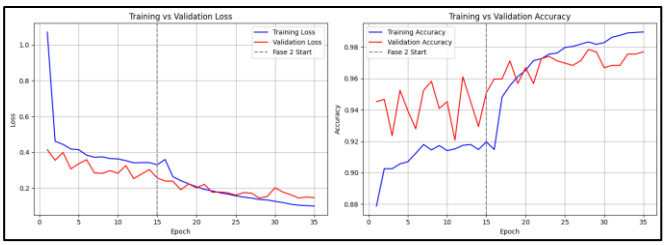


Figure 4. Training History Curve Freeze 20% Early Layers, LR=1×10⁻⁵

### C. Generalization Analysis

Table 9. Generalization Gap Analysis per Fine-Tuning Strategy (LR=1×10⁻⁵, BS=16)

| Strategy | Val Acc | Test Acc | Gap Acc | Val Loss | Test Loss | Gap Loss |
|---|---|---|---|---|---|---|
| Full FT | 0.9856 | 0.9727 | 0.0129 | 0.1353 | 0.1605 | 0.0252 |
| Freeze 10% | 0.9784 | 0.9691 | 0.0093 | 0.1489 | 0.1867 | 0.0378 |
| Freeze 20% | 0.9784 | 0.9727 | 0.0057 | 0.1434 | 0.1687 | 0.0253 |

Table 9 shows that all strategies generalize well, with small validation-to-test gaps. Full Fine-Tuning had the largest gap accuracy (1.29%) but the lowest test loss; Freeze 20% achieved the smallest gap (0.57%) with nearly identical test loss; Freeze 10% showed the largest gap loss (0.0378). A

trade-off thus exists between absolute performance and generalization consistency.

### D. Selection of the Best Fine-Tuning Strategy

Full Fine-Tuning is selected as the optimal strategy based on its superior validation performance (highest validation accuracy, 98.56%; lowest validation loss, 0.1353), which is subsequently confirmed by the lowest test loss (0.1605) and highest F1-score (97.35%) once evaluated on the test set. While Freeze 20% offered a smaller generalization gap, Full Fine-Tuning's stronger validation performance led to its selection as the baseline for the subsequent hyperparameter optimization.

### 2. Effect of Batch Size

This subsection analyzes the effect of batch size (32, 16, 8) on DenseNet201 performance across nine scenarios combining three fine-tuning strategies and three learning rates.

### A. Quantitative Performance Across Batch Sizes.

Quantitative performance was assessed using test accuracy, test loss, recall, precision, and F1-score. The comparison focused on the best-performing scenario per batch size under Full Fine-Tuning with LR=$1\times10^{-5}$, as shown in Table 10.

Table 10. Best-Scenario Performance Comparison per Batch Size (Full Fine-Tuning, LR=$1\times10^{-5}$)

| BS | LR | Test Acc | Test Loss | Avg Recall | Avg Precision | Avg F1-Score |
|---|---|---|---|---|---|---|
| 32 | $1\times10^{-5}$ | 0.9591 | 0.2040 | 0.9576 | 0.9639 | 0.9600 |
| 16 | $1\times10^{-5}$ | 0.9727 | 0.1605 | 0.9717 | 0.9757 | 0.9735 |
| 8 | $1\times10^{-5}$ | 0.9806 | 0.1117 | 0.9801 | 0.9823 | 0.9811 |

Table 10 shows that BS=8 achieved the highest test accuracy (98.06%), lowest test loss (0.1117), and highest F1-score (98.11%), followed by BS=16 and BS=32. The higher performance observed with BS=8 is consistent with previous findings that larger batch sizes do not necessarily yield higher accuracy in medical image classification [24]. From an optimization perspective, smaller mini-batches provide gradient estimates from fewer samples and result in more frequent parameter updates within an epoch, which can alter convergence and generalization behavior [25]. Because batch size and learning rate can also interactively affect model performance [26], the observed performance of BS=8 should be interpreted as a configuration-specific finding under the conditions evaluated in this study.

### B. Consistency Across Learning Rates

Performance consistency was evaluated across the tested learning rates using the accuracy range for each batch size as shown in Table 11.

Table 11. Accuracy Performance Range per Batch Size

| Batch Size | Best Accuracy | Worst Accuracy | Accuracy Range |
|---|---|---|---|
| 32 | 0.9670 | 0.9476 | 0.0194 |
| 16 | 0.9727 | 0.9591 | 0.0136 |
| 8 | 0.9806 | 0.9626 | 0.0180 |

Table 11 shows that BS=16 had the smallest accuracy range (1.36%), compared with 1.80% for BS=8 and 1.94% for BS=32, indicating the most consistent performance across the tested learning rates. In contrast, BS=8 achieved the highest peak accuracy but exhibited greater sensitivity to learning-rate changes. This difference is consistent with previous findings that batch size and learning rate interactively influence transfer-learning performance [26]. Thus, BS=16 provided greater robustness to the tested learning-rate variations, whereas BS=8 offered higher peak performance under its optimal configuration.

### C. Convergence Stability Analysis (Batch Size)

Convergence stability was analyzed through training history curves for the best scenario of each batch size: Full Fine-Tuning with LR=$1\times10^{-5}$.

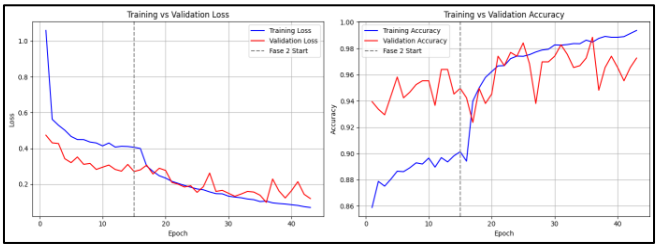


Figure 5. Training History Curve BS=8, Full Fine-Tuning, LR=$1\times10^{-5}$

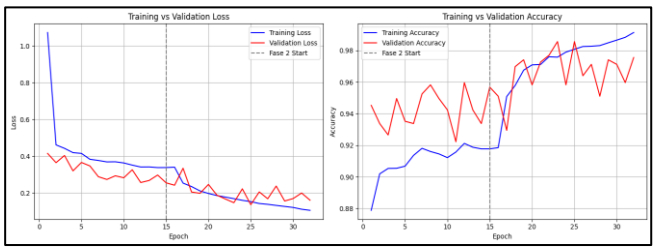


Figure 6. Training History Curve BS=16, Full Fine-Tuning, LR=$1\times10^{-5}$

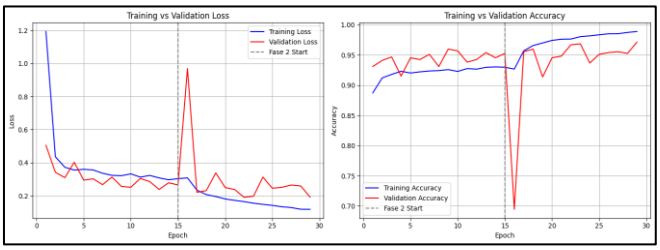


Figure 7. Training History Curve BS=32, Full Fine-Tuning, LR=$1\times10^{-5}$

Convergence stability was assessed from the training histories of the best-performing configuration for each batch size (Full Fine-Tuning, LR=$1\times10^{-5}$). Figure 5 (BS=8) exhibits consistent loss reduction alongside notable epoch-to-epoch validation fluctuations, which may be associated with the stochastic nature of smaller mini-batch updates [25]. Despite these fluctuations, the model converged and achieved the highest final performance. Figure 6 (BS=16) shows the most controlled convergence, characterized by smoother loss reduction and steady improvements in validation accuracy. In contrast, Figure 7 (BS=32) displays a pronounced validation-loss spike and a temporary drop in accuracy around epoch 16, coinciding with the transition to full fine-tuning. Although the model subsequently recovered and converged, its final performance remained below that of BS=8 and BS=16. These results indicate that convergence stability and final classification performance were not optimized by the same batch size.

### D. Generalization and Overfitting Risk Analysis

Table 12. Generalization Gap Analysis per Batch Size (All Scenarios)

| BS | Val. Acc | Test. Acc | Gap. Acc | Val. Loss | Test. Loss | Gap Loss |
|---|---|---|---|---|---|---|
| 32 | 0.9669 | 0.9591 | 0.0078 | 0.1910 | 0.2040 | 0.0130 |
| 16 | 0.9856 | 0.9727 | 0.0129 | 0.1353 | 0.1605 | 0.0252 |
| 8 | 0.9885 | 0.9806 | 0.0079 | 0.0983 | 0.1117 | 0.0134 |

Table 12 shows that all batch sizes produced validation–test accuracy gaps below 2%. BS=32 had the smallest accuracy gap (0.78%) but the lowest absolute accuracy, demonstrating that a smaller generalization gap alone does not indicate superior classification performance. BS=8 achieved the highest validation accuracy (98.85%) and lowest validation loss (0.0983), with a similarly small accuracy gap of 0.79%, indicating a small difference between validation and test performance. The absence of a sustained increase in validation loss in the training histories further provides no clear evidence of overfitting.

### E. Selection of the Best Batch Size

BS=8 was selected as the optimal batch size because it achieved the highest validation accuracy (98.85%) and, as shown in Table 10, the highest test accuracy (98.06%) and F1-score (98.11%), while maintaining a small validation–test accuracy gap of 0.79%. Although BS=16 showed greater consistency and BS=32 had the smallest generalization gap, neither achieved comparable absolute performance. Therefore, BS=8 was selected as the baseline batch size for the subsequent learning-rate optimization.

### 3. Effect of Learning Rate

This subsection analyzes the effect of learning rate ($1\times10^{-5}$, $5\times10^{-6}$, $1\times10^{-6}$) using Full Fine-Tuning with batch size 8, so performance differences can be attributed to learning-rate variation alone.

### A. Quantitative Performance Across Learning Rates

Quantitative performance was assessed using test accuracy, test loss, recall, precision, and F1-score under Full Fine-Tuning with BS=8, as shown in Table 13.

The results show a consistent performance advantage for LR=$1\times10^{-5}$, which achieved the highest test accuracy (98.06%) and F1-score (98.11%) and the lowest test loss (0.1117). Performance declined progressively at $5\times10^{-6}$ and $1\times10^{-6}$. Because the learning rate determines the step size of weight updates, smaller values produce more conservative parameter adjustments during fine-tuning [27]. In medical image transfer learning, reduced learning rates have also been reported to decrease classification performance in some settings, while the optimal value depends on the model and dataset characteristics [28]. Thus, the results suggest that the lower learning rates adapted more slowly within the available training budget, making $1\times10^{-5}$ the best-performing value among those evaluated.

Table 13. Performance Comparison per Learning Rate (BS=8, Full Fine-Tuning)

| LR | Test Acc | Test Loss | Avg Recall | Avg Precision | Avg F1-Score |
|---|---|---|---|---|---|
| $1\times10^{-5}$ | 0.9806 | 0.1117 | 0.9801 | 0.9823 | 0.9811 |
| $5\times10^{-6}$ | 0.9713 | 0.1605 | 0.9695 | 0.9755 | 0.9721 |
| $1\times10^{-6}$ | 0.9626 | 0.2417 | 0.9595 | 0.9697 | 0.9636 |

### B. Performance Trend and Cross-Batch-Size Consistency

Cross-batch-size consistency was evaluated using F1-score and test loss across all batch sizes under Full Fine-Tuning, as shown in Table 14.

Table 14. F1-Score and Test Loss Trend per Learning Rate Across Batch Sizes (Full Fine-Tuning).

| Batch Size | Metric | LR = $1\times10^{-5}$ | LR = $5\times10^{-6}$ | LR = $1\times10^{-6}$ | Best LR |
|---|---|---|---|---|---|
| 32 | F1-Score | 0.9600 | 0.9489 | 0.9547 | $1\times10^{-5}$ |
| | Test Loss | 0.2040 | 0.2429 | 0.2408 | $1\times10^{-5}$ |
| 16 | F1-Score | 0.9735 | 0.9616 | 0.9602 | $1\times10^{-5}$ |
| | Test Loss | 0.1605 | 0.2158 | 0.2340 | $1\times10^{-5}$ |
| 8 | F1-Score | 0.9811 | 0.9721 | 0.9636 | $1\times10^{-5}$ |
| | Test Loss | 0.1117 | 0.1605 | 0.2417 | $1\times10^{-5}$ |

Table 14 shows that LR = $1\times10^{-5}$ consistently achieved the highest F1-score and the lowest test loss across all three batch sizes. This pattern indicates that the highest learning rate evaluated in this study provided the best classification performance under the tested configurations.

### C. Convergence Stability Analysis (Learning Rate)

Convergence stability was analyzed using training history curves (training loss, validation loss, training accuracy, validation accuracy) under Full Fine-Tuning with BS=8.

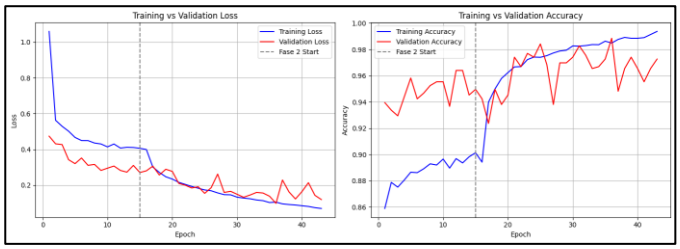


Figure 8. Training History Curve LR=1×10⁻⁵, BS=8, Full Fine-Tuning

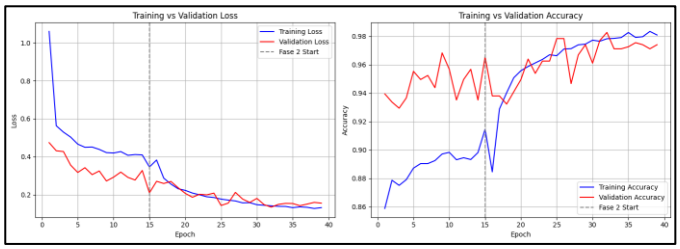


Figure 9. Training History Curve LR=5×10⁻⁶, BS=8, Full Fine-Tuning

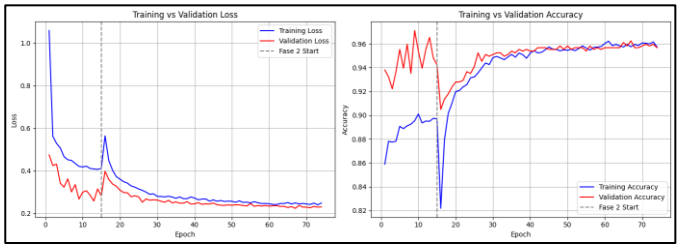


Figure 10. Training History Curve LR=1×10⁻⁶, BS=8, Full Fine-Tuning

Convergence stability was assessed from the training histories of the three learning rates (BS = 8, Full Fine-Tuning). Figure 8 (LR = 1 × 10⁻⁵) shows the fastest and most consistent loss reduction, with training accuracy approaching 99% and validation accuracy increasing without substantial fluctuation. Figure 9 (LR = 5 × 10⁻⁶) exhibits smoother but more gradual convergence, while Figure 10 (LR = 1 × 10⁻⁶) shows the slowest loss reduction and more conservative accuracy improvement. All configurations converged without divergence. This behavior is consistent with previous findings that smaller learning rates promote greater fine-tuning stability through more conservative parameter updates, although they may require more optimization steps to converge. In contrast, a relatively larger learning rate can accelerate parameter adaptation when appropriately selected. For the present experiment, LR = 1 × 10⁻⁵ provided the best balance between convergence speed and stability, resulting in the strongest final performance [29].

### D. Generalization and Overfitting Risk Analysis

Generalization was assessed using accuracy and loss gap indicators across learning rates under the BS=8 configuration (Tables 15 and 16).

Table 15. Generalization Capability Comparison Across Learning Rates (BS=8, Full Fine-Tuning)

| LR | Val Acc | Test Acc | Gap Acc | Val Loss | Test Loss | Gap Loss |
|---|---|---|---|---|---|---|
| 1×10⁻⁵ | 0.9885 | 0.9806 | 0.0079 | 0.0983 | 0.1117 | 0.0134 |
| 5×10⁻⁶ | 0.9827 | 0.9713 | 0.0114 | 0.1342 | 0.1605 | 0.0263 |
| 1×10⁻⁶ | 0.9625 | 0.9626 | 0.0001 | 0.2236 | 0.2417 | 0.0181 |

Table 15 shows all configurations generalize well, with small validation-to-test gaps. LR=1×10⁻⁵ achieved the highest absolute performance (val. acc. 98.85%, test acc. 98.06%) with a gap accuracy of 0.79% and gap loss of 0.0134. LR=5×10⁻⁶ produced a larger gap (1.14% accuracy, 0.0263 loss) with lower performance, while LR=1×10⁻⁶ had the smallest gap accuracy (0.01%) but also the lowest validation and test accuracy showing that an especially small gap simply reflects a model that has not yet reached its optimum.

Table 16. Generalization Gap Analysis per Learning Rate (BS=8, Across Fine-Tuning Strategies)

| LR | Best Acc | Worst Acc | Acc Range | Best Loss | Worst Loss | Loss Range |
|---|---|---|---|---|---|---|
| 1×10⁻⁵ | 0.9806 | 0.9720 | 0.0086 | 0.1117 | 0.1932 | 0.0815 |
| 5×10⁻⁶ | 0.9727 | 0.9634 | 0.0093 | 0.1557 | 0.2036 | 0.0479 |
| 1×10⁻⁶ | 0.9677 | 0.9626 | 0.0051 | 0.1947 | 0.2417 | 0.0470 |

Table 16 shows each learning rate has a different consistency profile across fine-tuning strategies. LR = 1 × 10⁻⁶ has the smallest accuracy and loss ranges (0.0051, 0.0470) but lower absolute accuracy than LR = 1 × 10⁻⁵, which achieved the highest accuracy in every strategy tested (97.20%–98.06%) despite a moderately higher sensitivity to strategy changes. LR = 5 × 10⁻⁶ shows intermediate performance, with an accuracy range of 0.0093 and a loss range of 0.0479.

### E. Selection of the Best Learning Rate

LR = 1 × 10⁻⁵ is selected as the optimal learning rate based on its validation performance: the highest validation accuracy (98.85%) and lowest validation loss (0.0983), the fastest and most stable convergence (Section C), and a validation–test gap of only 0.79% accuracy and 0.0134 loss. Its resulting test accuracy of 98.06% and F1-score of 98.11% further demonstrate strong performance on the held-out test set. LR = 1 × 10⁻⁶ has the smallest accuracy and loss ranges across fine-tuning strategies (0.0051 and 0.0470) but lower absolute accuracy (96.26%–96.77%), while LR = 5 × 10⁻⁶ shows intermediate performance with an accuracy range of 0.0093 and a loss range of 0.0479.

## 4. Classification Performance of the Selected Model

DOI: 10.34148/teknika.v15i3.1533
TEKNIKA, Volume 15(3), November 2026, pp. 421-431
ISSN 2549-8037, EISSN 2549-8045

Table 17. Classification Report by Class and ROC-AUC

| Class | Prec | Recall | F1-Score | ROC-AUC | Support |
|---|---|---|---|---|---|
| Normal | 0.9660 | 0.9854 | 0.9756 | 0.9986 | 548 |
| Pneumonia | 0.9859 | 0.9813 | 0.9836 | 0.9995 | 427 |
| Tuberculosis | 0.9951 | 0.9736 | 0.9842 | 0.9997 | 417 |
| Macro Avg | 0.9823 | 0.9801 | 0.9811 | 0.9993 | 1,392 |

Table 17 presents the classification report of the selected model on the 1,392 test images, which achieved an overall accuracy of 98.06% (1,365 correct). The Tuberculosis class attained the highest precision (0.9951) and ROC-AUC (0.9997), and no misclassification occurred between the Pneumonia and Tuberculosis classes. Figure 11 shows the corresponding confusion matrix.

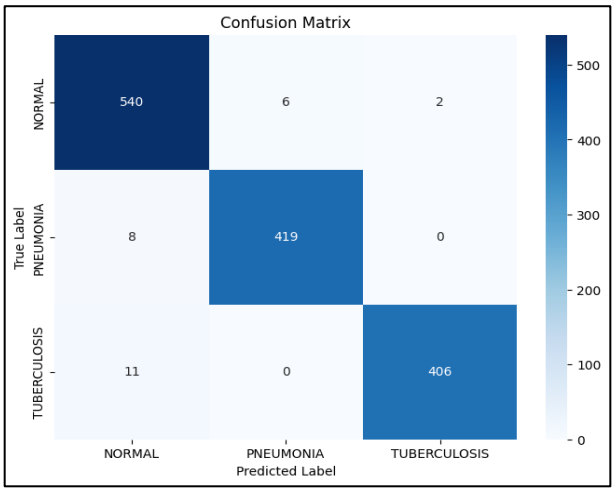


Figure 11. Confusion Matrix of the Selected Model

Based on the confusion matrix in Figure 11 and Table 17, the model correctly classified 1,365 of 1,392 test images, resulting in 27 errors. Most errors involved disease images predicted as Normal, consisting of 8 Pneumonia and 11 Tuberculosis cases, totaling 19 false negatives. Conversely, 8 Normal images were incorrectly classified as disease, including 6 as Pneumonia and 2 as Tuberculosis. No Pneumonia case was misclassified as Tuberculosis or vice versa, showing that the model distinguishes the two disease classes well. Thus, the main remaining errors involved distinguishing disease images from Normal images rather than differentiating between Pneumonia and Tuberculosis.

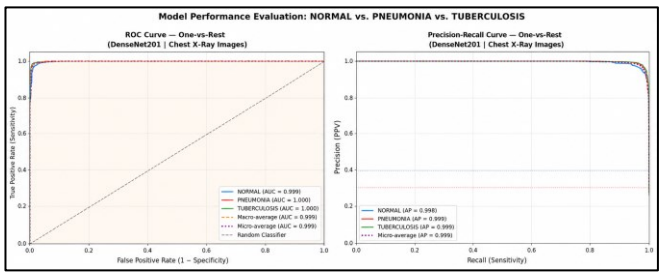


Figure 12. ROC Curve (Left) and Precision-Recall Curve (Right) One-vs-Rest, DenseNet201

The ROC curves in Figure 12 show very high AUC values for all classes (0.999 for Normal, 1.000 for Pneumonia, and 1.000 for Tuberculosis), with a macro-average of 0.999. The Precision-Recall curves also show high Average Precision values (0.998 for Normal and 0.999 for Pneumonia and Tuberculosis), indicating strong class separability across decision thresholds. However, the near-perfect AUC values, particularly 1.000 for Pneumonia and Tuberculosis, warrant cautious interpretation. Such performance may reflect the strong discriminative representations learned by DenseNet201, but may also be influenced by dataset characteristics or potential confounding factors, including data leakage or shortcut learning. For example, similarities or dependencies between training and test images, or class-specific image characteristics unrelated to pathology, could contribute to overly optimistic evaluation results. Therefore, although the results demonstrate strong discrimination on the test set, they should not be interpreted as evidence of perfect clinical generalization. Further validation using independent datasets and patient-level separation would be necessary to assess the robustness and generalizability of the model.

## 5. Grad-CAM Visualization and Clinical Interpretability

Grad-CAM was applied at the last convolutional layer (conv5_block32_concat) to visualize class-discriminative activation regions. Each visualization consists of three panels: the CLAHE-preprocessed input, the Grad-CAM heatmap (red = high activation, blue = low activation), and the overlay. Figures 13, 14, and 15 show results for the Normal, Pneumonia, and Tuberculosis classes respectively.

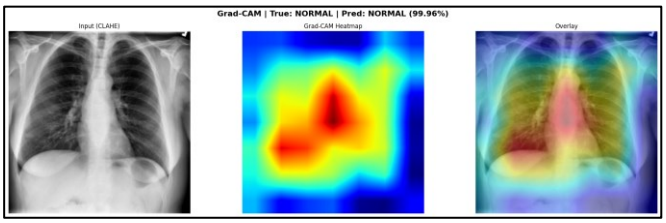


Figure 13. Grad-CAM Visualization Class Normal

Figure 13 shows the Grad-CAM visualization for a Normal image correctly classified with 99.96% confidence. Activation concentrates on the mediastinum and cardiomediastinal region, spreading weakly into both lung fields, indicating that the central anatomical structure, rather than any focal lesion, drives the model's decision. Unlike disease images, which typically show a focal parenchymal lesion, the Normal image shows no such pattern, consistent with the class's 98.54% recall.

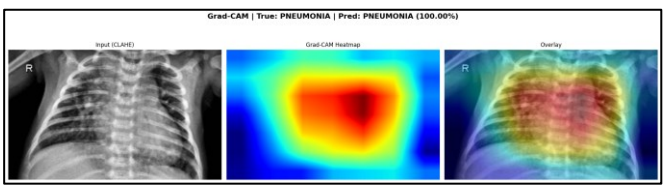


Figure 14. Grad-CAM Visualization Class Pneumonia

Figure 14 shows the Grad-CAM visualization for a Pneumonia image correctly classified with 100.00% confidence. High activation spreads diffusely and bilaterally across the mid-to-upper lung fields, showing a broader distribution than that observed in the Normal case and consistent with alveolar consolidation and peribronchial infiltrates characteristic of pneumonia [30], and confirming the model relies on lung-parenchyma features for this class. This matches its 98.59% precision, 98.13% recall, and the absence of any Pneumonia case misclassified as Tuberculosis.

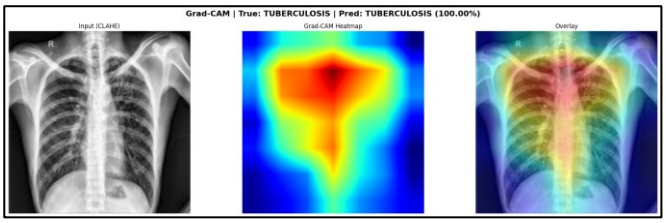


Figure 15. Grad-CAM Visualization Class Tuberculosis

Figure 15 shows the Grad-CAM visualization for a Tuberculosis image correctly classified with 100.00% confidence. Activation is dominant in the bilateral lung apices before extending toward the mediastinum, consistent with the predilection of *Mycobacterium tuberculosis* for the lung apices, driven by the higher partial oxygen pressure favoring growth of this obligate aerobe [1] and more concentrated than the patterns for Normal or Pneumonia. This matches the class's 99.51% precision, 97.36% recall, and the absence of any Tuberculosis case misclassified as Pneumonia.

Table 18. Summary of Grad-CAM Interpretation per Class

| Class | Confidence | Activation Region |
|---|---|---|
| Normal | 99.96% | Mediastinum and cardiomediastinal region, extending weakly into both lung fields |
| Pneumonia | 100.00% | Diffuse bilateral mid-to-upper lung fields |
| Tuberculosis | 100.00% | Bilateral apical regions extending toward the mediastinum |

Across all three classes, the model shows a distinct, class-specific attention pattern (central for Normal, diffuse bilateral for Pneumonia, apical for Tuberculosis), consistent with the confusion matrix's finding of no direct Pneumonia–Tuberculosis misclassification, suggesting that DenseNet201 learned discriminative representations without explicit anatomical supervision (Table 18). Grad-CAM nonetheless only reflects the region contributing most to the decision via last-layer activations, not clinical evidence of the true lesion location; these visualizations therefore serve to improve model transparency rather than as a diagnostic tool.

## 6. Comparison with Prior Work

Table 19 contextualizes the proposed approach against recent related studies.

Table 19. Comparison with Prior Work

| Study | Architecture | Task | Best Acc |
|---|---|---|---|
| Kiziloluk et al. [31] | AlexNet, DarkNet19 | Pneumonia + COVID-19 | 97.92% |
| Wajgi et al. [32] | VGG19 | TB vs Normal | 98.00% |
| Pamungkas et al. [33] | DenseNet121, DenseNet169, DenseNet201 | 3-class Pneumonia Normal TB | 97.11% |
| Behera et al. [34] | EfficientNetB7 + SE | Pneumonia vs Normal | 98.31% |
| This Study | DenseNet201 | Normal + Pneumonia + TB (3-class) | 98.06% F1: 0.9811 AUC: 0.999 |

The proposed approach achieved an accuracy of 98.06% for three-class classification of Normal, Pneumonia, and Tuberculosis. Compared with Pamungkas et al.[33], who evaluated DenseNet121, DenseNet169, and DenseNet201 on the same three disease classes, DenseNet169 achieved the best performance, with an accuracy of 97.11%. The proposed approach achieved a numerically higher accuracy by 0.95 percentage points. However, this comparison should not be interpreted as a direct improvement because the studies used different datasets, data distributions, preprocessing procedures, and training configurations. Instead, the result indicates that the proposed hyperparameter optimization strategy achieved competitive performance within a comparable three-class classification setting. Furthermore, the macro F1-score of 0.981 and ROC-AUC of 0.999 demonstrate strong and balanced classification performance across the three classes.

## IV. CONCLUSION

This study shows that transfer learning with DenseNet201 can achieve strong three-class chest X-ray classification when the training configuration is chosen carefully. Among 27 tested scenarios, Full Fine-Tuning with batch size 8 and learning rate $1\times10^{-5}$ performed best, reaching 98.06% test accuracy, a macro F1-score of 0.981, and a macro ROC-AUC of 0.999. Learning rate $1\times10^{-5}$ consistently outperformed the other rates across all batch sizes; batch size 8 gave the highest overall performance despite larger validation fluctuations, while batch size 16 trained most steadily and batch size 32 was least stable, with a validation-loss spike at the fine-tuning transition and the lowest final performance. Full Fine-Tuning also outperformed the partial-freezing strategies, which converged more stably but with a smaller generalization gap.

Grad-CAM visualization further showed class-specific activation patterns that were associated with different regions of the chest X-ray images: central activation for Normal,

DOI: 10.34148/teknika.v15i3.1533

diffuse bilateral activation for Pneumonia, and apical activation for Tuberculosis. These visualizations were obtained without anatomical supervision and are intended to illustrate the image regions that contributed to the model's classification decisions, rather than to determine disease locations or serve as a diagnostic tool. Future work should focus on reducing the 19 remaining disease-to-Normal false negatives through ensemble or attention-based approaches, and on validating the model with prospective clinical data, including external, cross-institutional chest X-ray datasets that vary in acquisition device, patient population, and imaging protocol, to assess whether the near-perfect AUC observed on this single pre-split dataset generalizes beyond it.

## REFERENCES


[1] A. Badu-Peprah, Y. D. Appiah, J. B. Babatunde, Z. Alebache, S. J. Badu-Peprah, and F. Quarshie, "Abnormal chest X-ray findings of patients with confirmed tuberculosis in Ghana," *Cureus*, vol. 17, no. 6, Art. no. e86359, 2025, doi: 10.7759/cureus.86359.

[2] Forum of International Respiratory Societies, "Pneumonia Remains a Leading Cause of Child Mortality," FIRS Official Website. [Online]. Available: https://firsnet.org/on-world-pneumonia-day-experts-demand-action-to-save-children

[3] World Health Organization, "Tuberculosis resurges as top infectious disease killer," World Health Organization. [Online]. Available: https://www.who.int/news/item/29-10-2024-tuberculosis-resurges-as-top-infectious-disease-killer

[4] K. K. R. Indonesia, "Indonesia's movement to end TB." [Online]. Available: https://kemkes.go.id/eng/indonesias-movement-to-end-tb

[5] T. Rahman *et al.*, "Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization," *IEEE Access*, vol. 8, pp. 191586–191601, 2020.

[6] D. W. Feyisa, Y. M. Ayano, and T. G. Debelee, "Weak localization of radiographic manifestations in pulmonary tuberculosis from chest X-ray: A systematic review," *Sensors*, vol. 23, no. 15, Art. no. 6781, 2023, doi: 10.3390/s23156781.

[7] M. A. Elemraid et al., "Accuracy of the interpretation of chest radiographs for the diagnosis of paediatric pneumonia," *PLoS ONE*, vol. 9, no. 8, Art. no. e106051, 2014, doi: 10.1371/journal.pone.0106051.

[8] S. Bennani et al., "Using AI to improve radiologist performance in detection of abnormalities on chest radiographs," *Radiology*, vol. 309, no. 3, Art. no. e230860, 2023, doi: 10.1148/radiol.230860.

[9] M. Bhandari, T. B. Shahi, B. Siku, and A. Neupane, "Explanatory classification of CXR images into COVID-19, pneumonia and tuberculosis using deep learning and XAI," *Computers in Biology and Medicine*, vol. 150, Art. no. 106156, 2022, doi: 10.1016/j.compbiomed.2022.106156.

[10] S. Dardouri, "A deep convolutional neural network model for lung disease detection using chest X-ray imaging," *Pulmonary Medicine*, vol. 2025, Art. no. 6614016, 2025, doi: 10.1155/pm/6614016.

[11] R. Jha, V. Bhattacharjee, and A. Mustafi, "Transfer Learning with Feature Extraction Modules for Improved Classifier Performance on Medical Image Data," *Sci. Program.*, vol. 2022, 2022, doi: 10.1155/2022/4983174.

[12] A. Davila, J. Colan, and Y. Hasegawa, "Comparison of fine-tuning strategies for transfer learning in medical image classification," *Image and Vision Computing*, vol. 146, Art. no. 105012, 2024, doi: 10.1016/j.imavis.2024.105012.

[13] G. Huang and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[14] K. Srilatha and E. Saikiran, "Advanced skin cancer detection: A deep learning and transfer learning framework for melanoma classification," *Journal of Information Systems Engineering and Management*, vol. 10, no. 23s, pp. 140–154, 2025, doi: 10.52783/jisem.v10i23s.3687.

[15] H. Malik, T. Anees, M. Din, and A. Naeem, "CDC_Net: Multi-classification convolutional neural network model for detection of COVID-19, pneumothorax, pneumonia, lung cancer, and tuberculosis using chest X-rays," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13855–13880, 2023, doi: 10.1007/s11042-022-13843-7.

[16] N. Bhushan, S. Mekhilef, K. S. Tey, M. Shaaban, and M. Seyedmahmoudian, "Dynamic K-Decay Learning Rate Optimization for Deep Convolutional Neural Network to Estimate the State of Charge," *Energies*, vol. 17, no. 16, Art. no. 3884, 2024, doi: 10.3390/en17163884.

[17] R. Bumbálek, T. Zoubek, J. de Dieu, M. Ufitikirezi, S. N. Umurungi, and R. Stehlík, "Implementation of Machine Vision Methods for Cattle Detection and Activity Monitoring," *Technologies*, vol. 13, no. 3, Art. no. 116, 2025, doi: 10.3390/technologies13030116.

[18] W. Samek and K.-R. Müller, "Towards Explainable Artificial Intelligence," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol. 11700, Springer, 2019, pp. 5–22, doi: 10.1007/978-3-030-28954-6_1.

[19] I. E. Ihongbe, S. Fouad, T. F. Mahmoud, A. Rajasekaran, and B. Bhatia, "Evaluating explainable artificial intelligence (XAI) techniques in chest radiology imaging through a human-centered lens," *PLoS ONE*, vol. 19, no. 10, Art. no. e0308758, 2024, doi: 10.1371/journal.pone.0308758.

[20] A. Hazlett, N. Ohashi, T. Rodriguez, and S. Adewole, "Chest disease detection in X-ray images using deep learning classification method," *arXiv preprint arXiv:2505.22609*, 2025.

[21] T. Zhou, X. Y. Ye, H. L. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense convolutional network and its application in medical image analysis," *BioMed Research International*, vol. 2022, Art. no. 2384830, 2022, doi: 10.1155/2022/2384830.

[22] P.-H. Huynh and T.-D. Truong, "Dynamic fine-tuning strategy of deep learning models for lung disease classification on chest X-ray images," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 8, no. 3, pp. 1047–1059, 2026, doi: 10.35882/jeeemi.v8i3.1352.

[23] C. Coil and N. Cheney, "What makes freezing layers in deep neural networks effective? A linear separability perspective," in *Proc. Fourth Int. Conf. Automated Mach. Learn.*, Proc. Mach. Learn. Res., vol. 293, pp. 9/1–32, 2025.

[24] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020, doi: 10.1016/j.icte.2020.04.010.

[25] S. Lee, C. He, and S. Avestimehr, "Achieving small-batch accuracy with large-batch scalability via Hessian-aware learning rate adjustment," *Neural Networks*, vol. 158, pp. 1–14, 2023, doi: 10.1016/j.neunet.2022.11.007.

[26] I. A. Usmani, M. T. Qadri, R. Zia, F. S. Alrayes, and O. Saidani, "Interactive effect of learning rate and batch size to implement transfer learning for brain tumor classification," *Electronics*, vol. 12, no. 4, Art. no. 964, 2023, doi: 10.3390/electronics12040964.

[27] A. Sharifnassab, S. Salehkaleybar, and R. S. Sutton, "MetaOptimize: A framework for optimizing step sizes and other meta-parameters," in *Proc. 42nd Int. Conf. Mach. Learn.*, Proc. Mach. Learn. Res., vol. 267, pp. 54300–54325, 2025.

[28] C. Gu and M. Lee, "Deep transfer learning using real-world image features for medical image classification, with a case study on pneumonia X-ray images," *Bioengineering*, vol. 11, no. 4, Art. no. 406, 2024, doi: 10.3390/bioengineering11040406.

[29] Z. Fu, A. M.-C. So, and N. Collier, "A stability analysis of fine-tuning a pre-trained model," *arXiv preprint arXiv:2301.09820*, 2023, doi: 10.48550/arXiv.2301.09820.

[30] N. Alsubaie, T. Alqahtani, and S. B. Rahayu, "Lung-centric feature analysis for accurate pneumonia detection in chest X-ray images," in *Computer Science & Information Technology (CS & IT)*, D. C. Wyld et al., Eds., AIRCC Publishing Corporation, 2024, pp. 129–143, doi: 10.5121/csit.2024.140108.

[31] S. Kiziloluk, E. Sert, M. Hammad, R. Tadeusiewicz, and P. Pławiak, "EO-CNN: Equilibrium optimization-based hyperparameter tuning for enhanced pneumonia and COVID-19 detection using AlexNet and DarkNet19," *Biocybernetics and Biomedical Engineering*, vol. 44, no. 3, pp. 635–650, 2024, doi: 10.1016/j.bbe.2024.06.006.

[32] R. Wajgi, G. Yenurkar, V. O. Nyangaresi, and S. Mallewar, "Optimized tuberculosis classification system for chest X-ray images: Fusing hyperparameter tuning with transfer learning approaches," *Engineering Reports*, vol. 6, no. 11, Art. no. e12906, 2024, doi: 10.1002/eng2.12906.

[33] Y. Pamungkas, S. I. Sabilla, P. N. Crisnapati, Y. Gao, and Y. Thwe, "A comparative study of respiratory diseases classification using Grad-CAM-based DenseNet architectures," *International Journal of Robotics and Control Systems*, vol. 6, no. 1, pp. 454–474, 2026, doi: 10.31763/ijrcs.v6i1.2164.

[34] S. K. Behera, K. Murali Gopal, and S. B. Punuri, "A deep learning-based pneumonia detection system with explainable AI for medical decision support," in *Proc. 2025 11th Int. Conf. Commun. Signal Process. (ICCSP)*, IEEE, 2025, pp. 694–699, doi: 10.1109/ICCSP64183.2025.11089357.