Jurnal Informatika: Jurnal pengembangan IT

Vol. 10, No. 4, 2025
ISSN : 2477-5126, DOI:10.30591/jpit.v10ix4.9086

The Impact of Image Pre-processing for Tuberculosis Prediction System Based on Chest X-ray Images

Rudi Kurniawan[1], Kevin Ilham Apriandy[2], Tessy Badriyah [3], Iwan Syarif [4]

[1,3,4]Department of Informatics and Computer Engineering, Electronic Engineering Polytechnic Institute of Surabaya, Jl. Raya ITS, Keputih, Sukolilo, Surabaya, 60111, Indonesia

[2]Study Program of Diploma of Computer Technology, Politeknik Internasional Tamansiswa Mojokerto, Jl. Taman Siswa No.30, Purwotengah, Magersari, Kota Mojokerto, 61311, Indonesia

Info Artikel

Riwayat Artikel:

Received 2025-06-24
Revised 2025-08-05
Accepted 2025-08-06

Corresponding Author:

Rudi Kurniawan

Email: fmrudiawan@gmail.com

Abstract *– With the rapid development of automated detection system using deep learning techniques on Chest X-ray (CXR) image datasets to the subjective assessment performed by healthcare professionals. Preprocessing is critical in medical image analysis as it helps highlight important anatomical features while suppressing irrelevant information, thus enabling the model to focus on meaningful patterns. In this paper, we investigate the impact of image preprocessing techniques on the performance of a tuberculosis prediction system based on CXR images using a deep learning approach. We used the "Tuberculosis Chest X-rays (Shenzhen)" dataset, which contains 1,344 CXR images (672 TB cases and 672 normal cases). We propose a five-step preprocessing pipeline consisting of resizing, heavy sharpen filtering, CLAHE (Contrast Limited Adaptive Histogram Equalization), horizontal flip augmentation, and data normalization. The findings indicate that the model utilising preprocessing markedly surpasses the one lacking it, attaining an accuracy, precision, recall, and F1-score of 71%, in contrast to 51%, 50%, 50%, and 36% without preprocessing, respectively. This study enhances the existing research on the application of deep learning in medical diagnostics and emphasises the significance of preprocessing for attaining dependable, high-performance systems.*

Keywords: *Convolutional Neural Networks; Computer-Aided Diagnosis; Convolutional Neural Networks; Deep Learning; Image Pre-processing; Tuberculosis.*

Abstrak *– Dengan pesatnya perkembangan sistem deteksi otomatis menggunakan teknik deep learning pada kumpulan data citra Chest X-ray (CXR) hingga penilaian subjektif yang dilakukan oleh tenaga kesehatan. Prapemrosesan sangat penting dalam analisis citra medis karena membantu menyoroti fitur anatomi penting sekaligus menghilangkan informasi yang tidak relevan, sehingga memungkinkan model untuk fokus pada pola yang bermakna. Dalam makalah ini, kami menyelidiki dampak teknik prapemrosesan citra pada kinerja sistem prediksi tuberkulosis berdasarkan citra CXR menggunakan pendekatan pembelajaran mendalam. Kami menggunakan dataset "Tuberculosis Chest X-rays (Shenzhen)" yang berisi 1,344 citra CXR (672 kasus TB dan 672 normal). Kami mengusulkan alur kerja prapemrosesan lima langkah yang terdiri dari pengubahan ukuran, penyaringan penajaman berat, CLAHE (Contrast Limited Adaptive Histogram Equalization), augmentasi flip horizontal, dan normalisasi data. Hasilnya menunjukkan bahwa model dengan prapemrosesan secara signifikan mengungguli model tanpa prapemrosesan, mencapai akurasi, presisi, perolehan kembali, dan skor F1 sebesar 71%, dibandingkan dengan 51%, 50%, 50%, dan 36% tanpa prapemrosesan. Studi ini berkontribusi pada semakin berkembangnya penelitian tentang penerapan pembelajaran mendalam untuk diagnostik medis dan menggarisbawahi pentingnya praproses dalam mencapai sistem yang andal dan berkinerja tinggi.*

Kata Kunci*: Convolutional Neural Networks, Diagnosis Berbantuan Komputer, Deep Learning, Prapemrosesan Gambar Tuberculosis.*

I. INTRODUCTION

Tuberculosis (TB) is an infectious illness caused by Mycobacterium, primarily affecting the human lungs [1]. Data from the World Health Organization (WHO) indicates that by 2022, TB will be classified as the second most severe worldwide infectious illness following COVID-19 [2]. In diagnosing patients suspected of having TB, doctors analyze medical history examination data and observe chest X-ray (CXR) images to find indications of infection [3].

In recent years, automated detection system using deep learning (DL) techniques on CXR image datasets have grown explosively to address the expertise and access gaps in the global effort for early TB screening [4]. It is expected that the earlier TB is detected, the quicker it can be treated, thereby reducing the risk of further infection [5]. Deep learning techniques are widely used to address various crucial problems [6], especially disease detection based on CXR images that utilize the learning results from Convolutional Neural Networks (CNN).

The development of accurate Computer-Aided Diagnosis (CAD) systems for medical purposes is becoming increasingly popular due to the exploitation of DL techniques and CNN-based architectures for image analysis applications [7]. CAD systems, especially for detecting tuberculosis based on CXR images, require a data set for the learning process. However, it is often the case that some images exhibit noise that can impact the performance of the CAD system, so to overcome this, preprocessing steps such as transformation, normalization, enhancement, noise remover, or filtering are required [8].

Unlike the two previous studies, which only applied simple preprocessing, this study proposes a more comprehensive preprocessing pipeline for chest X-ray (CXR) images for tuberculosis classification. Aldiansyah and Soleh only used basic techniques such as resizing, grayscale conversion, and data augmentation [9]. At the same time, Sastramandala et al. performed dataset splitting, image scaling to a resolution of 128×128 pixels, and general image augmentation [10]. In contrast, the pipeline proposed in this study includes five complementary stages: resizing to a resolution of 224×224 pixels, applying a strong sharpening filter (heavy sharpen), enhancing contrast with CLAHE, pixel normalization, and horizontal flip augmentation. This sequence of steps is designed to emphasize important anatomical features in the images without losing clinical information, rather than simply increasing the training data through augmentation or performing simple resizing as in previous approaches.

In this paper, we propose a preprocessing step for CXR images to support the performance of a CAD system in detecting infection in patients suspected of having TB. Anatomical structures such as the lungs, ribs, and diaphragm can have low contrast in CXR images, so image quality enhancement is needed to make it easier for the model to indicate TB by referring to critical visible features.

Several previous studies have attempted preprocessing, but often with limited steps. For example, Aldiansyah and Soleh applied only basic techniques such as resizing, grayscale conversion, and data augmentation [9]. Similarly, Sastramandala et al. performed dataset splitting, image scaling to 128×128 pixels, and general augmentation [10]. While these approaches improved model training, they remained relatively simple and may not sufficiently emphasize critical anatomical structures needed for accurate TB classification. To address this gap, this study proposes a more comprehensive preprocessing pipeline for CXR images consisting of five stages: resizing to 224×224 pixels, applying a heavy sharpening filter, enhancing contrast with CLAHE, pixel normalization, and horizontal flip augmentation. This sequence is designed to emphasize important anatomical features such as lungs, ribs, and diaphragm without losing clinical information.

This paper can enhance the development of more precise tuberculosis detection models by emphasizing critical elements in chest X-ray pictures, including pattern identification indicative of tuberculosis through preprocessing techniques. This work enhances awareness among researchers regarding the critical role of the preprocessing stage in developing deep learning-based detection systems by illustrating its substantial effect on model performance.

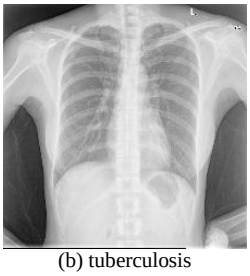
II. RESEARCH METHODOLOGY

This research proposes a thorough and practical image preprocessing strategy to enhance the efficacy of CXR-based tuberculosis detection systems. This study contrasts with prior research that typically employs minimal preprocessing techniques, such as padding and resizing, by introducing four primary preprocessing steps: resizing, a heavy sharpen filter, CLAHE, and horizontal flip augmentation, which have demonstrated enhancements in the model's stability and accuracy in tuberculosis detection. This strategy is distinctive because of its use of a combination of strategies aimed at enhancing picture feature quality and diversifying training data while preserving pertinent clinical information, which significantly influences the performance of the DenseNet-169-based model.

This research also presents an original contribution through the modification of the fully connected layer in the DenseNet-169 architecture utilised for feature extraction. The alterations, along with the implementation of GlobalMaxPooling2D, BatchNormalization, and Dropout, are especially tailored to optimise the model for the TB classification job, therefore mitigating overfitting and enhancing the model's generalizability [11]. This approach creates a more robust model for detecting patterns indicative of tuberculosis in CXR images [12], making this research significantly contribute to developing deep learning-based detection systems for tuberculosis.

We used a publicly shared dataset through the Kaggle website titled “Tuberculosis Chest X-rays (Shenzhen)” . There are two types of data in the data set: comma-separated values (CSV) files and folders containing CXR images. The folder containing the CXR images requires 4 gigabytes (GB) of space, as each raw image has 2344 x 2482 pixels.

(a) normal



(b) tuberculosis

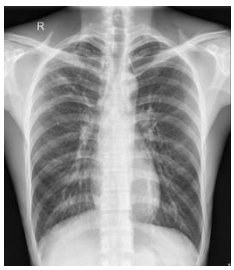
Figure 1. Sample of Tuberculosis CXR with (a) normal and (b) tuberculosis condition

This paper's four main sequential stages, which include exploratory data analysis, image preprocessing, model training, and model evaluation, have different definitions and objectives.

Figure 2. The Research Stages

Exploratory Data Analysis (EDA) is a statistical approach to observing the hidden patterns present in a data set by representing them in numerical, tabular, and visual descriptions [13]. There are three types of EDA, including univariate, bivariate, and multivariate. However, in this paper, we only use univariate and bivariate EDA to examine patient self-data with CXR images in the data set. Univariate EDA analyses one variable to understand the distribution of counts on that variable to obtain specific data set characteristics [14]. We used pie charts to visualize the data distribution of the variables of gender and diagnosis, which are categorical data. In addition, we used box plots to visualize numerical data related to the age variable and then combined the analysis with descriptive statistics. Meanwhile, Bivariate EDA is the process of analyzing the relationship between two variables in a data set to design better experiments by selecting relevant pairs of variables to test [14]. We used cross-tabulation to display the frequencies for each combination of categorical data in tabular form. We applied this cross-tabulation to the variables of gender and diagnosis.

Image pre-processing plays an essential role in developing CAD systems that ensure the accuracy and reliability of medical image analysis by providing detailed soft tissue visualization [15]. In addition, image preprocessing is necessary to reduce redundant features that do not affect decision-making in neural networks, thus making training converge faster [16]. In this scientific work, there are five steps in processing the raw image data set to make it ready to be processed by the model, each with a different purpose.



(a) Original Image

(b) Pre-processed Image

Figure 3. The comparison of original and pre-processed image

Image resizing is an operator used to resize an image by retaining important features to save fewer artifacts on advanced processing [17]. We significantly resized the image from 2344 x 2482 pixels to 224x224 pixels to let the model focus on more important and common features instead of small irrelevant details. Too large an image would make the model too complex to learn every detail, which could lead to overfitting.

A heavy sharpening filter is an image processing technique that enhances sharpness and detail in images [18], as shown in (1). In the CXR context, a heavy sharpening filter can help accentuate essential features, such as sharpening tissue fibre lines. This process is done because the CXRs in this dataset have low contrast, so the intensity difference between bone and other tissues needs to be clarified.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (1)$$

Although the image dataset has already undergone some image processing, it still needs quality enhancement using CLAHE to handle backgrounds with high homogeneity [19]. CLAHE is an image quality enhancement method that can provide clip boundaries for colour histograms and region sizes [20]. CLAHE has been widely used to process medical images, such as X-Ray and CT scans, because it can improve image contrast and display fine details that may be missed by the human eye [21].

Image augmentation is one of the practical training strategies without including additional images to expand the data set processed by neural network models [22]. Using horizontal flips in the image augmentation process in a tuberculosis detection system on X-ray images helps increase the variety of training data without changing the relevant clinical information. Tuberculosis often affects both lungs, so flipping the image horizontally maintains a reasonable anatomical orientation. However, the vertical flip is not used because it will flip the image from top to bottom, disturb critical anatomical structures such as the position of the heart and diaphragm, and may cause the model to detect invalid or misleading features. Thus, horizontal flip can increase the model's robustness to variation, while vertical flip can corrupt important information required for accurate detection.

TABLE 1
THE TOTAL DIFFERENCE OF IMAGE DATASET

Condition	Original	Pre-processed
Normal	326	652
Tuberculosis	336	672

Data normalization on image data that has been read its Read Green Blue (RGB) values in the form of an array changes the range of pixel values in the image to be within a particular scale [23], usually between 0 and 1, such as Min-Max Scaler formula shown in (2). The primary purpose of this normalization is to ensure that each pixel contributes equally to the training process of the machine learning model without any value dominating. Normalization also helps accelerate model convergence during training, reduces the risk of gradient vanishing or exploding, and improves model stability and performance.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

Before entering the model training process, the dataset is divided into three subsets: training (80%), validation (10%), and test (10%), as shown in Table 2. This division is a stratified hold-out, meaning that the proportions of positive and negative classes are maintained consistently across all three subsets. The training set (n=1071) is used for updating the model weights. In contrast, the validation set (n=133) is used to monitor performance during training—particularly in hyperparameter selection (e.g., learning rate, number of epochs) and the application of early stopping to prevent overfitting. The test set (n=120) is only opened after the training process is complete, providing an estimate of the model's generalization on truly new data.

TABLE 2
THE PROPORTION OF DATA SPLITTING

Type	Percentage	Total
Training	80%	1071
Validation	10%	133
Test	10%	120

Although k-fold cross-validation (e.g., 5- or 10-fold) is often used to obtain a more robust measure of performance variability, we chose a single 80–10–10 split scheme with stratification because: (1) the dataset size is relatively small, so each fold would be very similar and not add much new information, (2) the computational budget is limited due to the preprocessing pipeline and the size of the DenseNet-169 model, and (3) there is a need to establish a fixed test set to enable consistent experimental comparisons. To ensure the reliability of the results, each experiment is repeated several times with different seeds, and the performance variance on the validation set is analyzed to ensure model stability.

In this paper, model training is the process by which a machine learning model is trained to recognise and classify patterns in images that indicate the presence of tuberculosis. This process involves using a dataset of labelled chest X-rays, where the images are classified as containing signs of tuberculosis or not. Machine learning

models, such as convolutional neural networks (CNNs), learn from this data by adjusting their internal weights to minimize prediction error [24]. During training, the model evaluates and refines its predictions through successive iterations until it can accurately detect signs of tuberculosis in new X-ray images that were never seen before. The result is a system that can assist in the automated clinical diagnosis of tuberculosis.

Using pre-trained models to build a tuberculosis detection system on CXR images is a strategy that utilizes machine learning models pre-trained on large datasets, such as ImageNet, which includes millions of images and thousands of object categories. These models already have weights learned from standard image features, such as edges, texture, and shape [25]. When applied to the tuberculosis detection task, this pre-trained model does not start from scratch but is customized (through fine-tuning) to a more specific CXR dataset. Thus, the model can achieve better performance with less data faster, as it already understands the visual features.

In this paper, we use a pre-trained model architecture called DenseNet169. DenseNet169 is a CNN architecture belonging to the DenseNet family, designed to improve efficiency and performance in image recognition tasks. DenseNet169 has 169 densely connected layers, meaning that each layer is directly connected to all subsequent layers. Thus, information and gradient propagation can reduce problems such as vanishing gradients often occurring in intense networks [26].

TABLE 3
THE PARAMETER TOTAL OF MODEL

Type	Total
Trainable	4,993
Non-trainable	12,646,208

We created a tuberculosis detection system using DenseNet-169 with modifications to the fully connected layer. The DenseNet-169 architecture is used as the backbone for feature extraction from X-ray images. To adapt to DenseNet-169, which has dense neural networks and shortcut connections, the fully connected layer at the end of the DenseNet-169 model is modified to match the desired number of classes.

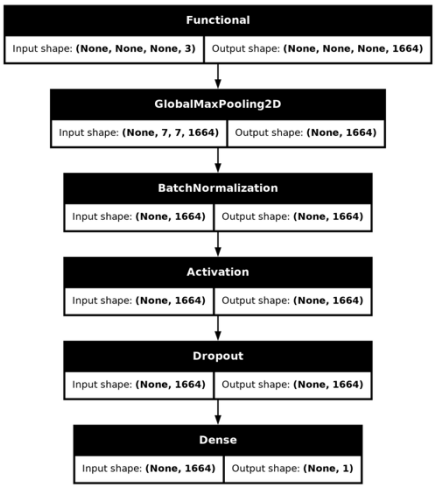


Figure 4. The architecture of the proposed pre-trained model

The model uses the DenseNet-169 architecture as the backbone for feature extraction from X-ray images, with modifications to the fully connected part to suit the classification task. Once the features are extracted, the model applies a GlobalMaxPooling2D layer to reduce the spatial dimension, followed by BatchNormalization to stabilize the activation distribution, and an Activation layer to add non-linearity. Next, Dropout prevents overfitting by randomly deactivating some neurons during training. Finally, the modified Dense layer produces an output with one neuron for binary classification (Tuberculosis or not).

Hyperparameters in CNN models are parameters that are not learned during the training process but are set before training begins to control how the model is trained and configured [27]. These hyperparameters include various aspects, such as loss function, optimizer, learning rate, and number of epochs. Setting the correct hyperparameters is critical to the model's performance, as they affect how well the model can learn from the data and generalize to unseen data.

TABLE 4
HYPERPARAMETER SETTING

Hyperparameter	Value
Loss Function	Binary Crossentropy
Optimizer	Adam
Learning Rate	0.99
Epoch	100
Batch Size	128
Weights	ImageNet

Several important hyperparameters are used in the tuberculosis detection system using CNN from X-ray images. Binary cross-entropy was chosen as the loss function because it is a binary classification problem where the model must predict the presence or absence of tuberculosis. Adam optimizer was used for its ability to automatically adjust the learning rate during training, making it efficient and stable for complex networks. A learning rate of 0.99 was initially selected to speed up the convergence of the model. Although this value is relatively high and could cause instability or slow convergence at the beginning of the training process, it was chosen after experimenting with smaller rates to find an optimal balance between training speed and accuracy. The high learning rate is intended to make larger updates to the model weights in the early stages of training, which can be beneficial in overcoming local minima and speeding up convergence when the model is far from the optimal solution. However, given the potential risk of the model failing to converge initially, stabilization techniques such as learning rate decay or adaptive learning rate adjustments through the Adam optimizer were employed to ensure the model could stabilize and converge effectively over time. The model was trained for 100 epochs to facilitate the acquisition of the dataset's features. A batch size of 128 was selected to concurrently process several images, enhancing training efficiency while minimising memory usage. The model's initial weights were established using pre-trained weights from ImageNet, enabling the utilisation of previously acquired features from standard datasets and accelerating the training process via transfer learning.

Furthermore, we employ callbacks to execute the model training procedure for predicting the label of each input CXR. Callbacks can be applied during training to monitor, control, or modify the training process. Callbacks are used to automatically perform various tasks, such as saving the model when the best performance is achieved and adjusting the learning rate dynamically. Using callbacks can make model training more efficient and stable, allowing users to optimize performance without manually monitoring the training process.

We use callbacks named Checkpoints during the model learning process, a technique used to save model weights periodically during the training process [27]. Callbacks are useful to prevent loss of training, which results in interruptions, such as system failure or a very long training process. Checkpoints are used to save the model at a particular stage when it achieves the best performance (e.g., based on a validation metric) and then reload it for use without having to retrain the model from scratch, thus avoiding overfitting by selecting weights from the best epoch, rather than from the final model which may already be overfitted.

TABLE 5
CHECKPOINT PARAMETER SETTING

Parameter	Value
monitor	val_loss
mode	min
save_best_only	True

In addition to checkpoints, we also use a callback called ReduceLROnPlateau, which automatically reduces the learning rate when the model performance stops improving during training [28]. When a monitored metric, such as the loss in the validation set, does not improve in a certain number of consecutive epochs, this callback will reduce the learning rate by a particular factor. This reduction in learning rate is essential because it allows the model to make finer and more precise adjustments to the weights, especially as the model approaches convergence, thus preventing the model from getting stuck in local minima and improving the overall accuracy of the model.

TABLE 6
REDUCELRONPLATEAU PARAMETER SETTING

Parameter	Value
monitor	val_loss
factor	0.99
min_lr	1e-10
patience	2

Model evaluation in building a tuberculosis detection system on chest X-ray images is the process of measuring the model's performance after training, ensuring that the model can detect tuberculosis accurately and reliably. This evaluation process uses metrics such as accuracy, precision, recall, and F1-score to assess how well the model classifies X-ray images as positive or negative. The evaluation is performed on test data not used during training, representing new X-ray images that may be encountered in actual clinical situations. Thus, model evaluation helps to ensure that the built detection system performs well on training data and reliably when used on data that has never been seen before [28].

Accuracy indicates how often the system makes a correct prediction [29], whether it predicts the patient has TB (positive) or does not have TB (negative) out of all predictions made. For example, if the system examines 200 patients, of which 150 predictions are correct (both TB and non-TB) and 50 predictions are incorrect, then the system's accuracy is 75%. Accuracy measures the total correctness of the predictions.

$$Accuracy = \frac{True\ Positive + True\ False}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \quad (3)$$

Precision indicates how accurate the system is in predicting which patients have TB out of all those predicted to be positive by the system. For example, if the system predicts 100 patients to have TB, but only 80 are positive for TB, then the precision is 80%. Precision is essential to reduce False Positives, which are errors predicting healthy patients as having TB, which can lead to unnecessary treatment or anxiety for the patient [29].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4)$$

The recall measures how well the system detects all TB patients [29]. For example, if 100 patients have TB, and the system successfully predicts 90 of them correctly (True Positive) but fails to detect ten patients (False Negative), then the recall is 90%. The recall is necessary when we want to minimize cases of False Negative, which is when the system does not detect TB patients, as it can be fatal if the patient does not get the proper treatment immediately [29].

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

F1-score is a metric that combines precision and recall to give a balanced picture of system performance, especially when there is an imbalance between positive and negative predictions [29]. For example, suppose the system has a precision of 80% (meaning that of all those predicted for TB, 80% are correct) and a recall of 90% (meaning that 90% of all TB patients are successfully detected). In that case, the F1-score will provide a harmonious average of the two.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

In the tuberculosis detection system evaluation model with nearly balanced classes, choosing the macro average is more appropriate than the weighted or micro average because the macro average calculates metrics (such as precision, recall, and F1-score) for each class separately and then takes the average of these values without considering the class size [30]. Model evaluation provides a clearer understanding of the model's performance in each class [31]. In problems with nearly balanced classes, the macro average can evaluate the model's performance fairly without the dominant class influencing the results too much, as may be the case with a weighted average [32]. In comparison, micro-averaging combines all decisions before calculating the metrics, which can obscure performance differences between classes. With macro averaging, we can ensure that the model performs proportionally on both tuberculosis and non-tuberculosis classes.

III. RESULT AND DISCUSSION

This study's dataset consists of two classes, namely tuberculosis and normal. Based on Figure 5, the dataset has a relatively balanced distribution between the two classes. The tuberculosis class covers 50.8% of the data, while the normal class covers 49.2%. The dataset's almost balanced distribution is significant in maintaining the prediction model's performance so that it is not affected by class imbalance that can cause bias in prediction towards the majority class.

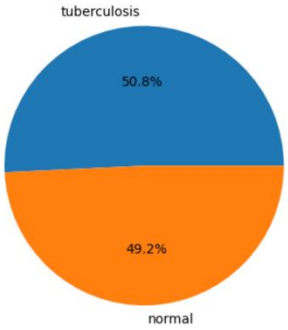


Figure 5. The distribution of class label

Based on Figure 6, the gender distribution in the dataset shows that the majority of patients suspected of having tuberculosis are male. 69.5% of the data are male patients, while the other 30.5% are female patients. Although there is an imbalance in gender distribution, this does not significantly affect the development of the CXR-based tuberculosis detection system. The prediction system developed focuses more on the visual patterns seen from the CXR so that demographic factors such as gender do not affect the accuracy or performance of the system in detecting the presence of tuberculosis.

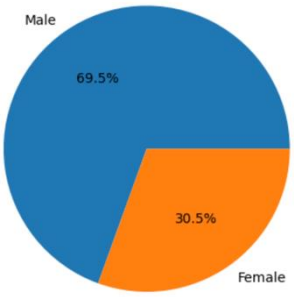


Figure 6. The distribution of gender patients

Figure 7 shows the age distribution of patients with CXRs suspected of having tuberculosis. The distribution is asymmetric, with a peak in the age range of 20 to 30 years. This age group dominates the dataset, with over 120 patients at around 20 years of age. After 30, the number of patients declines gradually, although there are still patients up to 80. The youngest age range in the dataset is around 0 to 10 years old, but with a minimal number of patients.

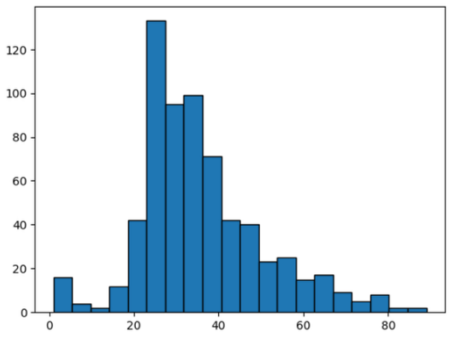


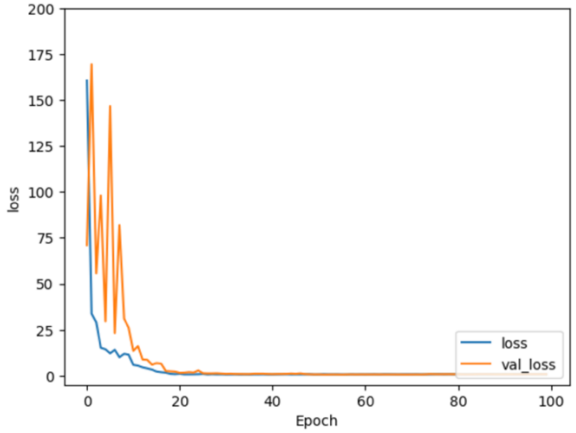
Figure 7. The distribution of gender patients

This pivot table below shows the distribution of the number of normal and tuberculosis cases by gender. In the female group, 101 patients with normal X-rays and 101 patients diagnosed with tuberculosis showed an equal distribution. Meanwhile, in the male group, there were 225 patients with normal X-ray results and 235 patients diagnosed with tuberculosis, indicating slightly more tuberculosis cases than normal results. This table provides an overview of the differences in the number of cases by gender in the dataset used.

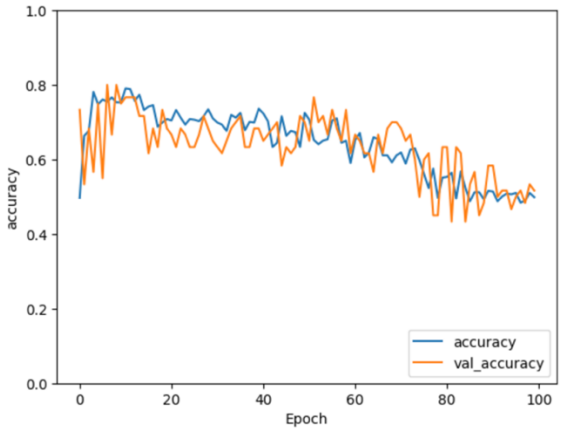
TABLE 7
PIVOT TABEL

Gender	Label	
	Normal	Tuberculosis
Female	101	101
Male	225	235

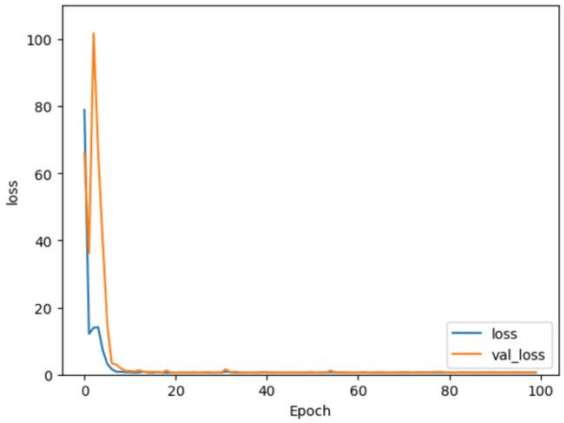
Figure 5 compares the loss and accuracy curves of model training with and without the application of preprocessing on the X-ray dataset for tuberculosis detection using the modified model based on the DenseNet-169 architecture.



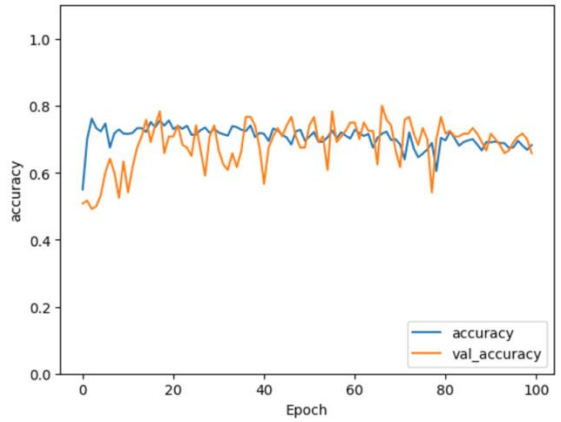
(a) Loss curve for training model without preprocessing dataset



(b) Accuracy curve for training model without preprocessing dataset



(c) Loss curve for training model with preprocessing dataset



(d) Accuracy curve for training model with preprocessing dataset

Figure 8. The impact difference between dataset with and without preprocessing to the learning curve accuracy and loss obtained by modified fully-connected in pre-trained DenseNet-169 model

In sub-figure (a), the loss curve of the model trained without preprocessing on the dataset is shown. Training and validation losses decrease rapidly during the first 20 epochs and stabilize near 0 as the number of epochs increases. This shows that the model learns well, although there is a slight fluctuation in the validation loss in the first few epochs.

Sub-figure (b) displays the accuracy curve of the same model. It can be seen that the model's accuracy increases sharply during the first ten epochs and reaches about 71%. However, there is a more considerable fluctuation in the validation accuracy compared to the training accuracy. This may indicate that the model slightly overfits the training data due to the less stable accuracy in the validation data.

After preprocessing was applied, the loss curve (sub-figure c) showed a faster and more consistent decline compared to the model without preprocessing (sub-figure a). Technically, the CLAHE stage enhances local contrast in areas with low intensity differences—such as small lesions or fiber lines—so that the initial filter in DenseNet-169 can extract abnormal lung tissue features more sharply. Similarly, the heavy sharpen filter emphasizes edges and fine textures on the CXR, helping the model “see” the boundaries of lesions that were previously obscured by noise or low contrast. This combination explains why the validation loss in the preprocessed model is more stable and less prone to overfitting.

On the accuracy curve (sub-figure d), fluctuations in validation accuracy also decrease, and the accuracy value remains around 71% throughout training. Horizontal flip augmentation and pixel normalization (0–1) add

data diversity and scale the input value range, making gradient descent more stable and the model less dependent on specific features of a single orientation or intensity. As a result, model generalization improves; unseen test data can be classified more consistently.

The table below compares model performance for CXR-based tuberculosis detection using four main metrics: accuracy, precision, recall, and F1-Score. This comparison is made between dataset preprocessing models and models that do not. The results of this comparison show a significant impact of preprocessing on the performance of the prediction system.

TABLE 8
THE COMPARISON OF PERFORMANCE MODEL BETWEEN ORIGINAL AND PRE-PROCESSED IMAGE DATASET

Metrics	Model with Original Dataset	Model with Pre-processed Dataset
Accuracy	0.51	0.71
Precision	0.50	0.72
Recall	0.50	0.71
F1 Score	0.36	0.71

Firstly, in terms of accuracy, the model using preprocessing achieved a value of 0.71, much higher than the accuracy of the model without preprocessing, which was only 0.51. This enhancement demonstrates that preprocessing aids the model in classifying CXRs with greater accuracy, indicating that the forecast outcomes are closer to the actual findings.

Moreover, the model utilising preprocessing had a precision of 0.72, but the model lacking preprocessing reached just 0.50. The enhanced precision indicates that the model utilising preprocessing is more effective at predicting tuberculosis patients. This model generates a reduced number of false positive prediction mistakes.

The model with preprocessing exhibited enhanced recall, achieving a value of 0.71, in contrast to 0.50 for the model without preprocessing. A greater recall value signifies that the preprocessed model is more effective at identifying positive tuberculosis cases, hence diminishing the likelihood of false negatives.

The F1-Score, which integrates precision and recall into a singular metric, yielded a value of 0.71 for the model with preprocessing. Conversely, the model without preprocessing attained a value of 0.36. This signifies that the preprocessing model is both accurate and precise in identifying tuberculosis patients and exhibits greater balance in its predictions.

The metric comparison table reinforces these findings. Precision increased from 0.50 to 0.72—an improvement of nearly 44%—because the denoised and contrast-enhanced data reduced false positives: normal structures that resemble infections are no longer mistakenly detected. Recall jumped from 0.50 to 0.71 because small and subtle features, such as nodules or consolidation lines, became more prominent through CLAHE and sharpen filters, enabling the model to capture more positive cases and reduce false negatives. The F1-Score improved dramatically from 0.36 to 0.71, reflecting a much better balance between precision and recall thanks to the preprocessing pipeline that enhances diagnostic signal quality.

Overall, data preprocessing significantly improved the model's performance. Of all the metrics measured, the preprocessing model showed the greatest improvement, making it a more practical choice in developing a CXR-based tuberculosis prediction system.

IV. CONCLUSION

Based on the analysis, it can be concluded that the image preprocessing stage plays an important role in improving the performance of the model for detecting tuberculosis based on CXR. Image preprocessing helps to improve image quality by accentuating important features related to tuberculosis indication patterns, such as lungs, ribs, and diaphragm. Techniques including resizing, sharpen filter, CLAHE, horizontal flipping, and data normalisation via min-max scaling markedly enhanced the model's predictive outcomes.

The analysis of the loss and accuracy curves indicates that the model trained on the preprocessed dataset exhibits superior performance stability and diminished risk of overfitting, as evidenced by the more consistent validation loss and validation accuracy curves relative to the model lacking preprocessing. Furthermore, regarding performance criteria such as accuracy, precision, recall, and F1-Score, models that use preprocessing have significantly superior outcomes across all metrics compared to those that do not.

Data preparation enhances image quality and significantly influences the model's capacity to categorise tuberculosis cases with more accuracy, precision, and consistency. This underscores the significance of the preprocessing phase in the creation of deep learning-based illness detection systems, such as for tuberculosis, rendering it an indispensable step for enhancing the model's overall performance.

Subsequent research may enhance this pipeline by investigating its applicability to additional medical imaging tasks, such as the identification of pneumonia, COVID-19, or cancer through CT scans and MRI pictures. Subsequent research may explore the effects of sophisticated preprocessing strategies, such as multi-scale preprocessing or the integration of domain-specific augmentation methods, to enhance the generalisation

capacity of models across varied medical datasets. The incorporation of this preprocessing pipeline into real-time diagnostic systems is a possible avenue for enhancing clinical decision-making tools in diverse healthcare settings.

ACKNOWLEDGEMENT

Our deepest gratitude is expressed to Mrs. Tessy Badriyah, Ph.D and Prof. Iwan Syarif as the supervising lecturers, for their guidance, support, and valuable input during the preparation of this research. We would also like to thank all lecturers and staff of the Postgraduate Program in Applied Informatics and Computer Engineering PENS who have provided knowledge and facilities that support the completion of this scientific work. It is anticipated that the findings of this research will significantly contribute to the advancement of science in this domain.

REFERENCE

[1] S. A. Rima, M. Zannat, S. S. Haque, A. Kawsar, S. S. Jennifer, and A. W. Reza, "Tuberculosis Bacteria Detection Using Deep Learning Techniques," in *Lecture notes in networks and systems*, 2023, pp. 297–308. doi: 10.1007/978-981-99-7093-3_20.

[2] A. Iqbal, M. Usman, and Z. Ahmed, "Tuberculosis chest X-ray detection using CNN-based hybrid segmentation and classification approach," *Biomedical Signal Processing and Control*, vol. 84, p. 104667, Mar. 2023, doi: 10.1016/j.bspc.2023.104667.

[3] D. Susilo and N. Wahyono, "An Analysis of Image Enhancement Effects on Convolutional Neural Network-based Pulmonary Tuberculosis Detection," *E3S Web of Conferences*, vol. 465, p. 02054, Jan. 2023, doi: 10.1051/e3sconf/202346502054.

[4] K. Santosh, S. Allu, S. Rajaraman, and S. Antani, "Advances in Deep Learning for Tuberculosis Screening using Chest X-rays: The Last 5 Years Review," *Journal of Medical Systems*, vol. 46, no. 11, Oct. 2022, doi: 10.1007/s10916-022-01870-8.

[5] S. Kazemzadeh et al., "Deep learning detection of active pulmonary tuberculosis at chest radiography matched the clinical performance of radiologists," *Radiology*, vol. 306, no. 1, pp. 124–137, Sep. 2022, doi: 10.1148/radiol.212213.

[6] M. S. Ahmed et al., "Joint Diagnosis of Pneumonia, COVID-19, and Tuberculosis from Chest X-ray Images: A Deep Learning Approach," *Diagnostics*, vol. 13, no. 15, p. 2562, Aug. 2023, doi: 10.3390/diagnostics13152562.

[7] T. Micaraseth, S. Shuangshoti, K. Chaiprabha, N. Wanpiyarat, P. Chantranuwatana, and G. Phanomchoeng, "Comprehensive framework for tuberculosis detection using deep learning and image processing in Whole-Slide Images," *IEEE Access*, p. 1, Jan. 2025, doi: 10.1109/access.2025.3568073.

[8] B. S. Negara and I. Syurfi, "Covid-19 Detection on Chest X-ray Images Using Semi-supervised Deep Learning with MixMatch and CLAHE," *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pp. 728–734, Nov. 2024, doi: 10.1109/icimcis63449.2024.10956872.

[9] D. R. Aldiansyah and M. Soleh, "Analisa performa arsitektur model convolutional neural network dengan variasi jumlah hidden layer untuk klasifikasi tuberculosis pada citra X-Ray," *JURNAL FASILKOM*, vol. 14, no. 3, pp. 729–734, Dec. 2024, doi: 10.37859/jf.v14i3.7949.

[10] M. B. Sastramandala, R. N. Sari, and N. Novrina, "Pemanfaatan Deep Learning menggunakan Convolutional Neural Network Terhadap Penyakit Tuberkulosis melalui Citra Rontgen," *Jurnal Minfo Polgan*, vol. 14, no. 1, pp. 631–642, May 2025, doi: 10.33395/jmp.v14i1.14818.

[11] J. D. K. H, "Implementation and Efficient Analysis of Preprocessing Techniques in Deep Learning for Image Classification," *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 20, Aug. 2023, doi: 10.2174/1573405620666230829150157.

[12] A. Khazalah et al., "Image processing identification for Sapodilla using convolution Neural network (CNN) and transfer learning techniques," in *Studies in computational intelligence*, 2022, pp. 107–127. doi: 10.1007/978-3-031-17576-3_5.

[13] E. Kotei and R. Thirunavukarasu, "Tuberculosis detection from chest X-Ray image modalities based on transformer and convolutional neural network," *IEEE Access*, vol. 12, pp. 97417–97427, Jan. 2024, doi: 10.1109/access.2024.3428446.

[14] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, p. 109347, Jan. 2023, doi: 10.1016/j.patcog.2023.109347.

[15] A. Shabbir, M. Shabbir, A. R. Javed, M. Rizwan, C. Iwendi, and C. Chakraborty, "Exploratory data analysis, classification, comparative analysis, case severity detection, and internet of things in COVID-19 telemonitoring for smart hospitals," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 35, no. 4, pp. 507–534, Feb. 2022, doi: 10.1080/0952813x.2021.1960634.

[16] Applied Univariate, Bivariate, and Multivariate Statistics Using Python. 2021. doi: 10.1002/9781119578208.

[17] S. P. D'Almeida, S. Kamath, R. K, and M. KN, "Preprocessing Techniques for Rectal Cancer Diagnosis using MR Images," *ICSCA '24: Proceedings of the 2024 13th International Conference on Software and Computer Applications*, Feb. 2024, doi: 10.1145/3651781.3651809.

[18] G. Lu, X. Ge, T. Zhong, Q. Hu, and J. Geng, "Preprocessing enhanced image compression for machine vision," *IEEE Transactions on Circuits and Systems for Video Technology*, p. 1, Jan. 2024, doi: 10.1109/tcsvt.2024.3441049.

[19] S. Ghassel, A. Jabbarpour, J. Lang, E. Moulton, and R. Klein, "The effect of resizing on the natural appearance of scintigraphic images: an image similarity analysis," *Frontiers in Nuclear Medicine*, vol. 4, Feb. 2025, doi: 10.3389/fnume.2024.1505377.

[20] Riko Prananda Prayugo and Lailan Sofinah Harahap, "Median Filter Optimization and Sharpening Techniques to Improve Digital Image Quality", *Info Sains*, vol. 15, no. 01, pp. 117–127, Jul. 2025.

[21] A. Ponraj and A. Canessane, "Radial Basis Function Networks and Contrast-Limited Adaptive Histogram Equalization Filter Based Early-Stage Breast Cancer Detection Techniques," *Journal of Computer Science*, vol. 19, no. 6, pp. 760–774, Jun. 2023, doi: 10.3844/jcssp.2023.760.774.

[22] R.-C. Chen, C. Dewi, Y.-C. Zhuang, and J.-K. Chen, "Contrast limited adaptive histogram equalization for recognizing road marking at night based on Yolo models," *IEEE Access*, vol. 11, pp. 92926–92942, Jan. 2023, doi: 10.1109/access.2023.3309410.

[23] R. -C. Chen, C. Dewi, Y. -C. Zhuang and J. -K. Chen, "Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models," in *IEEE Access*, vol. 11, pp. 92926-92942, 2023, doi: 10.1109/ACCESS.2023.3309410.

[24] E. Goceri, "Medical image data augmentation: techniques, comparisons and interpretations," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12561–12605, Mar. 2023, doi: 10.1007/s10462-023-10453-z.

[25] M. Shantal, Z. Othman, and A. A. Bakar, "A novel approach for data feature weighting using correlation coefficients and Min–Max normalization," *Symmetry*, vol. 15, no. 12, p. 2185, Dec. 2023, doi: 10.3390/sym15122185.

[26] J.-H. Park, Y.-S. Kim, H. Seo, and Y.-J. Cho, "Analysis of Training Deep learning Models for PCB defect Detection," *Sensors*, vol. 23, no. 5, p. 2766, Mar. 2023, doi: 10.3390/s23052766.
[27] S. Krishnapriya and Y. Karuna, "Pre-trained deep learning models for brain MRI image classification," *Frontiers in Human Neuroscience*, vol. 17, Apr. 2023, doi: 10.3389/fnhum.2023.1150120.
[28] M. C et al., "A Robust DL Approach for Detection of Invasive Ductal Carcinoma in Whole Slide Images using DenseNet169," *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, Gautam Buddha Nagar, India, 2024, pp. 297-301, doi: 10.1109/IC3SE62002.2024.10593576.
[29] O. C. R. Rachmawati and Z. M. E. Darmawan, "The Comparison of Deep Learning Models for Indonesian Political Hoax News Detection," *COMMIT (Communication and Information Technology) Journal*, vol. 18, no. 2, pp. 123-135, Aug 2024, doi: 10.21512/commit.v18i2.10929
[30] Z. Đ. Vujovic, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, Jan. 2021, doi: 10.14569/ijacsa.2021.0120670.
[31] O. C. R. Rachmawati, A. R. Barakbah, and T. Karlita, "The Comparison of Activation Functions in Feature Extraction Layer using Sharpen Filter", *JAETS*, vol. 6, no. 2, pp. 1254–1267, Jun. 2025.
[32] A. Berger and S. Guda, "Threshold optimization for F measure of macro-averaged precision and recall," *Pattern Recognition*, vol. 102, p. 107250, Jun. 2020, doi: 10.1016/j.patcog.2020.107250.
[33] K. V. Andini and H. Akbar, "Komparasi Inception-v3 dan Inception-ResNet-v2 pada Citra Mammogram untuk Deteksi Dini Kanker Payudara," *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 14, no. 3, pp. 780–787, Jul. 2025, doi: 10.30591/smartcomp.v14i3.9283.
[34] I. Kurniawan, "Penerapan Metode YOLOv10 Untuk Deteksi Penyakit Pada Tanaman Kelapa Sawit Berdasarkan Citra Daun," *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 14, no. 3, pp. 768–779, Jul. 2025, doi: 10.30591/smartcomp.v14i3.8485.