ATCSJ
Applied Technology and Computing Science Journal
Open Access p-ISSN 2621-4458 | e-ISSN 2621-4474

Available online to journal2.unusa.ac.id
UNUSA
S4-Accredited – SK No.79/M/KPT/2023
Journal Page is available to https://journal2.unusa.ac.id/index.php/ATCSJ

Tuberculosis Detection based on Lung X-Ray Images Using Convolutional Neural Networks (CNN)

Rudi Kurniawan[1], Tessy Badriyah[2*], Iwan Syarif[3]

[1,2,3]*Politeknik Elektronika Negeri Surabaya, Indonesia*
Kampus PENS, Keputih, Sukolilo, Surabaya
[1]rudi.kurniawanai@gmail.com, [2*]tessy@pens.ac.id, iwanarif@pens.ac.id

Article history:

Received 10 September 2024
Revised 28 October 2024
Accepted 20 November 2024
Available online 10 December 2024

Keywords:

Tuberculosis, Convolutional Neural Networks, Chest X-ray, Early Stopping.

Abstract

Tuberculosis (TB) is an infectious disease caused by Mycobacterium tuberculosis, primarily affecting the lungs. Despite being preventable and curable, TB remains a significant global health issue, especially in developing countries. The success of TB treatment heavily depends on the accuracy of the diagnosis, which typically requires expertise from pulmonology or radiology specialists to interpret chest X-ray images. This study aims to design an assistive tool for TB detection that can automatically diagnose the disease using chest X-ray data. The study implemented a Convolutional Neural Network (CNN) architecture to analyze the X-ray images. Additionally, image preprocessing and early stopping methods were employed to enhance accuracy performance, optimize computation, and prevent overfitting. Experiment was conducting using 75% of the data as training data to generate the model and then applied to 25% of the data as testing data. This study comparing image sizes in RGB and grayscale modes. Experimental results show that the use of early stopping has a significant impact on training time, reducing training time substantially in almost all scenarios without drastically sacrificing accuracy. Without early stopping, accuracy does tend to be higher, as seen in grayscale color mode with an image size of 128x128, where the accuracy reaches 0.992, and in RGB mode with an image size of 64x64 which reaches 0.995. However, training time also increases significantly, for example for a 299x299 image with RGB mode, the training time reaches 927 seconds. Therefore, while RGB yields slightly higher accuracy, grayscale is recommended due to significantly faster training times. Additionally, the early stopping mechanism proves effective in reducing computational time, making the training process more efficient.

I. INTRODUCTION

Tuberculosis or commonly called TB, is still a serious global health problem, especially in developing countries. Tuberculosis is an infectious disease caused by Mycobacterium Tuberculosis which can attack the human body, generally the lungs [1]. In recent decades, the rate of spread of TB in the world has indeed

[2*]Corresponding Author

decreased, but along with the Covid-19 pandemic, TB cases have increased again. According to data from the WHO report, Indonesia itself is in second place with the largest number of TB sufferers in the world after India, followed by China in third place [2]. In 2020, as many as 56% of men over the age of 15, 32% of women, and 12% of children under the age of 15 had TB [3]. Not all TB sufferers have serious symptoms, around 90-95% of sufferers are not aware that they are infected with the disease [4]. In general, TB bacteria attack the lungs, but infection can affect other organs such as the brain, heart, liver and others [5]. There are several ways to detect TB, the first way is for patients to increase their awareness of TB-related symptoms, so that they can be detected early. The second way is screening, which is regular examination of patients who are at risk of developing TB symptoms. In the screening process, medical imaging plays a very important role in this case, chest X-ray images which are useful for diagnosis [6]. WHO recommends taking lung X-ray images as a screening and triage tool for tuberculosis [7]. In clinical practice, lung X-ray imaging is widely used to be examined by experienced doctors in the field to detect TB, but this is time consuming and requires a fairly long process [5]. Information technology is now undergoing a very significant transformation and innovation, especially in the field of image processing. One of the technologies used is Computer Vision and Machine Learning which has progressed and is used in various aspects, including in the health sector [4]. The use of Deep Learning has increased drastically over time, especially in the field of pattern recognition in fields such as speech and vision processing. Convolutional Neural Network (CNN) is one part of Deep Learning that has brought breakthroughs in the field of pattern recognition of images [8].

CNN shows high performance in classifying images naturally so that it can have a major impact on various aspects of life, especially in the health sector such as diagnosing diseases, examining and determining disease conditions, decision support tools for doctors, and can even assist in certain surgical activities [9]. TB diagnosis can be done in various ways, one of which is by using the results of lung X-ray images. However, the interpretation of these X-ray results requires special expertise from a lung specialist or radiologist [10].

Research [1] uses the Convolutional Neural Network (CNN) method to classify pulmonary tuberculosis. This study focuses on 2 classes, namely Tuberculosis and Normal with the dataset used from The Montgomery County (MC) CXR dataset with a total of 138 data in png format. In the preprocessing stage, data augmentation is used such as horizontal left and right, zoom, flip left and right resizing. In CNN, the structure uses an extraction stage consisting of convolution layers, batch normalization, relu activation function, dropout, and max pooling while the classification stage contains fully connected layers, flatten, density and softmax activation functions. With this modeling, an accuracy of 87.1% was obtained. Researchers [11] utilized chest X-ray image data from patients with tuberculosis and pneumonia. The CNN model is used to help diagnose these two diseases. The data used has been labeled as many as 4273 pneumonia images, 1989 normal images and 394 tuberculosis images. The data is divided into 80% training set data and 20% testing set data. The data set has gone through 3 stages of preprocessing, namely image resizing, changing RGB images to grayscale and Gaussian standardization of the image. On the train data, sampling techniques were carried out in the form of undersampling and oversampling data to balance the train data between classes. The results show that the best model is produced when trained using oversampled train data with an Area under Curve value for the tuberculosis class of 0.99 and an Area under Curve value for the pneumonia class of 0.98. Therefore, this best model is able to identify 86% of tuberculosis and 96% of pneumonia.

Chang et al. [12] used transfer learning technique on sputum culture images labeled with tuberculosis and obtained precision and sensitivity of 99% and 98%. However, classification of TB culture images requires specific samples from patients and is not more efficient when compared to classification using X-ray images. Generalized pre-trained CNN is used in a study by [13] for TB disease detection. This study aims to evaluate how CNN can be an alternative to Decision Tree-based systems for medical image classification. Researchers applied CNN to chest X-ray image data to identify if a patient has TB. In this study, the CNN model used is VGG16 in classifying X-ray images to identify patients with TB. Furthermore, data augmentation is also applied to see if it can improve better accuracy. Augmentation is done using TFLearn Data Augmentation available in TensorFlow with 16 random batch sizes from the dataset. The results indicate that the accuracy increases when VGG16 is applied to images that have undergone an augmentation process and achieves an accuracy of 81.25% and 80% with and without image augmentation, respectively. Accuracy in the classification process can be improved through different algorithms or modifying existing algorithms or by combining several algorithms into an ensemble model. In its implementation, increasing the accuracy in detecting tuberculosis based on chest X-ray images with an efficient method can make this Computer Aided Diagnosis device more reliable. Generally, X-ray images are used as a whole in detecting lung abnormalities using CNN, although diseases such as TB are only in certain areas. Therefore, the focus of this study is to apply the segmentation method to X-ray images in detecting TB based on transfer learning techniques with CNN. By focusing on points in certain areas of the X-ray image, it is expected to significantly improve the performance of TB detection.

The development of a TB detection system based on X-ray results using Convolutional Neural Network (CNN) is expected to help improve the speed and accuracy of TB diagnosis [14]. To overcome this problem, this study uses X-ray images of the lungs with the convolutional neural networks method that applies early stopping to make computation more efficient and avoid overfitting. Because often in the iteration process, the computation process to generate a model often passes through an epoch that must be set in advance, so this iteration must be passed until finished starting from epoch 1 to epoch n. While the best model results are seen from the comparison graph of the accuracy of the training data and testing data to see the best model that is not in an overfitting condition. For this reason, this study conducted training with a Deep Learning approach using the Convolutional Neural Networks (CNN) method which uses early stopping to make computation more efficient and avoid overfitting.

II. METHODS

Figure 1 System Design explains the stages carried out, namely data collection, data preprocessing, implementation of the convolutional neural networks method with early stopping and applying the modeling results to the test data.

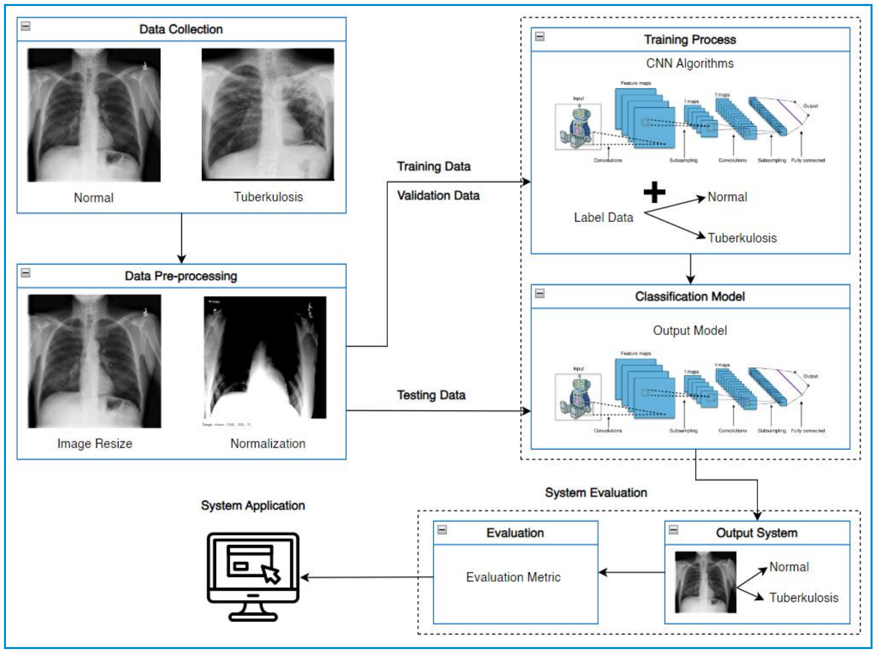

Figure 1. System Design

A. *Data Collection*

The data collection process in this study utilizes lung X-ray images categorized into two classes: Normal and Tuberculosis. The dataset, sourced from the study by Rahman et al. (5), comprises 3,500 normal lung X-ray images and 700 images with TB conditions, all in PNG format. These images were originally collected as part of a clinical study focused on reliable TB detection. Ethical considerations for this dataset include anonymizing patient data to protect confidentiality and ensuring that permissions were obtained in accordance with standard clinical data-sharing practices. This approach aligns with data protection regulations and ethical guidelines, promoting responsible use of patient images in AI research.

B. *Data Preprocessing*

In data processing, resizing techniques, data transformation into grayscale format and normalization are used. There are 5 (five) image sizes used: (64x64, 128x128, 224x224, 250x250, 299x299). Transforming image into grayscale is often a pragmatic step to simplify and speed up the training process without losing important information needed by the CNN. Normalization is a preprocessing step commonly used in machine learning to scale the features to a similar range.

The purpose of normalization is to scale the pixel values of the features to a range that is suitable for training machine learning models, typically between 0 and 1. The process involves dividing each value in the feature matrix X by 255, which is the maximum pixel value in typical image data. By dividing each pixel value by the maximum possible value (255 in this case), the pixel values are scaled to a range between 0 and 1. This normalization ensures that each feature contributes equally to the learning process and helps prevent certain features from dominating due to their larger scale.

C. *Performance Measurement*

In Figure 1, the Evaluation metric section uses Accuracy which is obtained from the confusion matrix in Table 1 below.

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN) \quad (5)$$

Table 1. Confusion Matrix

		Actual values	
		TRUE	FALSE
Prediction values	TRUE	**TP** (True Positive) Correct result	**FP** (False Positive) Unexpected result
	FALSE	**FN** (False Negative) Missing result	**TN** (True Negative) Correct absence of result

D. Convolutional Neural Networks Architecture

In this study, the convolutional neural networks method uses the following architecture:

1. Input Layer

input_shape: This model receives input with a shape defined by input_shape. This input shape is usually an image that is converted into a tensor shape, for example (height, width, number_of_channels).

2. First Convolution Layer

Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=input_shape); This layer has 32 filters/kernels measuring 3x3 that will perform convolution operations on the input image. Activation uses the ReLU (Rectified Linear Unit) function which helps in overcoming non-linearity problems. Kernel_initializer='he_uniform' is a way to initialize the weights in the model using a uniform distribution that is in accordance with the He method. padding='same' ensures that the output of the convolution operation is the same size as the input, because padding will add pixels to the edges of the image. input_shape is only needed in the first layer to determine the shape of the input.

3. First MaxPooling Layer

MaxPooling2D((2, 2)): This layer performs pooling operation with 2x2 filters, which means taking the maximum value from each 2x2 block, reducing the spatial dimension of the previous convolution output. This helps in reducing the feature size and computation required.

4. Second Convolution Layer

Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'):

This second convolution layer has 64 filters, which makes the model deeper and capable of capturing more complex features.

Followed by MaxPooling2D((2, 2)) to reduce the output dimension.

5. Third Convolution Layer

Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'):

This layer has 128 filters, which is more than the previous layer, to detect more detailed features.

Followed by MaxPooling2D((2, 2)) again to reduce the output dimension.

6. Flatten Layer

Flatten(): This layer flattens the output of the convolution and pooling layers into a one-dimensional vector, which can then be connected to the fully connected (Dense) layer.

7. Fully Connected Layer (Dense Layer)

Dense(128, activation='relu', kernel_initializer='he_uniform'):

This dense layer has 128 fully connected neuron units, using ReLU activation.

Dense(2, activation='softmax'): This last layer has 2 output units and uses the Softmax activation function to output the probability of each class (binary classification).

8. Compile the Model

opt = Adam(learning_rate=0.001): The optimizer used is Adam with a learning rate of 0.001, which is popular because it is adaptive and efficient in the training process. model.compile(optimizer=opt, loss='sparse_categorical_crossentropy', metrics=['accuracy']): The model is compiled using the sparse_categorical_crossentropy loss function, which is suitable for classification with labels that have not been converted to one-hot encoding. Accuracy is used as a metric to evaluate the performance of the model.

E. Early Stopping Implementation

The ModelCheckpoint and EarlyStopping methods will be implemented in this study. Both will be instantiated as objects and utilized as callback functions in fit_generator. ModelCheckpoint facilitates model preservation by monitoring a specific parameter. In this scenario, validation accuracy is observed using val_acc for ModelCheckpoint. The model will be saved only if the current epoch's validation accuracy surpasses that of the previous epoch. EarlyStopping, on the other hand, halts model training prematurely if there's no improvement in the monitored parameter. Here, validation accuracy is monitored with val_acc for EarlyStopping. A patience of 3 is set, meaning training will cease if there's no validation accuracy improvement over 3 epochs. In the experimental results of Figure 2, it shows that setting epoch = 15 will stop at epoch = 6, after 3x epochs the accuracy value is not improved. This method will be very efficient because the iteration will stop automatically without having to be run until the entire epoch has been determined.

```
Epoch 1/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 0s 700ms/step - accuracy: 0.8258 - loss: 3.6268
Epoch 1: val_accuracy improved from -inf to 0.95143, saving model to CNNModel.keras
99/99 ━━━━━━━━━━━━━━━━━━━━ 78s 775ms/step - accuracy: 0.8264 - loss: 3.6020 - val_accuracy: 0.9514 - val_loss: 0.1243
Epoch 2/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 0s 709ms/step - accuracy: 0.9741 - loss: 0.0764
Epoch 2: val_accuracy improved from 0.95143 to 0.96762, saving model to CNNModel.keras
99/99 ━━━━━━━━━━━━━━━━━━━━ 77s 778ms/step - accuracy: 0.9741 - loss: 0.0764 - val_accuracy: 0.9676 - val_loss: 0.0844
Epoch 3/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 0s 685ms/step - accuracy: 0.9776 - loss: 0.0607
Epoch 3: val_accuracy did not improve from 0.96762
99/99 ━━━━━━━━━━━━━━━━━━━━ 74s 746ms/step - accuracy: 0.9776 - loss: 0.0607 - val_accuracy: 0.9552 - val_loss: 0.1130
Epoch 4/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 0s 682ms/step - accuracy: 0.9785 - loss: 0.0615
Epoch 4: val_accuracy did not improve from 0.96762
99/99 ━━━━━━━━━━━━━━━━━━━━ 73s 742ms/step - accuracy: 0.9785 - loss: 0.0615 - val_accuracy: 0.9657 - val_loss: 0.0939
Epoch 5/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 0s 679ms/step - accuracy: 0.9888 - loss: 0.0293
Epoch 5: val_accuracy did not improve from 0.96762
Early stopping due to no improvement in validation accuracy.
99/99 ━━━━━━━━━━━━━━━━━━━━ 73s 739ms/step - accuracy: 0.9888 - loss: 0.0293 - val_accuracy: 0.9533 - val_loss: 0.1370
```

Figure 2. Early Stopping Mechanism

III. RESULTS AND DISCUSSION

In this study, experiments were conducted with various different image sizes, namely 64x64, 128x128, 224x224, 299x299, then display a comparison of model accuracy, training time and memory usage for each image size. For this purpose, it is necessary to create:

1. Create a function that can be repeated for each image size.
2. Measure training time.

3. Calculate memory usage.
4. Save the accuracy results of each experiment for comparison.

Experiments were also conducted with and without early stopping to see the difference in training time, whether the use of early stopping would be able to reduce training time significantly.

Table 2. Comparison of accuracy, training time and memory used **without early stopping** on several variations of Color Mode and Image Size

No	Color Mode	Image Size	Accuracy	Training time (s)	Memory used (MB)
1	Grayscale	64x64	0.98476190 47619047	38	-1120
2	Grayscale	128x128	0.99238095 23809524	164	408
3	Grayscale	224x224	0.98095238 09523809	493	-4791
4	Grayscale	250x250	0.97238095 23809523	615	-4501
5	Grayscale	299x299	0.98666666 66666667	898	-5971
6	RGB	64x64	0.99523809 52380953	45	287
7	RGB	128x128	0.98761904 76190476	183	-8068
8	RGB	224x224	0.98952380 95238095	560	2735
9	RGB	250x250	0.99619047 61904762	677	-500
10	RGB	299x299	0.98095238 09523809	927	-5494

The impact of early stopping on training time and accuracy was notable in these experiments, with clear reductions in training time across various image sizes and color modes. Early stopping consistently led to shorter training durations by halting the training process once no significant improvement in accuracy was observed. For example, for grayscale images at 64x64 resolution, early stopping reduced the training time from 38 seconds to 22 seconds, and for RGB images at 299x299 resolution, it decreased from 927 seconds to 802 seconds. These reductions in training time demonstrate that early stopping can be an efficient approach to optimizing resource use, especially for large image resolutions where computational load is higher. However, the effect of early stopping on model accuracy was nuanced. While in many cases early stopping did not substantially reduce accuracy, there were slight variations in performance across different image sizes and color modes. For instance, the accuracy for grayscale images at 224x224 resolution with early stopping was marginally lower than without early stopping (0.971 vs. 0.980). This suggests that early stopping may occasionally prevent the model from achieving its highest potential accuracy, especially on datasets or image sizes where feature complexity may require longer training times to achieve optimal learning.

Table 3. Comparison of accuracy, training time and memory used with early stopping on several variations of Color Mode and Image Size

No	Color Mode	Image Size	Accuracy	Training time (s)	Memory used (MB)
1	Grayscale	64x64	0.97238095 23809523	22	186
2	Grayscale	128x128	0.97333333 33333334	98	375
3	Grayscale	224x224	0.97142857 14285714	423	29
4	Grayscale	250x250	0.97523809 52380952	318	-5025
5	Grayscale	299x299	0.98190476 1904762	619	-5898
6	RGB	64x64	0.95904761 9047619	36	268
7	RGB	128x128	0.97523809 52380952	171	-5313
8	RGB	224x224	0.97333333 33333334	285	2250
9	RGB	250x250	0.99047619 04761905	403	60
10	RGB	299x299	0.97904761 9047619	802	-6090

IV. CONCLUSIONS AND RECOMMENDATIONS

This study uses the convolutional neural networks method to diagnose tuberculosis by analyzing several image sizes and in two color modes: RGB and grayscale. The early stopping mechanism is also implemented in CNN for computational time efficiency. And it is proven that by using the early stopping mechanism the time required for the training process becomes shorter and more efficient. The conclusions and recommendations simply state that the researcher thinks about each data presented relating back to the stated purpose in the introduction. It contains a summary of what is learned from the results obtained referenced to the introduction and conclusion, a reader should have a good idea of this research, even if only specific details.

Experiments conducted show that the model accuracy increases with increasing image size and RGB color mode compared to grayscale mode. On a 64x64 image with grayscale color mode, the accuracy reaches 0.972, but with the same size in RGB color mode, the accuracy decreases slightly to 0.959. An image size of 250x250 in RGB color mode produces the highest accuracy, which is 0.990. However, some memory usage data shows negative values, which may indicate errors in memory measurement or recording. Overall, image size and color mode have a significant effect on accuracy, with RGB tending to produce better results at larger image sizes.

Although RGB color mode tends to provide higher accuracy, grayscale color mode is more recommended due to the significant difference in training time. For example, for a 299x299 image, grayscale mode takes about 619 seconds to train, while RGB mode takes more than 800 seconds. Therefore, for applications that require time efficiency, grayscale mode is a more optimal choice without sacrificing too much accuracy.

V. ACKNOWLEDMENT

This study was funded under the Master's Thesis Scheme (Penelitian Tesis Magister) by the Directorate of Research and Community Service through the Center for Research and Community Service (P3M) of Politeknik Elektronika Negeri Surabaya, and the researchers are grateful for their support.

VI. REFERENCES

[1] M. Oloko-Oba and S. Viriri, Diagnosing tuberculosis using deep convolutional neural network, vol. 12119 LNCS. Springer International Publishing, 2020. doi: 10.1007/978-3-030-51935-3_16.
[2] Organização Mundial de Saúde, World health statistics 2022 (Monitoring health of the SDGs). 2022. [Online]. Available: http://apps.who.int/bookorders.
[3] J. Chakaya et al., “International Journal of Infectious Diseases Global Tuberculosis Report 2020 – Re fl ections on the Global TB burden , treatment and prevention efforts,” 2021, doi: 10.1016/j.ijid.2021.02.107.
[4] A. C. Sitepu, M. Sigiro, and J. Panjaitan, “Deteksi Penyakit Tuberkulosis Dengan Atensi Ganda Convolutional Neural Network Pada Citra X-Ray,” JUTISAL J. Tek. Inform. …, vol. 2, no. 1, 2022, [Online]. Available: https://universal.ac.id/jurnal/index.php/jutisal/article/view/11
[5] T. Rahman et al., “Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization,” IEEE Access, vol. 8, pp. 191586–191601, 2020, doi: 10.1109/ACCESS.2020.3031384.
[6] Q. H. Nguyen et al., “Deep Learning Models for Tuberculosis Detection from Chest X-ray Images,” 2019 26th Int. Conf. Telecommun., pp. 381–385, doi: 10.1109/ICT.2019.8798798.
[7] D. X. Andrew, D. X. X. Adam, and F. Auffermann, “Perceptual and Interpretive Error in Diagnostic Radiology — Causes and Potential Solutions,” Acad. Radiol., pp. 1–13, doi: 10.1016/j.acra.2018.11.006.
[8] L. N. Q. Vo et al., “Early evaluation of an ultra-portable x-ray system for tuberculosis active case finding,” Trop. Med. Infect. Dis., vol. 6, no. 3, 2021, doi: 10.3390/TROPICALMED6030163.
[9] Z. Z. Qin et al., “Tuberculosis detection from chest x-rays for triaging in a high tuberculosis-burden setting: an evaluation of five artificial intelligence algorithms,” Lancet Digit. Heal., vol. 3, no. 9, pp. e543–e554, 2021, doi: 10.1016/S2589-7500(21)00116-3.
[10] M. T. Khan, A. C. Kaushik, L. Ji, S. I. Malik, S. Ali, and D. Q. Wei, “Artificial neural networks for prediction of tuberculosis disease,” Front. Microbiol., vol. 10, no. MAR, pp. 1–9, 2019, doi: 10.3389/fmicb.2019.00395.
[11] L. A. Andika, H. Pratiwi, and S. Sulistijowati Handajani, “Convolutional neural network modeling for classification of pulmonary tuberculosis disease,” J. Phys. Conf. Ser., vol. 1490, no. 1, 2020, doi: 10.1088/1742-6596/1490/1/012020.
[12] P. Gidwani, U. Gori, A. Dedhia, and N. Shah, “Tuberculosis Detection Using Convolutional Neural Network,” SSRN Electron. J., 2021, doi: 10.2139/ssrn.3866534.
[13] M. Harahap, A. P. S. Pasaribu, D. R. Sinaga, R. Sipangkar, and S. Samuel, “Classification of Tuberculosis Based on Lung X-Ray Image With Data Science Approach Using Convolutional Neural Network,” Sinkron, vol. 7, no. 4, pp. 2193–2197, 2022, doi: 10.33395/sinkron.v7i4.11711.
[14] P. Senakama and J. Konvolusi, “Sistem diagnosa penyakit tbc berdasarkan gambar x- ray dengan dense convolutional network (densenet),” vol. 1, no. September, pp. 667–674, 2022.
[15] L. Delrue, R. Gosselin, and B. Ilsen, “Difficulties in the Interpretation of Chest Radiography,” 2011, doi: 10.1007/978-3-540-79942-9.