



Deep Learning Optimization of the EfficientNet Architecture for Classification of Tuberculosis Bacteria

Aeri Rachmad^{1*}, Husni¹, Juniar Hutagalung², Dian Hapsari³, Suci Hernawati⁴, Mohammad Syarief¹,
Eka Mala Sari Rochman¹, Yuli Panca Asmara⁵

¹ Department of Informatics, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan 69162, Indonesia

² Department of Information System, STMIK Triguna Dharma, Medan 20146, Indonesia

³ Department of Informatics Engineering, ITATS, Surabaya 60117, Indonesia

⁴ Head of the Batuputih Community Health Center, Sumenep 69453, Indonesia

⁵ Faculty of Engineering and Quantity Surveying, INTI International University, Negeri Sembilan 71800, Malaysia

Corresponding Author Email: aery_r@trunojoyo.ac.id

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.111008>

ABSTRACT

Received: 13 July 2024

Revised: 3 September 2024

Accepted: 14 September 2024

Available online: 31 October 2024

Keywords:

TB, CNN, EfficientNet, median filter, Adam

Tuberculosis (TB) remains a significant global issue, particularly in countries with low economic status and limited healthcare systems. One of the primary challenges is accurate early diagnosis, especially through microscopic examination of sputum samples. However, subjective interpretation and variations in microscopic image quality often hinder diagnostic accuracy. In recent years, the use of Convolutional Neural Networks (CNN) has increased to enhance TB diagnosis effectiveness. This study utilizes the EfficientNet architecture to understand the model's effectiveness in detecting TB in medical images. The dataset used consists of 1266 images, divided into training and testing data with a ratio of 70:30. Additionally, a median filter technique was applied for image preprocessing. Several optimization algorithms are used in this research, namely RMSprop, Stochastic Gradient Descent (SGD), Adam, and Stochastic Gradient Descent with Momentum (SGDM), to find the best scenario. The test results show that Adam optimization provides the best performance compared to the others. The results showed excellent performance, with a low loss rate (9.20%) and high accuracy (98.03%). The relatively fast model training time (122.81 seconds) also adds to the model's efficiency value. This confirms that EfficientNet B0 is an attractive choice for TB classification, with the hope that further development will improve accuracy and efficiency in diagnosing this disease.

1. INTRODUCTION

TB remains a highly significant global public health issue, particularly in countries with low economic levels and limited healthcare systems, where the prevalence of the disease is often high and its impact is widespread [1]. This disease is caused by the bacterium *Mycobacterium tuberculosis*, which spreads through the air and can infect anyone. If not identified and treated properly, it can lead to serious complications and even death [2]. The main challenge in controlling the spread of this disease is finding ways to identify and diagnose it early and accurately, even though preventive and control measures have been implemented [3].

One of the key techniques for achieving early detection of tuberculosis is by identifying TB bacteria through microscopic examination of sputum samples [2]. Although the microscopic identification of TB bacteria in sputum samples has advantages in terms of speed and high cost efficiency, the main challenges are subjective interpretation and variations in the quality of microscopic images, which often hinder the accuracy and consistency of the diagnostic results provided by

this method [3]. In recent years, along with rapid advancements in technology, especially in the development of artificial intelligence, specifically CNN, research has focused on overcoming these challenges and enhancing the effectiveness of the tuberculosis diagnosis process [4].

The feed-forward CNN is a model developed from the Multilayer Perceptron algorithm, an innovation that enables deep learning in image processing [5]. In this structure, each set of parameters to be adjusted in the convolutional layer, often referred to as convolutional filters, aims to extract deeper visual meaning from the original image input into the network [6]. This process aids in identifying meaningful features from the image, such as color patterns, textures, and shapes, which are then utilized for further analysis [7]. On the other hand, parameters placed in the fully connected layers aim to classify the extracted visual features into predefined target classes, such as distinguishing between TB and non-TB bacteria in the medical field. The convolutional layers in this architecture play a key role in forming a hierarchical abstraction of visual concepts from the initial images, with earlier layers focusing on low-level features like color and simple shapes, while

deeper layers within the network capture more complex visual concepts, such as identifying sub-parts of objects present in the image [8].

Through the proposed optimization methods, namely RMSprop, SGD, Adam, and SGDM, this research aims to provide deeper insights into the strengths and limitations of each model in the context of TB detection application in medical images. By applying CNN methods using the EfficientNet architecture, this step is expected to offer a more holistic understanding of the effectiveness and scalability of these models in addressing the challenges of TB detection in medical images.

2. MATERIALS AND METHODS

2.1 Methodology

The steps outlined in Figure 1 for identifying the bacteria causing TB involve analyzing microscopic images obtained by collecting several microscopic images related to TB disease.

These images then undergo a series of preprocessing steps that include various processes to ensure optimal image quality before being used in further analysis. An important stage in preprocessing is the use of a median filter, a highly effective

method in reducing noise and improving image details. The use of a median filter allows for filtering of microscopic images so that important information related to microbiological structures remains preserved while removing unwanted disturbances.

After completing the preprocessing stage, the next step is to train the model using previously separated data. This data is divided into two main parts: training data and testing data. In this testing, 70% of the total data is used for the model training process, while the remaining 30% is used to evaluate the performance of the trained model. This division aims to ensure that the model has the ability to effectively apply information from unseen data, thereby accurately identifying TB disease images.

The next step in this study involves the disease identification stage through the analysis of TB images. Here, the model that has undergone the training stage will be used to classify these images into two categories: TB bacteria or non-TB bacteria. The CNN architecture used in the research for classification is EfficientNet, which serves as a crucial foundation in implementing CNN. EfficientNet excels in recognizing and analyzing essential features in images, enabling the model to make accurate decisions based on the information available in these features. This ensures obtaining accurate final results.

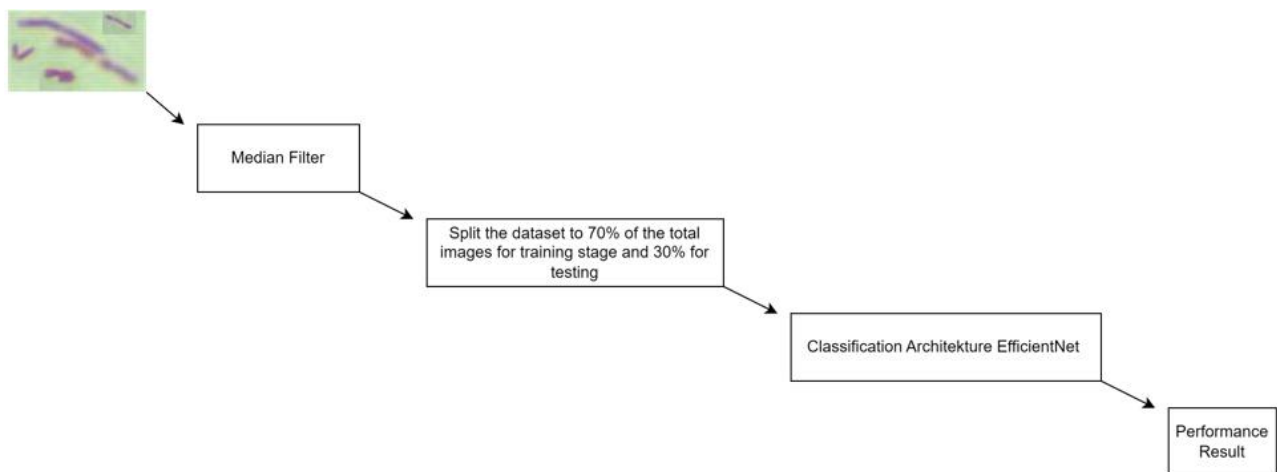


Figure 1. System diagram

2.2 Dataset

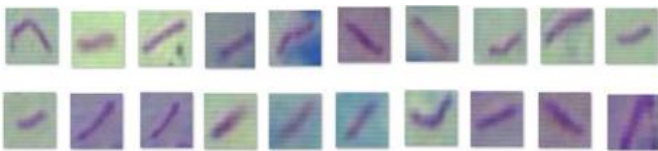


Figure 2. Dataset TB

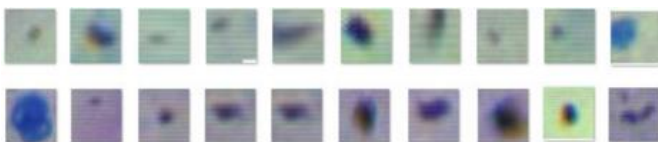


Figure 3. Dataset non-TB

The dataset used in this research consists of sputum images captured through microscopy, totaling 1266 images. This data is divided into two types: sputum images from TB patients,

comprising 633 data, and sputum images from non-TB patients, also comprising 633 data (refer to Figures 2 and 3). Images of sputum with dimensions of 800×600 pixels are included in the image dataset. The Ziehl-Neelsen (ZN) staining method was used to obtain these images. A Labomed Digi 3 digital microscope with an $L \times 400$ and an iVu 5100 digital camera with 5.0 MP were used for the process of taking pictures. At a $1000 \times$ magnification, the sputum images were captured with a resolution of 120 and a color depth of 24 bits.

2.3 Median filter

The median filter, first introduced by Tukey [9], plays an important role as an image processing technique that does not rely on linearity but instead uses a non-linear approach in processing image data [10]. The median filter method is designed with the primary goal of reducing the noise level present in an image, while simultaneously smoothing the distribution of pixel values [11]. The median filter process involves a series of structured steps, starting with sorting the

$$v_i = \beta_2 v_i + (1 - \beta_2) \left(\frac{\partial L}{\partial \theta_i} \right)^2 \quad (6)$$

Parameters (5) and (6) tend to experience bias as they approach the value of 1, especially if the initialization of time steps and decay rates is very small. To address this, bias correction and moment estimation are required by dividing parameters (5) and (6) by the difference between 1 and the decay factor. With this step, the bias that arises in the initial estimation can be corrected, resulting in more accurate parameters.

$$\hat{m}_i = \frac{m_i}{1 - \beta_1} \quad (7)$$

$$\hat{v}_i = \frac{v_i}{1 - \beta_2} \quad (8)$$

The developers of the Adam algorithm recommend using a beta-1 value of 0.9, a beta-2 value of 0.999, and an epsilon value of 10^{-8} . After obtaining the optimal values of parameters (5) and (6), the Adam formula can be computed as follows:

$$\theta_{t+1} = \theta_t - \frac{\hat{m}_i}{\sqrt{\hat{v}_i + \epsilon}} \quad (9)$$

Based on the formula of Adam, it is known that Adam uses the foundation of RMSProp but with gradient estimation through momentum methods. This combination aims to enhance training speed. With this method, Adam has succeeded in surpassing previous optimization algorithms both in training stages and in other experiments. However, Adam introduces new hyperparameters that can complicate hyperparameter tuning when facing increasingly complex problems.

SGDM is a variation of gradient descent optimization that consistently updates parameters for each training data. In the parameter update process, SGDM does not iterate, making it faster, especially for large datasets. The standard value for the learning rate in SGD is 0.01. The parameter update process in SGD can be described in Eq. (8) [21].

$$G_t = \nabla \theta J(\theta; x^{(i)}, y^{(i)}) \quad (10)$$

with θ as the updated parameter, η as the learning rate, and $x^{(i)}$ and $y^{(i)}$ as the training data. This ensures that each learning iteration updates parameters according to the processed data.

2.6 Confusion matrix

Confusion matrix is a table used to describe the performance of a classification method by comparing the model's predicted outcomes against the actual values of the observed objects [22]. By dividing the prediction outcomes into four categories: true positive, false positive, true negative, and false negative, the confusion matrix provides deep insights into how well a model can identify true and false objects. This aids in evaluating the strengths and weaknesses of the classification model, as well as enabling better optimization of the classification strategies employed [23]. Here is a table of the confusion matrix:

| | | | |
|----------|-------------------------------------|---|---|
| | Positive | Negative | |
| Positive | True Positive (TP) | False Negative (FN) | Sensitivity $\frac{TP}{(TP + FN)}$ |
| Negative | False Positive (FP) | True Negative (TN) | Specificity $\frac{TN}{(TN + FP)}$ |
| | Precision $\frac{TP}{(TP + FP)}$ | Negative Predictive Value $\frac{TN}{(TN + FN)}$ | Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$ |

Figure 6. Confusion matrix

The confusion matrix consists of four main components (refer to Figure 6):

True Positive (TP): This is the number of data points correctly predicted as positive by the model.

True Negative (TN): This is the number of data points correctly predicted as negative by the model.

False Positive (FP): This is the number of data points incorrectly predicted as positive by the model (negative).

False Negative (FN): This is the number of data points incorrectly predicted as negative by the model (positive).

3. RESULT

3.1 Test scenario

The next step is to train the EfficientNet architecture model using microscopic images of tuberculosis bacteria. Table 1 below summarizes a series of experiments aimed at gaining a deep understanding of the influence of optimizers, batch sizes, and learning rates on model performance. Consequently, test scenarios were conducted by varying combinations of the Adam, RMSProp, SGD, and SGDM optimizers while maintaining consistency with a batch size of 16, a learning rate of 0.0001, and 20 epochs. Table 1 presents below contains the results obtained from these experiments, providing a solid foundation for further analysis to determine the optimal strategy for the appropriate learning configuration.

Table 1. Test scenario

| | Optimizer | Batch | Learning Rate | Epoch |
|---------------|-----------|-------|---------------|-------|
| Test Scenario | Adam | 16 | 0.0001 | 20 |
| | RmsProp | 16 | 0.0001 | 20 |
| | SGD | 16 | 0.0001 | 20 |
| | SGDM | 16 | 0.0001 | 20 |

3.2 Test result

The trial results on the images were conducted using the EfficientNet model architecture, which has been proven to be efficient in resource utilization. The trial results indicate that the EfficientNet model with the Adam optimizer, batch size of 16, learning rate of 0.0001, and 20 epochs can be seen in Figures 7-10.

Test results using the RMSprop optimizer in Figure 7 show that the model has quite good performance on training data with an accuracy of 90.48% and a loss of 23.51%. However, the model performance on validation data experienced a significant decrease, with an accuracy of 51.23% and a loss of 68.49%. This indicates the possibility of overfitting, where the model fits the training data too well but is less able to generalize to new data.

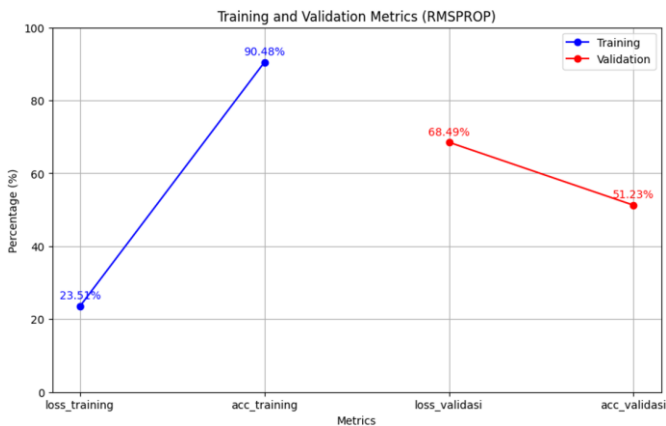


Figure 7. RMSProp optimizer test results

Test results using the SGD optimizer in Figure 8 show that the model has unsatisfactory performance on training data with an accuracy of 58.47% and a loss of 67.77%. Apart from that, the model performance on validation data also did not show significant improvement, with an accuracy of 37.93% and a loss of 69.20%. This indicates that the model has not been able to learn and generalize well, both on training data and validation data.

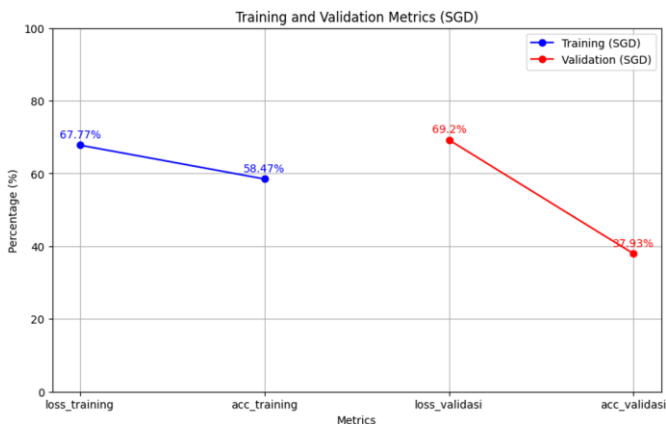


Figure 8. SGD optimizer trial results

Test results using the SGDM optimizer in Figure 9 show that the model has quite good performance on training data with an accuracy of 60.57% and a loss of 66.95%. Apart from that, the model performance on validation data also shows adequate results with an accuracy of 66.50% and a loss of 68.78%. This indicates that the model is able to learn and generalize quite well, although there is still room for further improvement in reducing loss values and increasing accuracy on training and validation data.

The test results using the Adam optimizer in Figure 10 achieved a loss value of 9.20%, which shows a relatively low level of information loss during the training process.

Additionally, the model accuracy reached 98.03%, demonstrating the model's ability to classify data with a high level of correctness. Furthermore, the precision metric of 98.04% indicates the proportion of true positive results, while the recall metric of 98.03% indicates the proportion of true positive data correctly identified from all actually positive data. The F1-score of 98.03% combines both metrics to provide an overall assessment of the model's quality in predicting data classes. The time required to train the EfficientNet model was 122.81 seconds, which is relatively fast considering the model's good performance in producing accurate results. Overall, the trial results demonstrate that the EfficientNet architecture performs very well in data classification, with high accuracy and low loss rates, as well as the ability to maintain a balance between precision and recall. The training time for the model is also quite efficient, making EfficientNet an attractive choice for various classification tasks. Table 2 and Figure 11 present a comparison of metrics between the testing and validation stages, illustrating the consistency and performance of the model in both stages.

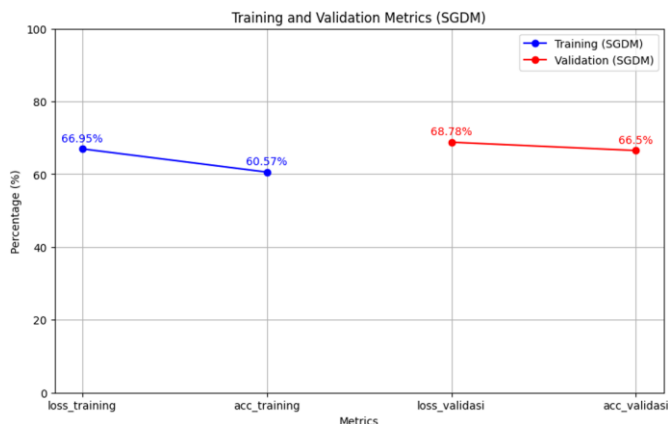


Figure 9. SGDM optimizer trial results



Figure 10. Adam optimizer test results

Table 2. Comparison of metrics between training and validation stages using Adam optimization

| | Loss | Acc | Precision | Recall | F1-Score | Time (s) |
|------------|-------|-------|-----------|--------|----------|----------|
| Training | 16.61 | 97.24 | 97.25 | 97.24 | 97.24 | 122.81 |
| Validation | 9.20 | 98.03 | 98.04 | 98.03 | 98.03 | 122.81 |

Figure 12 depicts the loss performance of the EfficientNet B0 model in TB classification. There is significant variation in loss between training and validation data during model

iterations. Initially, there is substantial variation in loss for both types of data. However, as the iterations progress, the loss on the validation data starts to stabilize and decrease,

indicating that the model is improving its performance and becoming better at generalizing on new data. Nevertheless, some points still show significant increases in loss.

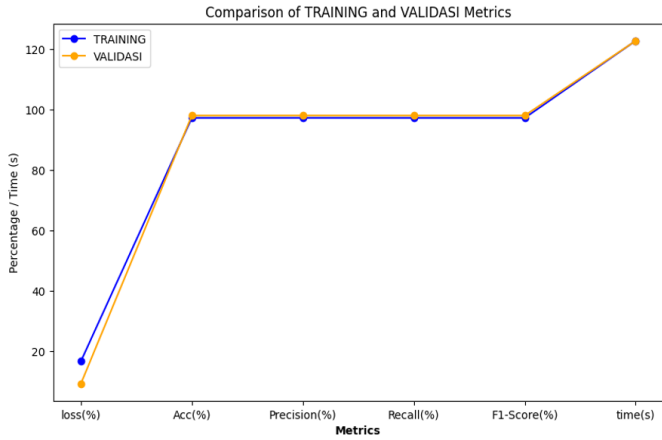


Figure 11. The EfficientNet test results using Adam optimization

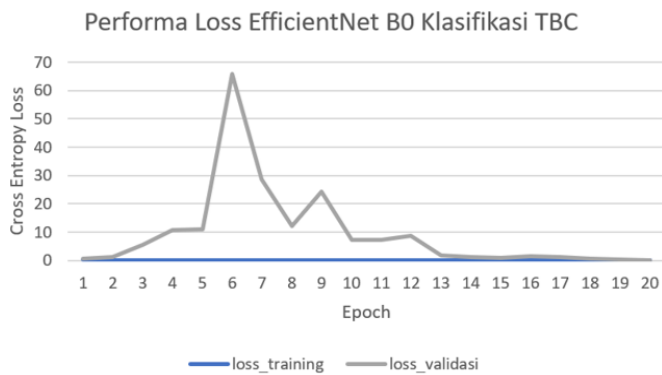


Figure 12. EfficientNet loss performance graph

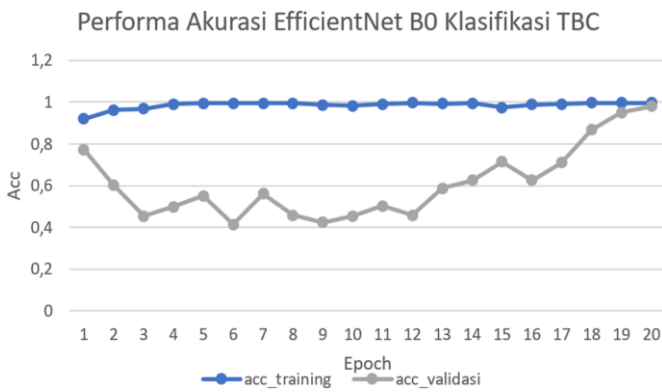


Figure 13. EfficientNet accuracy performance graph

Figure 13 shows the accuracy performance of the EfficientNet B0 model in TB classification. There are fluctuations in accuracy on both training and validation data during model iterations. Initially, there is significant variation in accuracy for both types of data. However, as the iterations progress, an overall improvement in accuracy on both data sets is observed, indicating the model's enhanced ability to classify data more accurately. Some points show significant increases in accuracy, indicating an overall improvement in model performance.

4. CONCLUSIONS

The conclusion from the experiments with EfficientNet using the Adam optimizer with a batch size of 16, learning rate of 0.0001, and 20 epochs shows excellent performance in classifying TB data, with a low loss rate (9.20%) and high accuracy (98.03%). Precision, recall, and F1-score metrics also indicate the model's ability to predict data classes effectively. The relatively fast model training time (122.81 seconds) adds value to the model's efficiency. Performance graphs demonstrate a consistent decrease in loss on validation data and an overall increase in accuracy. This confirms that EfficientNet B0 is an attractive choice for TB classification, with the hope that further development will improve accuracy and efficiency, contributing positively to the medical field, particularly in diagnosing diseases like tuberculosis.

REFERENCES

- [1] Rochman, E.M.S., Suprajitno, H., Rachmad, A., Santosa, I. (2023). Utilizing LSTM and K-NN for anatomical localization of tuberculosis: A solution for incomplete data. *Mathematical Modelling of Engineering Problems*, 10(4): 1114-1124. <https://doi.org/10.18280/mmep.100403>
- [2] World Health Organization. (2019). *Global tuberculosis report 2019*. Geneva, Switzerland: World Health Organization, pp. 1-297.
- [3] Sugirtha, G.E., Murugesan, G. (2017). Detection of tuberculosis bacilli from microscopic sputum smear images. In *2017 Third International Conference on Biosignals, Images and Instrumentation (ICBSII)*, Chennai, India, pp. 1-6. <https://doi.org/10.1109/ICBSII.2017.8082271>
- [4] Kumar, S., Arif, T., Alotaibi, A.S., Malik, M.B., Manhas, J. (2023). Advances towards automatic detection and classification of parasites microscopic images using deep convolutional neural network: Methods, models and research directions. *Archives of Computational Methods in Engineering*, 30(3): 2013-2039. <https://doi.org/10.1007/s11831-022-09858-w>
- [5] Rachmad, A., Syarief, M., Hutagalung, J., Hernawati, S., Rochman, E.M.S., Asmara, Y.P. (2024). Comparison of CNN architectures for mycobacterium tuberculosis classification in sputum images. *Ingénierie des Systèmes d'Information*, 29(1): 49-56. <https://doi.org/10.18280/isi.290106>
- [6] Rachmad, A., Sonata, F., Hutagalung, J., Hapsari, D., Fuad, M., Rochman, E.M.S. (2023). An automated system for osteoarthritis severity scoring using residual neural networks. *Mathematical Modelling of Engineering Problems*, 10(5): 1849-1856. <https://doi.org/10.18280/mmep.100538>
- [7] Hassanpour, M., Malek, H. (2020). Learning document image features with SqueezeNet convolutional neural network. *International Journal of Engineering*, 33(7): 1201-1207. <https://doi.org/10.5829/ije.2020.33.07a.05>
- [8] Leadholm, N., Stringer, S. (2022). Hierarchical binding in convolutional neural networks: Making adversarial attacks geometrically challenging. *Neural Networks*, 155: 258-286. <https://doi.org/10.1016/j.neunet.2022.07.003>
- [9] He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J. (2016). Deep

- residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vegas, NV, USA, pp.770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [10] Zhu, Y., Huang, C. (2012). An improved median filtering algorithm for image noise reduction. *Physics Procedia*, 25: 609-616. <https://doi.org/10.1016/j.phpro.2012.03.133>
- [11] Khan, S., Lee, D.H. (2017). An adaptive dynamically weighted median filter for impulse noise removal. *EURASIP Journal on Advances in Signal Processing*, 2017: 1-14. <https://doi.org/10.1186/s13634-017-0502-z>
- [12] Murinto, B.M. (2012). Analisis perbandingan metode 2D median filter dan multi level median filter pada proses perbaikan citra digital. *Jurnal Informatika*, 6(2): 654-662.
- [13] Maulana, I., Andono, P.N. (2016). Analisa perbandingan adaptif median filter dan median filter dalam reduksi noise salt & pepper. *Cogito Smart Journal*, 2(2): 157-166. <https://doi.org/10.31154/cogito.v2i2.26.157-166>
- [14] Boateng, K.O., Asubam, B.W., Laar, D.S. (2012). Improving the effectiveness of the median filter. *International Journal of Electronics and Communication Engineering*, 5(1): 85-97.
- [15] Atila, Ü., Uçar, M., Akyol, K., Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61: 101182. <https://doi.org/10.1016/j.ecoinf.2020.101182>
- [16] Tan, M., Le, Q.V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, pp. 10691-10700.
- [17] Liu, J., Huang, Y. (2020). Comparison of different CNN models in tuberculosis detecting. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(8): 3519-3533. <https://doi.org/10.3837/tiis.2020.08.021>
- [18] Kusumah, H., Zahran, M., Rifqi, K., Putri, D., Waktu Hapsari, E. (2023). Deep learning pada detektor jerawat: Model YOLOv5. *Journal Sensi: Strategic of Education in Information System*, 9(1): 24-35. <https://doi.org/10.33050/sensi.v9i1.2620>
- [19] Arouri, Y., Sayyafzadeh, M. (2022). An adaptive moment estimation framework for well placement optimization. *Computational Geosciences*, 26(4): 957-973. <https://doi.org/10.1007/s10596-022-10135-9>
- [20] Kingma, D.P., Ba, J.L. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980v4*.
- [21] Chen, R.C., Dewi, C., Huang, S.W., Caraka, R.E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1): 52. <https://doi.org/10.1186/s40537-020-00327-4>
- [22] Rochman, E.M.S., Setiawan, W., Hardi, S., Permana, K.E., Husni, Asmara, Y.P., Rachmad, A. (2024). Classification of salt quality based on the content of several elements in the salt using machine learning. *Mathematical Modelling of Engineering Problems*. 11(4): 1005-1012. <https://doi.org/10.18280/mmep.110417>
- [23] Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, 17(1): 168-192. <https://doi.org/10.1016/j.aci.2018.08.003>