Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

Implementation of Convolutional Neural Network (CNN) MobileNetV2 in Lung Disease Classification from X-Ray Images

Mohammad Faris Fawwaz*, Arif Aryaguna Nauli, Roslina Roslina

Software Engineering Technology Study Program, Politeknik Negeri Medan, Medan, Indonesia
Email: [1,*]mohammadfarisfawwaz@students.polmed.ac.id, [2]arifaryagunanauli@students.polmed.ac.id, [3]roslina@polmed.ac.id

ARTICLE INFORMATION

ARTICLE HISTORY:
Submited : October 05, 2025
Revised : December 18, 2025
Accept : May 30, 2026
Publish : May 30, 2026

KEYWORD

Lung Disease Classification;
X-Ray Image;
MobileNetV2;
Data Augmentation;
Mixup

CORRESPONDENCE AUTHOR

Email:
mohammadfarisfawwaz@students.polmed.ac.id

A B S T R A C T

The classification of lung diseases from X-ray images is often challenged by significant data imbalance, where minority classes like COVID-19 constitute only approximately 20% of the dataset compared to the majority classes. This condition can degrade model performance and introduce bias. This study aims to analyze the impact of data balancing strategies and training parameter variations to improve the accuracy of a Convolutional Neural Network (CNN) model based on the MobileNetV2 architecture. The experimental process systematically compared two learning rates (1e-3 and 1e-4) and two optimizers (Adam and RMSprop) across four distinct data handling scenarios: no augmentation, geometric augmentation only, the Mixup technique only, and a combination of both. The model was evaluated on a four-class X-ray image dataset comprising COVID-19, Normal, Pneumonia, and Tuberculosis. The optimal results were achieved by applying the combined approach of geometric augmentation and Mixup with a 1e-3 learning rate and the Adam optimizer. This configuration significantly outperformed other scenarios, reaching a testing accuracy of 96.62% and an average F1-Score of 96.63%, demonstrating excellent model generalization. This high-performing model has been successfully implemented in a mobile application using Flutter and TensorFlow Lite, serving as a practical tool to support the early diagnosis of lung diseases

1. INTRODUCTION

Lung diseases such as Pneumonia, Tuberculosis, and COVID-19 remain major diagnostic challenges, contributing significantly to morbidity and mortality worldwide. Early detection is essential to ensure timely treatment and reduce complications. Chest X-ray imaging is one of the most widely used diagnostic tools for lung disease detection, but manual interpretation requires radiology expertise and is time-consuming, which can limit efficiency in clinical practice [1]. Recent advances in artificial intelligence (AI), particularly deep learning, have opened new opportunities for automated medical image analysis. Convolutional Neural Networks (CNNs) are especially effective in extracting complex features directly from raw images, enabling accurate classification without handcrafted descriptors [2], [3]. These models have demonstrated strong performance across diverse medical imaging tasks, yet their reliability is often compromised by the quality and distribution of training data. One of the most persistent challenges is data imbalance, where majority classes dominate minority classes. This imbalance can lead to biased models that achieve high overall accuracy but poor sensitivity for rare conditions [3].

Several studies have demonstrated the impact of data imbalance on CNN performance. Karlita et al. applied MobileNetV2 for COVID-19 detection, achieving 81% accuracy but only 75% recall, indicating difficulty in identifying positive cases [4]. Maysanjaya reported 89.58% accuracy for pneumonia classification, but the high loss value (47.43%) suggested overfitting due to unbalanced data [5]. Bahri et al. achieved 89% accuracy with CNN for Tuberculosis detection, but the F1-score of 0.89 highlighted reduced sensitivity for minority classes [3]. Similarly, Hekmatyar et al. used Faster R-CNN for COVID-19 pneumonia detection, reporting 85–86% accuracy, but the limited number of COVID-19 samples remained a bottleneck [6]. Interestingly, Safitri and Srimarinda compared CNN with Support Vector Machine (SVM) for pneumonia detection. Their study found that while CNN achieved 0.82 accuracy, SVM slightly outperformed with 0.83 accuracy and higher sensitivity (0.85). This result underscores that in imbalanced datasets, sensitivity is often more critical than accuracy, and threshold tuning can improve performance [1]. Such findings emphasize the importance of evaluating models beyond accuracy, especially in medical contexts where false negatives can be fatal.

To mitigate data imbalance, augmentation techniques have been widely adopted. Geometric transformations such as rotation, flipping, and zooming increase dataset diversity, improving generalization. More advanced methods like Mixup augmentation which blends two images with random weights have proven effective in reducing bias toward majority classes and enhancing sensitivity for minority classes [7]. These approaches not only expand the effective dataset size but also encourage the model to learn more robust decision boundaries, reducing overfitting and improving fairness across classes.

MobileNetV2, a lightweight CNN architecture, has gained popularity due to its efficiency and suitability for mobile deployment. Harahap et al. applied MobileNetV2 to brain tumor classification, achieving 78–83% accuracy,

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

demonstrating its effectiveness even in resource-constrained environments [8]. Gao et al. used transfer learning with MobileNetV2 for lung tumor classification from CT scans, achieving an impressive accuracy of 95.2%. Their findings emphasized the importance of balanced datasets in ensuring robust generalization [9]. Tripathi et al. introduced MobileNetV2L2, incorporating L2 regularization and CLAHE preprocessing, and achieved 95.51% accuracy. Their work highlighted that augmentation and regularization techniques can effectively mitigate the negative effects of data imbalance [7]. Beyond medical imaging, MobileNetV2 has also been successfully applied in ecological domains. Maulana et al. implemented MobileNetV2 for bird species detection, achieving reliable accuracy across 200 species. Their work demonstrated that MobileNetV2 can efficiently handle large-scale classification tasks on mobile devices [2]. This reinforces the architecture's versatility and suitability for real-world applications where computational resources are limited.

It is important to emphasize that automated classification systems are not substitutes for medical professionals. Instead, they serve as supportive tools that can assist radiologists in identifying potential abnormalities more efficiently. Final diagnosis and treatment decisions must always be validated by physicians to ensure patient safety and uphold ethical standards [1]. By framing AI as a decision support system, researchers can highlight its potential benefits without undermining the ethical responsibilities of healthcare providers.

From the literature, it is clear that while CNNs and MobileNetV2 are effective for image classification, their performance deteriorates under imbalanced datasets. Most prior studies have not explicitly focused on strategies to handle data imbalance, despite its critical importance in medical applications. This research addresses that gap by implementing a combination of Mixup augmentation and geometric transformations to balance data distribution. In addition, the study considers optimization strategies to further enhance model performance, ensuring that the proposed approach achieves both accuracy and sensitivity across classes. The main contribution of this study lies in integrating MobileNetV2 with advanced augmentation techniques for lung disease classification from chest X-rays. In addition to theoretical evaluation, the proposed model is briefly demonstrated in a mobile application environment to highlight its potential usability in real clinical settings. By improving sensitivity for minority classes such as Tuberculosis and COVID-19, the system supports early and equitable diagnosis, ultimately contributing to better healthcare outcomes while respecting the ethical boundaries of medical practice.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In the conduct of research, the ensuing stages of inquiry are requisite:

a. Problem Identification
   The initial phase of this research focused on identifying the core impediments in multi-class lung disease classification. Through a comprehensive literature review, two primary problems were established severe data imbalance, where the dominance of majority classes leads to model bias and low sensitivity for minority classes like COVID-19 and high inter-class visual similarity, which causes classification errors among diseases with overlapping radiological features. Consequently, the research objective was formulated to address these issues by optimizing the MobileNetV2 architecture through advanced data balancing strategies (Mixup) and geometric augmentation.
b. Data Acquisition
   Research data were procured from multiple chest X-ray image sources to ensure representativeness and diversity. The primary source comprised a Kaggle dataset from Aminu Kano Teaching Hospital, Nigeria [10], containing 2,600 images equally distributed across four classes (Pneumonia, COVID-19, Tuberculosis, and Normal). The secondary source, derived from a repository [11], included 5,228 images across three classes (COVID-19, Normal, Pneumonia), previously employed in the LiteCovidNet [12] and CheXImageNet [13] studies. The tertiary source [14] encompassed 2,494 Tuberculosis and 514 Normal images from a hospital in Pakistan. Additionally, internal data from Sundari Hospital Medan were incorporated, consisting of 116 images categorized as Tuberculosis (53), Normal (18), and Pneumonia (45).
c. Data Preprocessing
   Prior to utilization in model training, all X-ray image data undergoes a preprocessing phase to ensure standardization and quality assurance. This process commences with data cleansing, wherein image samples exhibiting distortion, excessive noise, or disruptive medical artifacts in the pulmonary region are manually removed. Subsequently, all images are resized to 224x224 pixels to conform to the standard input dimensions of the MobileNetV2 architecture. Finally, the dataset is randomly partitioned into three segments: 70% for training data, 15% for validation data, and 15% for testing data.
d. Data Augmentation
   To enhance diversity and address class imbalance, data augmentation was exclusively applied to the training dataset. This process involved class balancing through the utilization of the Mixup technique, wherein two images were merged with stochastic weighting. Subsequently, geometric augmentations, such as random rotations of ±10°, zooms of up to 10%, and horizontal flips, were implemented so as to augment spatial variance and attenuate the risk of overfitting.

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

e. Implementation of the MobileNetV2 CNN Model
The implementation leveraged transfer learning using the MobileNetV2 architecture pre-trained on ImageNet, where all 155 base layers from the initial convolution to the final bottleneck block were frozen to preserve pre-learned feature maps and reduce computational cost. A custom trainable classification head comprising GlobalAveragePooling2D, Dropout (0.2) to prevent overfitting, Dense ReLU, and a final Dense Softmax layer for the four target classes was appended to this base. To identify the optimal configuration, experimentation systematically contrasted Adam and RMSprop optimizers at learning rates of 1e-3 and 1e-4 over 50 epochs using categorical crossentropy loss, utilizing EarlyStopping and ReduceLROnPlateau callbacks for convergence, while evaluating four distinct data strategies ranging from baseline to combined geometric and Mixup augmentation.

f. Model Evaluation
Following training, the model undergoes evaluation utilizing test data to ascertain its performance and generalization capabilities. This evaluation employs a Confusion Matrix to compute quantitative metrics, encompassing accuracy, precision, recall, and the F1-score.

g. System Development
This stage encompasses the design and development of the classification system into a mobile application prototype. The system development is executed utilizing the Waterfall methodology, employing the Flutter framework, Dart programming language, and TensorFlow Lite (TFLite) for model integration.

h. System Testing
The functionality of the developed system is tested via Black Box Testing methodology. This testing focuses on functionality verification, without regard to internal code structure, specifically by providing X-ray image inputs and ensuring the resultant classification output aligns with expectations.

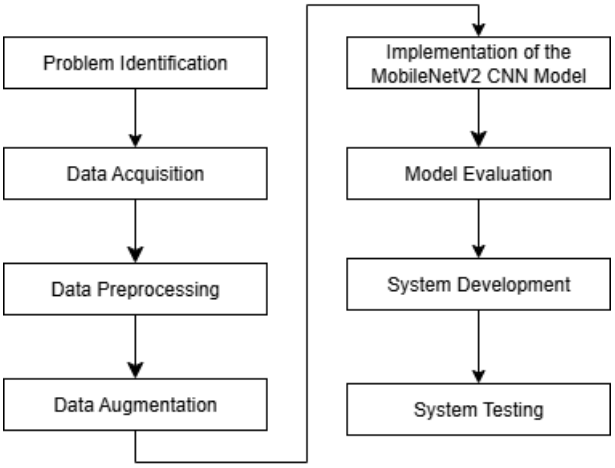

Figure 1. Research Stages

2.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) constitutes a widely employed deep learning algorithm, esteemed for its capacity to autonomously extract salient features, thereby obviating the necessity for manual feature engineering [15]. Convolutional Neural Networks (CNNs) employ convolutional layers to discern local patterns such as edges and textures, followed by hierarchical processing through activation functions, pooling layers, and fully connected layers for classification purposes. Generally, CNNs operate in two distinct phases: feature extraction, accomplished via convolutional layers and pooling operations; and classification, achieved by transforming extracted features into a one-dimensional vector, subsequently processed by fully connected layers in conjunction with a softmax function to yield a predictive outcome [16]. By virtue of these merits, CNNs find extensive application in computer vision, facial recognition, and medical image analysis.

2.3 MobileNetV2

MobileNetV2 constitutes a CNN architecture predicated upon computational efficiency and reduced parametric complexity, thereby optimizing its deployment upon resource-constrained devices such as mobile platforms [2]. This architecture employs a combination of depthwise convolution for feature extraction on each image channel and pointwise convolution of size 1x1 for their recombination. This process effectively reduces the parametric quantity and computational burden, rendering it highly efficient for tasks such as object detection [17]. As a refinement, MobileNetV2 was developed, incorporating a bottleneck residual block optimized for the preservation of salient information and the enhancement of power efficiency [18]. This architecture comprises two distinct types of residual blocks, featuring strides of 1 and 2, layered to a total depth of 155 layers across 16 blocks, thereby rendering it suitable for image classification on mobile devices.

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

2.4 Data Augmentation

Data augmentation constitutes a fundamental technique in deep learning, employed to systematically enhance the volume and diversity of training datasets without alteration of original labels [19]. The primary objective of this process is to augment model performance and accuracy. This technique engenders novel samples via geometric transformations, such as rotations and reflections, as well as photometric adjustments, encompassing modifications to brightness and contrast. This method not only mitigates data limitations, but also enhances the model's generalization prowess by emulating real-world conditions [20]. In the classification of pulmonary X-ray images, augmentation serves to simulate variations in patient positioning and beam intensity. Furthermore, advanced methodologies such as Mixup, which amalgamates two training images into a novel synthetic sample, are employed to enrich visual features and mitigate the risk of overfitting [21].

2.5 Flutter

Flutter is an open-source framework developed by Google for constructing cross-platform applications from a single Dart codebase. Its primary advantage lies in the hot reload feature, which substantially expedites the development process by instantly displaying the results of code modifications [22]. Its widget-based architecture, powered by the Skia rendering engine, facilitates the creation of user interfaces that are both flexible and efficient, while also possessing an aesthetically pleasing appearance [23]. Despite its popularity, Flutter exhibits certain limitations, such as a propensity for larger application sizes and a third-party library ecosystem that is not as extensive as that of native development. Nevertheless, with the robust support of Google and an ever-expanding community, Flutter remains highly regarded as a propitious solution for contemporary mobile application development [23].

2.6 Tensorflow Lite

TensorFlow Lite constitutes a framework integral to the TensorFlow ecosystem, meticulously designed for the execution of complex models on resource-constrained devices, encompassing Android applications [24]. The primary objective is to facilitate machine learning inference directly on devices (on-device inference), thereby diminishing latency and augmenting user privacy by obviating the necessity of transmitting data to servers. This process entails the conversion of standard TensorFlow models into a lightweight and optimized .tflite format. A salient advantage resides in TensorFlow Lite's capacity to operate autonomously, independent of an internet connection, rendering it exceptionally reliable for applications necessitating functionality in areas characterized by diminished connectivity. Furthermore, this framework exhibits enhanced flexibility, particularly in instances where developers employ custom models to address variegated specific requirements within machine learning-based applications [25].

3. RESULT AND DISCUSSION

This chapter delineates the findings of the conducted research, accompanied by pertinent discussion. The exposition shall commence with an elucidation of the applied methodology and utilized data.

3.1 Data Description

The X-ray image data employed in this study originate from a confluence of public and internal sources. Public sources encompass datasets from Kaggle and Mendeley Data. Internal sources were procured through collaborative efforts with Sundari Hospital Medan. All data are categorized into four classes: Normal, COVID-19, Pneumonia, and Tuberculosis, with the distribution thereof presented in Table 1.

Table 1. Quantity of X-Ray Images per Class

Class	Number of Images
COVID-19	2.276
Normal	2.984
Pneumonia	2.495
Tuberculosis	3.197
Total	10.952

3.2 Data Preprocessing

During the data cleaning phase, images exhibiting blur, noise, text occlusion, or a lack of pulmonary emphasis were removed. This process resulted in a 24.5% reduction in the dataset, equivalent to 2,679 images.

Table 2. Data Volume Following Cleaning

Class	Prior to Cleaning	Post Cleaning
COVID-19	2.276	1.683
Normal	2.984	2.402
Pneumonia	2.495	2.108

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

Class	Prior to Cleaning	Post Cleaning
Tuberculosis	3.197	2.080
Total	10.952	8.273

Subsequent to data cleaning, all images (8,273 in total) were resized to dimensions of 224×224 pixels. This particular dimension was selected to conform to the standard input format required by the MobileNetV2 architecture. Upon completion of the preprocessing stage, the X-ray image data was partitioned into three distinct sets: training data, validation data, and testing data, with proportional allocations of 70%, 15%, and 15%, respectively. The distribution of data quantities across each class, subsequent to this partitioning process, is delineated in Table 3 below.

Table 3. Data Distribution Following Partitioning

Class	Training	Validation	Testing
COVID-19	1.178	252	253
Normal	1.681	361	360
Pneumonia	1.476	316	316
Tuberculosis	1.456	312	312
Total	5.791	1.241	1.241

Given the persisting disparity in data volume across the various classes, data augmentation was employed to increase sample sizes and harmonize class distribution prior to model training.

3.3 Data Augmentation

Following the preprocessing and data partitioning procedures, the subsequent phase involves data augmentation and balancing. The initial step in balancing the training data is executed via the Mixup method, an oversampling technique that merges two images from the same class utilizing random weights (0–1) to yield novel synthetic images. This process is iterated until all classes possess an equivalent number of images as the majority class, thereby ensuring a balanced training data distribution and mitigating bias in the model towards specific classes. The Mixup process for images is depicted in Figure 2 below:

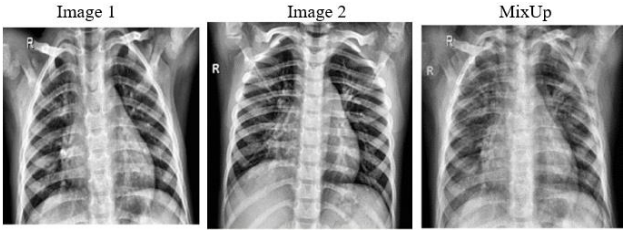

Figure 2. Image Mixup Process

Subsequent to balancing the training data, offline augmentation is performed employing simple transformations such as 10° rotation, 10% zoom, and horizontal flipping. An example of the image augmentation outcome is presented in Figure 3 below:

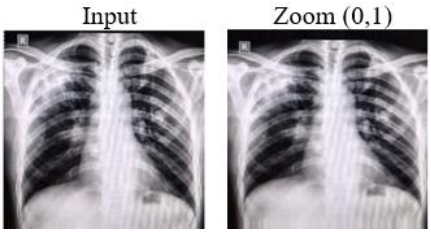

Figure 3. Example of Data Augmentation Outcome

Each image, both original and Mixup-generated, undergoes a single augmentation technique to enrich data variability without altering the labels. The final distribution of the training data following Mixup and augmentation is displayed in Table 4.

Table 4. Training Data Distribution After Mixup and Augmentation

Class	Training Data Count
COVID-19	3.362
Normal	3.362
Pneumonia	3.362
Tuberculosis	3.362
Total	13.448

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

3.4 Hyperparameter Experiment Results

Hyperparameter experiments were conducted on data subjected solely to preprocessing steps (cleaning, resizing, and splitting), without the implementation of augmentation or Mixup techniques. The objective of this approach was to ascertain the baseline performance of the MobileNetV2 architecture in its pure form against the original data, prior to the introduction of additional variations. Employing transfer learning, the pre-trained ImageNet MobileNetV2 architecture was utilized as a feature extractor. Subsequent to the removal of its inherent classification layer and the freezing of its weights, the model was retrained to classify X-ray images into four categories (COVID-19, Normal, Pneumonia, and Tuberculosis). To optimize the performance of the MobileNetV2 architecture, experiments were performed using the Adam and RMSprop optimizers with learning rates of 1e-3 and 1e-4. The results of these testing scenarios are presented in Table 5.

Table 5. Hyperparameter Experiment Results

Scenario	Optimizer	Learning Rate	Accuracy (Test)	Precision (Average)	Recall (Average)	Loss (Test)	F1-Score (Average)
1	Adam	1e-3	96.21%	96.44%	96.01%	0.1314	96.19%
2	Adam	1e-4	94.36%	94.54%	94.09%	0.1619	94.28%
3	RMSprop	1e-3	95.89%	95.91%	95.90%	0.1349	95.90%
4	RMSprop	1e-4	94.44%	94.62%	94.13%	0.1644	94.33%

Based on the experimental results presented in Table 5, the optimal hyperparameter configuration is the utilization of the Adam optimizer with a learning rate of 1e-3. This scenario yielded the best overall performance, with a testing accuracy of 96.21% and the lowest loss value of 0.1314. Furthermore, this scenario also achieved the highest average F1-Score of 96.19%, indicating that the model attained the most favorable balance between precision (96.44%) and recall (96.01%) compared to other configurations. These results also affirm that a learning rate of 1e-3 is consistently more effective than 1e-4 for both optimizers tested.

3.5 Augmentation and Data Balancing Experiment Results

Following the identification of optimal hyperparameter configurations through initial testing, this study proceeded with a focus on enhancing the quality of the training data. This step is critical for addressing class imbalance and limitations in data variation, which frequently impede medical image classification. Utilizing the Adam optimizer and a learning rate of 1e-3 as a foundation, four distinct data augmentation approaches were experimentally evaluated to determine their respective impacts on model performance. These testing scenarios were designed to compare the performance of a baseline model with models trained using geometric augmentation, Mixup, and a combination of both. The results of these testing scenarios are presented in Table 6.

Table 6. Augmentation and Data Balancing Experiment Results

Scenario Augmentation & Balancing	Accuracy (Test)	Precision (Average)	Recall (Average)	Loss (Test)	F1-Score (Average)
Without Augmentation & Mixup	96.37%	96.37%	96.31%	0.1255	96.34%
Mixup Only	96.37%	96.48%	96.34%	0.1210	96.41%
Augmentation Only	95.73%	95.76%	95.64%	0.1269	95.76%
Augmentation + Mixup	96.62%	96.83%	96.47%	0.1266	96.63%

Based on the experimental results presented in Table 6, it is concluded that the most effective augmentation approach is the combination of geometric augmentation and Mixup. This scenario demonstrates significant superiority by not only achieving the highest accuracy (96.62%) and F1-Score (96.63%), but also by successfully elevating the average precision (96.83%) and recall (96.47%) to their peak values. This indicates that the model becomes more accurate in its predictions (precision) and more reliable in identifying all positive cases (recall). In comparison, the use of Mixup alone yields only a marginal enhancement in both metrics while achieving the lowest loss, whereas geometric augmentation alone diminishes both precision and recall. Thus, it is evidenced that the synergy between data balancing through Mixup and the introduction of variation through geometric augmentation constitutes a comprehensive strategy for maximizing model performance. As depicted in Figure 4, the confusion matrix reveals that the model is highly robust, with very few misclassifications. For instance, out of 253 COVID-19 test images, only 4 were misclassified as Tuberculosis and 2 as Pneumonia. This indicates that despite the visual similarities between these lung diseases, the model successfully learned distinctive features for each class. To further substantiate the analysis, the results of this experiment is visualized using an accuracy and loss graphs as shown in figures 4 and 5. These visualize are showing trends on how the model improves its performance from initial to the last stage of the experiment. These graphs may also help evaluate the effectiveness of each and every stage of the optimization to also show how loss is reduced as the accuracy is increased.

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

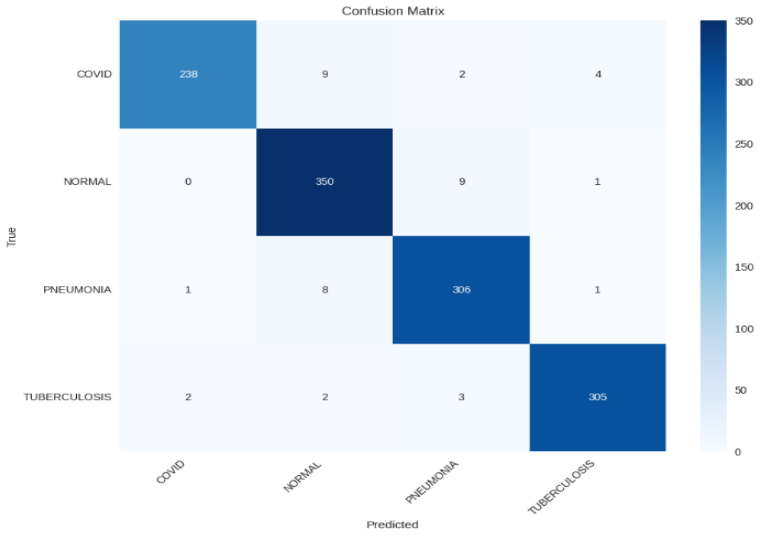

Figure 4. Confusion Matrix of the Optimal Experiment

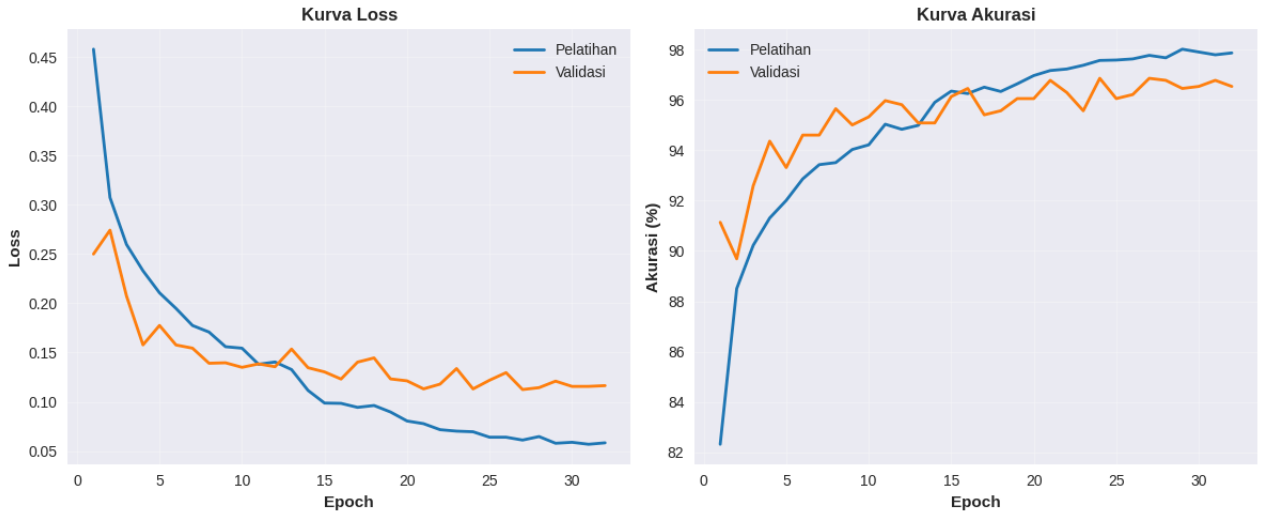

Figure 5. Accuracy and Loss Curves of the Optimal Experiment

These graphs provide critical insight into the model's generalization capability. As observed in the Accuracy Curve (Right), there is a minimal divergence (gap) between the training accuracy (blue line) and validation accuracy (orange line) throughout the 32 epochs. In typical deep learning scenarios prone to overfitting especially with imbalanced medical data the training accuracy often rapidly approaches 100% while validation accuracy plateaus significantly lower. However, in this experiment, the validation curve closely tracks the training curve, eventually converging at approximately 96%. This tight alignment proves that the Mixup strategy combined with geometric augmentation successfully acted as a regularizer. By training on synthetic interpolated images, the model was prevented from memorizing specific pixel patterns of the training data, forcing it instead to learn robust features that apply equally well to unseen data. Furthermore, the Loss Curve (Left) corroborates this stability; the validation loss decreases steadily and stabilizes around 0.12 without diverging or increasing at later epochs. This confirms that the high testing accuracy of 96.62% is a result of genuine learning and generalization, effectively addressing the overfitting issue raised in the problem statement.

3.6 Implementation of the Mobile Application System

The optimal model derived from the research, exhibiting an accuracy of 96.62%, was subsequently converted into the TensorFlow Lite (TFLite) format. This model was thereafter implemented within a mobile application, developed utilizing the Flutter framework and the Dart programming language. The ensuing sections delineate the appearance and functionality of each primary application page:

a. SplashScreen Page

This page constitutes the initial loading screen displayed upon application commencement. Its function is to exhibit the application's identity while concurrently preparing the system in the background prior to navigation to subsequent pages. The visual representation of the SplashScreen page is depicted as follows:

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

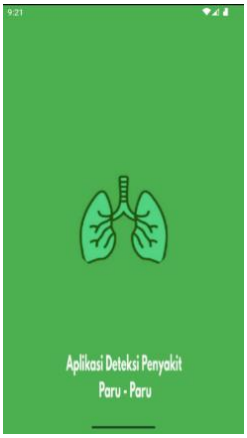

Figure 6. SplashScreen Page Display

b. Onboarding Page

Subsequent to the splash screen, novel users are presented with a series of introductory, or onboarding, pages. The purpose of these pages is to furnish pertinent information, such as an advisory that the application serves solely as a diagnostic support tool, and to introduce the principal features available. The visual representation of the Onboarding page is depicted as follows:

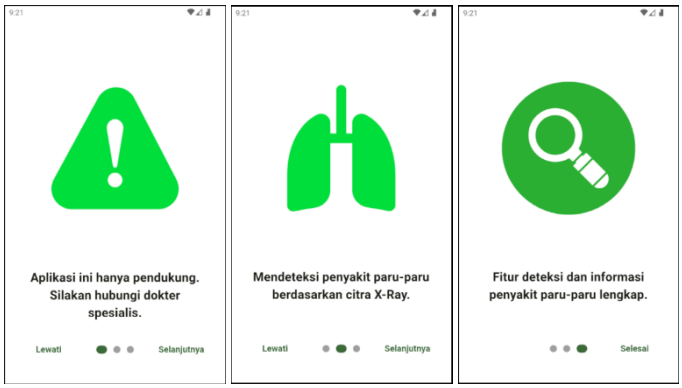

Figure 7. Onboarding Page Display

c. Main (Home) Page

The main page serves as a central navigation hub, providing access to two core functionalities: "Deteksi Penyakit Paru - Paru" to initiate classification processes, and "Informasi Penyakit Paru - Paru" to provide disease-related data. The visual representation of the Home page is depicted as follows:

Figure 8. Home Page Display

d. Disease Detection Page

On this page, users are afforded the capability to upload radiographic images from the device's gallery or capture images directly via the camera. Upon selection of an image, the user may engage the "Deteksi Penyakit" button to commence the classification process by the model. The visual representation of the Disease Detection page is depicted as follows:

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

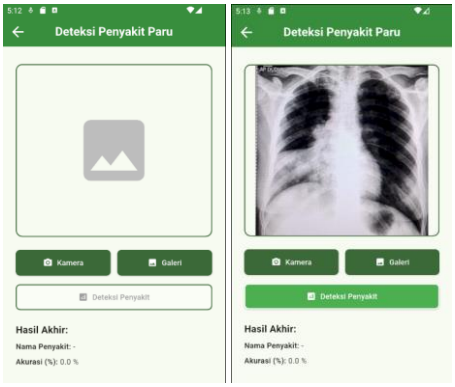

Figure 9. Disease Detection Page Display

e. Classification Result Page

Upon completion of the detection process, the application shall exhibit the results on this particular page. The information displayed thereby comprises the name of the detected ailment, coupled with the percentage representing the accuracy level of the model's prediction. The visual representation of the Classification Result page is depicted as follows:

Figure 10. Classification Result Page Display

f. Disease Information Page

This feature is designed to furnish users with educational content concerning the disease types detectable by the application. Upon user selection of the information menu, the application shall present a roster of selectable diseases, namely COVID-19, Pneumonia, and Tuberculosis. Subsequent to user selection of a particular disease, the application will transition to a detailed page exhibiting comprehensive information regarding said ailment, encompassing illustrative images, concise explanations, etiologies, and prevalent symptoms. The visual representation of the Disease Information page is depicted as follows:

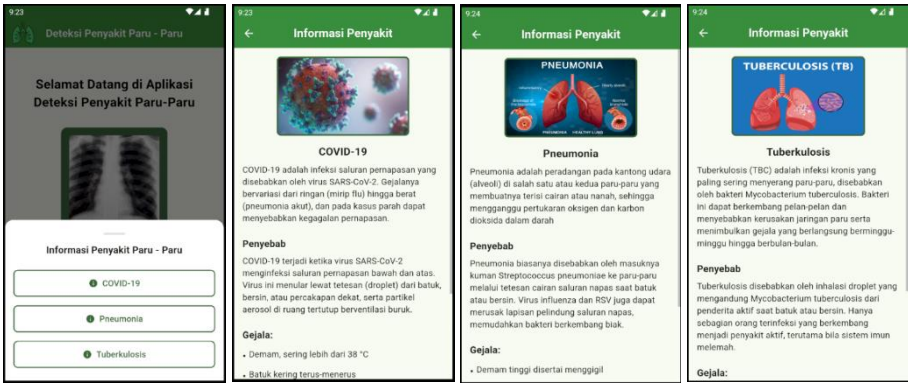

Figure 11. Disease Information Page Display

3.7 Mobile Application System Testing

Subsequent to the implementation of the application system, the ensuing phase involves system testing to ascertain the proper execution of each function in accordance with the design specifications. This testing is focused on the application's functionality from the user's perspective, employing the Black Box Testing methodology. Initial testing will encompass core functionalities, including fundamental application workflows such as image uploading, detection execution, and menu navigation. The outcomes of the functionality testing are summarized in Table 7 below.

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 12−22
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.131
https://ejurnal.pdsi.or.id/index.php/bids/index

Table 7. Application Functionality Testing Results

No	Test Case	Test Step	Result	Status
1	SplashScreen Display	User launches the application	The application displays the SplashScreen for a few seconds, then automatically transitions to the Onboarding page.	Success
2	Onboarding Navigation	On the first Onboarding page, the user presses "Selanjutnya"; on the second page, the user presses "Selanjutnya" again; on the last page, the user presses "Selesai" to enter the application	The application transitions to the second Onboarding page; the application transitions to the last Onboarding page; the application enters the Main Page (Home).	Success
3	Skip Onboarding	On the first or second Onboarding page, the user presses "Lewati".	Direct transition to the third onboarding page.	Success
4	Navigation to Detection Page	From the Main Page, the user presses the "Deteksi Penyakit Paru - Paru" button	The application transitions to the Disease Detection Page.	Success
5	Navigation to Disease Information	From the Main Page, the user presses the "Informasi Penyakit Paru - Paru" button	The application displays a dialog box with a list of disease options..	Success
6	Initial State of Detection Button	User enters the Detection Page without loading an image	The "Deteksi Penyakit" button is disabled (cannot be pressed).	Success
7	Upload Image from Gallery	User presses the "Galeri" button; the user selects one X-ray image from the device's gallery	The device's gallery opens; the selected image is successfully loaded, and the "Deteksi Penyakit" button becomes active.	Success
8	Take Photo with Camera	User presses the "Kamera" button; the user takes a new photo	The camera interface opens; the photo is successfully taken, and the "Deteksi Penyakit" button becomes active.	Success
9	Classification Process	After the image is loaded, the user presses the "Deteksi Penyakit" button	The application displays the results page with the disease name and confidence percentage.	Success
10	View Detail Information	User opens the information list dialog; the user selects one disease (e.g., "Pneumonia")	The application transitions to a detail page displaying specific information about Pneumonia,	Success

The outcomes from the functionality testing in Table 7 reveal that all test cases are marked with a status of 'Success'. These findings subsequently form the basis for the maintenance phase, where no major functional bugs requiring rectification were identified.

4. CONCLUSION

The research concludes that applying data balancing and augmentation techniques effectively improves the performance of the MobileNetV2 Convolutional Neural Network (CNN) in classifying four lung conditions: COVID-19, Normal, Pneumonia, and Tuberculosis. These methods successfully overcome issues related to uneven data distribution and the high visual similarity among disease classes, which often hinder model generalization. Experimental results demonstrate that using the Adam optimizer with a learning rate of 1e-3 provides the strongest baseline. However, the highest accuracy was achieved when incorporating a comprehensive augmentation process. Combining the Mixup method for class balancing with various geometric augmentations yielded the best results, reaching 96.62% accuracy and an average F1-Score of 96.63%. Visual analysis of the training dynamics further confirms that this strategy effectively acts as a regularizer, preventing overfitting and ensuring the model generalizes well to unseen data. Additionally, the optimized model was successfully converted into TensorFlow Lite format and integrated into a mobile application built with Flutter. Black Box Testing verified that all features, from image upload to result display, function properly. Despite its strong outcomes, this study is limited by dataset size, suggesting future work should involve larger, diverse datasets and clinical testing for practical validation.

REFERENCES

[1] I. O. Safitri and R. D. Srimarinda, "Penerapan Metode Support Vector Machine (SVM) dan Convolutional Neural Network (CNN) dengan Pendekatan Threshold Tuning untuk Klasifikasi Diagnosa Pneumonia pada Data Chest X-Ray Images Kata Kunci–

Pneumonia, Chest X-ray, Convolutional Neural Network (CNN), Support Vector Machine (SVM), Machine Learning," *INFERENSI*, vol. 5, pp. 2721–3862, 2024

[2] S. Andika Maulana, S. Husna Batubara, Y. Permata Putri Pasaribu, H. Syahputra, and F. Ramadhani, "Deteksi Burung Menggunakan Convolutional Neural Network (Cnn) Dengan Model Arsitektur Mobilenetv2," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 4, pp. 6108–6114, Jun. 2024, doi: 10.36040/jati.v8i4.10126.

[3] S. Bahri, R. Wajhillah, and M. F. Adiwisastra, "Diagnosa Tuberculosis Paru Berbasis Citra X-ray Menggunakan Convolutional Neural Network," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 6, no. 2, pp. 181–186, Jan. 2022, doi: 10.31294/ijcit.v6i2.11844.

[4] T. Karlita, E. M. Yuniarno, I. K. E. Purnama, and M. H. Purnomo, "Detection of COVID-19 on Chest X-Ray Images using Inverted Residuals Structure-Based Convolutional Neural Networks," in *2020 3rd International Conference on Information and Communications Technology, ICOIACT 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 371–376. doi: 10.1109/ICOIACT50329.2020.9332153.

[5] I. Md. D. Maysanjaya, "Klasifikasi Pneumonia pada Citra X-rays Paru-paru dengan Convolutional Neural Network," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 9, no. 2, pp. 190–195, May 2020, doi: 10.22146/jnteti.v9i2.66.

[6] H. D. Hekmatyar, W. A. Saputra, and C. Ramdani, "Klasifikasi Pneumonia Dengan Deep Learning Faster Region Convolutional Neural Network Arsitektur VGG16 dan ResNet50," *InComTech : Jurnal Telekomunikasi dan Komputer*, vol. 12, no. 3, p. 186, Dec. 2022, doi: 10.22441/incomtech.v12i3.15112.

[7] A. Tripathi, T. Singh, R. R. Nair, and P. Duraisamy, "Improving Early Detection and Classification of Lung Diseases With Innovative MobileNetV2 Framework," *IEEE Access*, vol. 12, pp. 116202–116217, 2024, doi: 10.1109/ACCESS.2024.3440577.

[8] F. A. A. Harahap, A. N. Nafisa, E. N. D. B. Purba, and N. A. Putri, "Implementasi Algoritma Convolutional Neural Network Arsitektur Model Mobilenetv2 Dalam Klasifikasi Penyakit Tumor Otak Glioma, Pituitary Dan Meningioma," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA)*, vol. 5, no. 1, pp. 53–61, Mar. 2023, doi: 10.29303/jtika.v5i1.234.

[9] Z. Gao, Y. Tian, S.-C. Lin, and J. Lin, "A CT Image Classification Network Framework for Lung Tumors Based on Pre-trained MobileNetV2 Model and Transfer learning, And Its Application and Market Analysis in the Medical field," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.04996

[10] A. Musa, M. I. Adamu, H. A. Kakudi, and Y. Lawal, "Nigeria Chest X-ray Dataset," 2024, *Kaggle*. doi: 10.34740/kaggle/dsv/9370352.

[11] S. Kumar, "COVID19+PNEUMONIA+NORMAL Chest X-Ray Image Dataset," 2021, *Kaggle*. doi: 10.34740/KAGGLE/DSV/1857760.

[12] S. Kumar *et al.*, "LiteCovidNet: A lightweight deep neural network model for detection of COVID-19 using X-ray images," *Int J Imaging Syst Technol*, vol. 32, no. 5, pp. 1464–1480, Sep. 2022, doi: 10.1002/ima.22770.

[13] S. Shastri, I. Kansal, S. Kumar, K. Singh, R. Popli, and V. Mansotra, "CheXImageNet: a novel architecture for accurate classification of Covid-19 with chest x-ray digital images using deep convolutional neural networks," *Health Technol (Berl)*, vol. 12, no. 1, pp. 193–204, Jan. 2022, doi: 10.1007/s12553-021-00630-x.

[14] S. Kiran and D. I. Jabeen, "Dataset of Tuberculosis Chest X-rays Images," 2024, *Mendeley Data*. doi: 10.17632/8j2g3csprk.2.

[15] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[16] M. R. R. Allaam and A. T. Wibowo, "Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network (CNN)," *Proceeding of Engineering*, vol. 8, no. 2, p. 1153, Apr. 2021, doi: 10.34818/eoe.v8i2.14708.

[17] M. N. Baay, A. N. Irfansyah, and M. Attamimi, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," *Jurnal Teknik ITS*, vol. 10, no. 1, Aug. 2021, doi: 10.12962/j23373539.v10i1.59790.

[18] A. Souid, N. Sakli, and H. Sakli, "Classification and predictions of lung diseases from chest x- rays using mobilenet v2," *Applied Sciences (Switzerland)*, vol. 11, no. 6, Mar. 2021, doi: 10.3390/app11062751.

[19] R. Santoso, "Augmentasi Data pada Prasasti Logam untuk Deteksi Aksara Kawi," *JURNAL FASILKOM*, vol. 14, no. 1, pp. 234–241, Apr. 2024, doi: 10.37859/jf.v14i1.6952.

[20] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J Imaging*, vol. 9, no. 2, Feb. 2023, doi: 10.3390/jimaging9020046.

[21] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2688.

[22] N. Nursobah, M. I. Saad, and J. A. J. Kansil, "Implementation of the Flutter Framework for Developing an E-Commerce Application," *TEPIAN*, vol. 5, no. 4, pp. 127–135, Dec. 2024, doi: 10.51967/tepian.v5i4.3110.

[23] S. Alden and B. N. Sari, "Implementasi Algoritma CNN Untuk Pemilahan Jenis Sampah Berbasis Android Dengan Metode CRISP-DM," *Jurnal Informatika*, vol. 10, no. 1, pp. 62–71, Mar. 2023, doi: 10.31294/inf.v10i1.14985.

[24] M. Aqil *et al.*, "Integration of smartphone technology for maize recognition," *IOP Conf Ser Earth Environ Sci*, vol. 911, no. 1, p. 012037, Nov. 2021, doi: 10.1088/1755-1315/911/1/012037.

[25] Y. H. Natbais and A. B. S. Umbu, "Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih Tensorflow Lite," *TEKNOTAN*, vol. 17, no. 2, p. 83, Aug. 2023, doi: 10.24198/jt.vol17n2.1.