



Comparison of CNN Architectures for Mycobacterium Tuberculosis Classification in Sputum Images

Aeri Rachmad^{1*}, Mohammad Syarief¹, Juniar Hutagalung², Suci Hernawati³, Eka Mala Sari Rochman¹,
Yuli Panca Asmara⁴

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

⁴ 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/isi.290106>

ABSTRACT

Received: 27 August 2023

Revised: 24 November 2023

Accepted: 8 December 2023

Available online: 27 February 2024

Keywords:

tuberculosis, sputum images, convolutional neural network (CNN), classification, AlexNet

Tuberculosis (TB) is a preventable and treatable infectious disease, but remains a serious problem in high-risk countries. Accurate early detection remains a challenge despite prevention efforts. The primary method of detecting tuberculosis is identifying bacteria in sputum samples using a microscope. This research focuses on the use of Convolutional Neural Network (CNN) with the AlexNet, ResNet-18, ResNet-50, and VGG-16 architectures in the early detection and classification of Tuberculosis (TB) through processing images of TB patients' sputum. A dataset of sputum images was collected and processed to ensure quality and adequate representation. Each CNN model was trained using deep learning techniques on the prepared dataset. The aim of this research is to compare the performance of each model in recognizing and classifying sputum images containing Mycobacterium tuberculosis bacteria and those without TB bacteria. The research results show that AlexNet architecture outperforms ResNet-18, ResNet-50 and VGG-16 in classification accuracy of Mycobacterium tuberculosis. The best validation accuracy achieved was 93.42% with the fastest time of 5 minutes and 52 seconds using AlexNet architecture. Identifying the most appropriate AlexNet architectural model could unlock the potential for developing automated systems that efficiently identify TB, thereby enabling faster and more timely medical intervention.

1. INTRODUCTION

Tuberculosis (TB) is one of the infectious diseases that remains a serious global health challenge. The World Health Organization (WHO) reports that millions of people are infected with tuberculosis every year and thousands of deaths are caused by this disease [1]. Prevention, early detection and proper treatment are key to combating the spread of tuberculosis [2]. To address the challenges faced in early detection and diagnosis of TB, artificial intelligence technology, particularly Convolutional Neural Networks (CNN), has become a promising focus of research in the medical field [3].

Tuberculosis bacteria have small variations in size, with a length of between 1 and 4 μm . while the thickness measures between 0.3 and 0.6 μm [4, 5]. A distinctive feature of tuberculosis bacteria found in sputum is their reddish coloration, referred to as acid-resistant bacteria (BTA). This coloration results from the application of Ziehl-Neelsen (ZN) staining methods to the staining fluid [6].

Numerous studies have focused on identifying and categorizing sputum images containing TB bacteria. For example, in 2012, Purwanti and Widiyanti achieved 91.33%

accuracy using the Learning Vector Quantization (LVQ) method [7]. Other researchers achieved 77.5% accuracy by using ANN [8]. Mithra and Emmanuel's research in 2018, achieved 91.38% precision using the (GFNN) Gaussian Fuzzy Neural Network technique. Furthermore, by using deep neural networks, Dinesh achieved 95.05% accuracy [9].

In the era of information technology and artificial intelligence, image processing methods and computation-based approaches are increasingly being applied in the medical field, including disease detection and diagnosis [4]. Convolutional Neural Networks (CNN) have proven to be highly effective in object recognition in images and have shown impressive results in various applications, including the recognition of cancer and other diseases [5].

The Convolutional Neural Network (CNN) method has found wide application among researchers due to its reputation as a leading algorithm for object detection [10]. Previous research has proposed AlexNet, LeNet, ZFNet, VGGNet, ResNet and GoogleNet as CNN architectural models [11]. The CNN design with ResNet has shown the capacity to group pictures inside the ImageNet dataset with an exactness of 80.62% for the best accuracy. In 2015, it outperformed competing architectures such as GoogleNet, AlexNet and

VGGNet in several competitions [12].

The research involves collecting a dataset of sputum images from TB patients and adopting a CNN-based approach to detect the Mycobacterium tuberculosis bacteria in the images. Data processing is carried out to ensure the quality and adequate representation of the training data.

Each CNN model is then trained on the prepared dataset using deep learning techniques. The aim is to compare the performance of each model in detecting and classifying sputum images with and without TB bacteria. The results of this research will help understand the strengths and weaknesses of each model in the context of tuberculosis detection applications.

Through the use of artificial intelligence technologies, by comparing several CNN architectures such as AlexNet, ResNet-18, ResNet-50, and VGG-16, this research can make a significant contribution to early detection and automatic accurate diagnosis of TB tuberculosis.

2. MATERIALS AND METHOD

In this study, the image dataset comprises sputum images with dimensions of 800×600 pixels. These images were obtained through the Ziehl-Neelsen (ZN) staining method. The image capture process utilized a Labomed Digi 3 digital microscope equipped with an Lx400 and an iVu 5100 digital camera module with a 5.0-megapixel capacity. The captured sputum images have a resolution of 120, a color depth of 24 bits, and were acquired at a magnification level of 1000x. These specific details provide a comprehensive description of the image data used in the study, offering insights into its spatial characteristics and the imaging equipment employed for data acquisition.

In the process of identifying TB bacteria, there are several stages that need to be carried out, as seen in Figure 1. First, the preprocessing stage involves using a median filter. The median filter is used to improve the image quality. Its purpose is to remove noise from the image so that the TB bacteria image becomes clearer and black spots can be eliminated. In the next stage, all detected TB or non-TB bacteria will be grouped into two classes, and then the image size will be resized to 50x50 pixels.

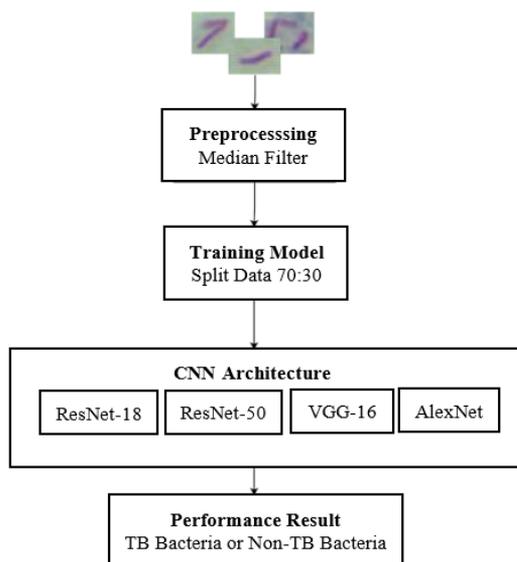


Figure 1. System diagram

The third stage is the model training, where the experimental data is divided into training data and testing data. The training data consists of 70% of the total data, while the testing data consists of 30% of the total data. After the model training is completed, the final stage is the classification stage. This stage identifies the disease from the TB images and produces two classes: TB bacteria or non-TB bacteria, using a Convolutional Neural Network (CNN) with several architecture experiments including ResNet-18, ResNet-50, AlexNet, and VGG-16.

2.1 Median filter

The Median Filter is an image processing method that is a non-linear filter developed by Tukey [13]. The purpose of this method is to reduce noise and smoothen the image [14]. In this method, the pixel values in a group of pixels with an odd number are sorted, and the median value of that group is calculated. Subsequently, this median value is used to replace the pixel value at the center of the filter window or region [15]. The median filter method is highly effective in removing noise from images and is recognized as one of the best filtering methods [16]. Below is an example of applying the median filter method with a 3×3 matrix containing main pixels, as seen in Figure 2 [13].

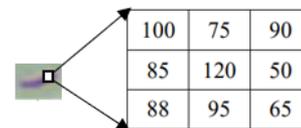


Figure 2. 3×3 pixel area

2.2 Convolutional neural network

CNN (Convolutional Neural Network) is a type of neural network developed and used for image recognition [17]. It is extremely efficient in image classification, image retrieval, object detection, and more. CNN automatically extracts features from images related to a specific domain [18]. In Figure 3 [19], it can be observed that the process of inputting data into the Convolutional Neural Network (CNN) starts with passing through the Convolutional Layer. In this layer, the input data undergoes a convolution operation to detect various features in the image.

After passing through the convolutional layer, the data goes through the Rectified Linear Unit (ReLU) Layer, which aims to recognize non-linear patterns in the data and activate neurons by transforming negative values into zeros, thus enhancing efficiency and data representation.

Next, the data undergoes the Pooling Layer, where larger data areas are reduced to smaller areas by taking the maximum or average values from the data. This helps in reducing data dimensions and extracting the most important features.

After the convolution, ReLU activation, and pooling processes are completed, the data is directed to the Fully Connected Layer. In this layer, each neuron is connected to every neuron in the previous layer. This allows the network to learn more complex relationships in the data.

Finally, the results from the Fully Connected Layer are fed into the Softmax Layer, where probability values are computed for each class. Softmax converts numerical values into probabilities, resulting in a probability distribution for each class.

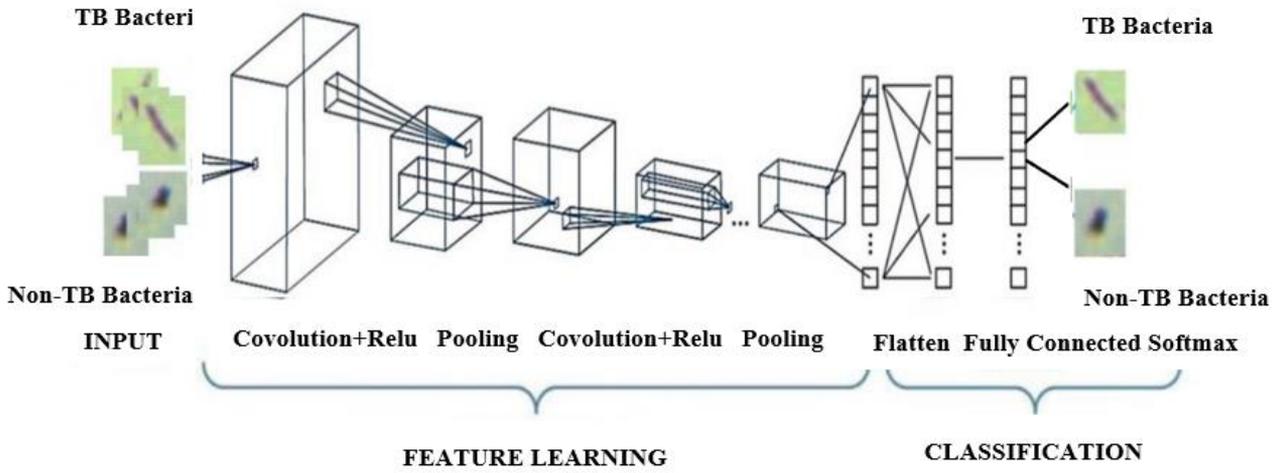


Figure 3. CNN architecture

As a result, the input data are placed in the class with the greatest likelihood, meaning that the class with the highest probability is the most likely class that corresponds to the input data.

2.2.1 ResNet architecture

The Deep Residual Network, also known as Residual Network (ResNet), was designed by Kaiming He. Based on the research titled "Deep Residual learning for image recognition" in 2016, ResNet emerged as the champion in the ILSVRC 2015 competition, outperforming architectures like VGG, GoogleNet, and others. The main concept behind developing the ResNet model was to address the issue of vanishing gradient descent, which often occurs in deep convolutional neural networks (CNN) [12].

ResNet (Deep Residual Network) tackles this problem by implementing skip connections or shortcuts. These skip connections allow direct flow of information from one layer to the input of a deeper layer, enabling ResNet to have more layers without facing performance degradation [18]. Figure 4 illustrates how skip connections are applied in ResNet, especially in the second and third layers that contain ReLU functions and batch normalization between these architectures are present [12]. By implementing these skip connections, ResNet has achieved better performance in handling deep learning problems with numerous *layers*.

The ResNet architecture encompasses various configurations distinguished by the number of layers. Specifically, these configurations are denoted as ResNet-18,

ResNet-34, ResNet-50, ResNet-101, and ResNet-152. Each of these variants represents a different depth in terms of the number of layers, providing flexibility for researchers and practitioners to choose a model that aligns with the specific requirements of their tasks or computational resources [19].

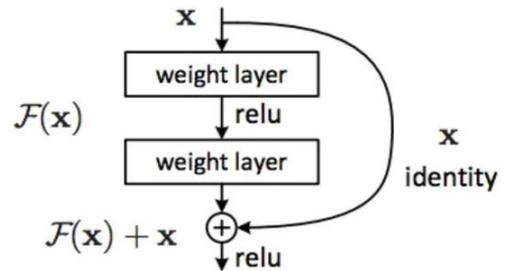


Figure 4. Deep Residual Network

2.2.2 AlexNet architecture

AlexNet is one of the early examples of applying deep learning to large-scale image classification, developed by Alex Krizhevsky and his colleagues [20]. Figure 5 [21] shows that AlexNet consists of 8 layers, with 5 convolutional layers in the first layer and 3 fully connected layers in the other layers. The first (1) and second (2) convolutional layers are followed by normalization and pooling layers in sequence, while the last convolutional layer is followed by one pooling layer [22].

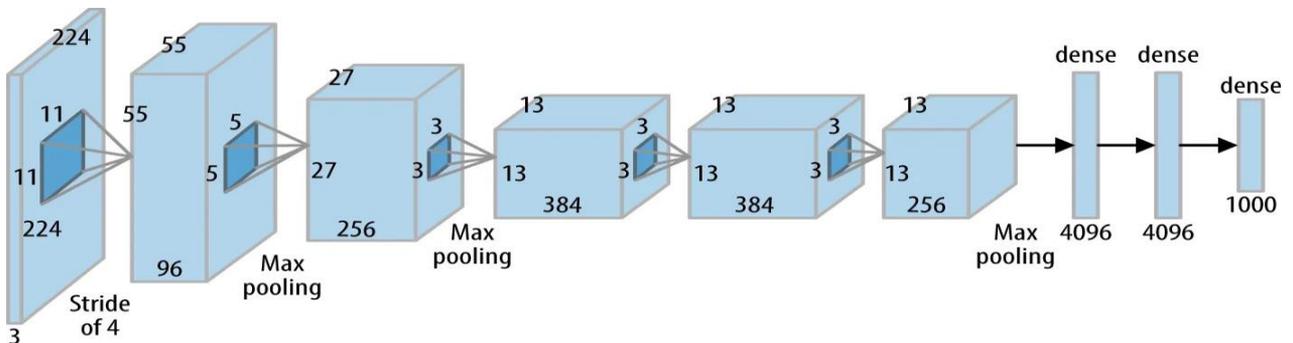


Figure 5. AlexNet architecture

2.2.3 VGG-16 architecture

The VGG model, designed by Karen Simonyan and Andrew Zisserman, is a Convolutional Neural Network (CNN) renowned for its impressive performance in classification and recognition tasks [23]. This deep neural network operates on input images with dimensions of 224×224×3 (RGB format). The image undergoes processing through convolutional and max-pooling layers, employing a 3×3 kernel size and a stride of 1 [11]. The max-pooling layer utilizes a 2×2 window size with a stride of 2. In its fully connected layers, the VGG model comprises 4096 channels, culminating in 6 channels in the last layer. The final layer incorporates the softmax activation function, which is commonly used in multi-class classification

problems. For clarity, the VGG-16 architecture is visually represented in Figure 6 and Figure 7 [23]. These architectural details provide a comprehensive understanding of the VGG model's configuration and the sequential flow of operations through its layers.



Figure 6. Layer architecture of VGG-16

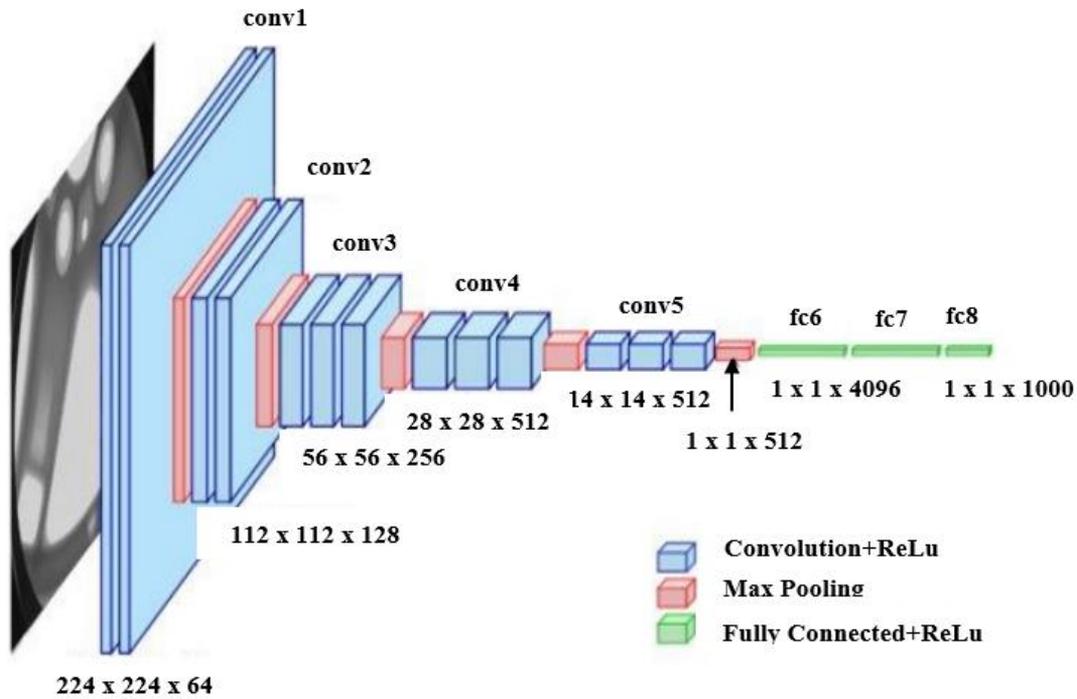


Figure 7. Architecture of VGG-16

2.3 Confution matrix

The calculation of accuracy in machine learning is commonly known as the Confusion Matrix. The Confusion Matrix provides detailed information about the comparison of the classification results performed by the system or model with the actual classification results from the data [24, 25]. At this stage, there are several common and frequently used performance matrices, namely accuracy, precision, and recall. The equations to find the values of accuracy, precision, and recall are as follows [26, 27]:

(a) Accuracy

Accuracy measures how accurate the model is in overall classification. It is the ratio of the number of correct predictions (True Positive and True Negative) to the total number of data examples.

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / \text{Total Data}$$

(b) Precision

Precision measures how many of the examples classified as positive are truly positive. It helps identify the rate of false positive errors.

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive})$$

(c) Recall

Recall measures how many of the overall positive examples

are correctly identified by the model. It helps identify the rate of false negative errors.

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative})$$

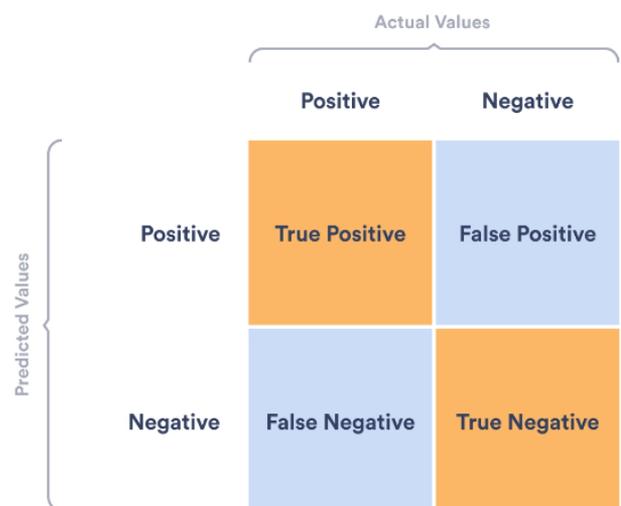


Figure 8. Confution matrix

3. RESULT

The research was conducted using a 64-bit Windows 10 laptop with specific hardware specifications, including an Intel Core i5 processor running at 1.6-1.8GHz and 8GB of RAM. The software employed for the study was MATLAB R2019a, a platform known for its capabilities in numerical computing, including tasks such as image processing and machine learning.

These hardware and software details have implications for the study's outcomes. The laptop's processing power and memory capacity may influence the reported processing times, especially considering the complexity of the neural network architectures and the size of the datasets used. Replicating or extending the research should consider the compatibility of available resources.

Furthermore, researchers should be aware that variations in hardware or software versions could result in slightly different results when replicating experiments. Considering the constraints of the laptop's specifications, it may be worthwhile to optimize code and models to ensure efficient processing within these limitations. Overall, understanding the impact of the chosen hardware and software environment is crucial for

interpreting and applying the research findings.

This stage is a trial stage where the data is divided into two classes. The first class is TB Bacteria and the second class is Non-TB Bacteria. This test divides the data into 70% training data and 30% testing data. 866 data were used for training and 380 data were used for testing. The testing phase evaluates several architectures available in CNN, including ResNet-18, AlexNet, ResNet-50 and VGG-16. Table 1 shows the scenarios that will be conducted in this research.

In the test results, ResNet-18 demonstrated an accuracy of 86.32% with a testing time of 11 minutes and 23 seconds. Figure 9 shows that ResNet-18 is a promising architecture for image classification with a relatively good accuracy rate and efficient computational time.

Table 1. Trial scenario

Scenario	Architecture	Learning Rate	Bath Size	Epoch
1	ResNet 18	0.0001	4	1-5
2	AlexNet	0.0001	4	1-5
3	ResNet 50	0.0001	4	1-5
4	VGG 16	0.0001	4	1-5

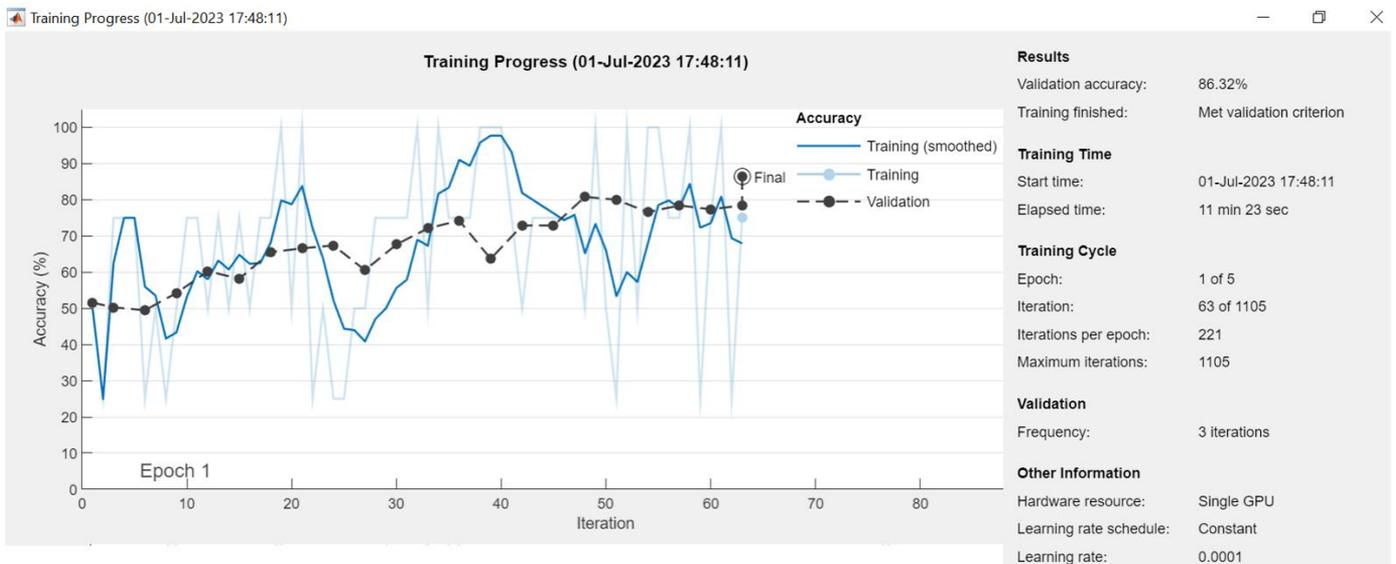


Figure 9. Graph of ResNet-18 scenario test results

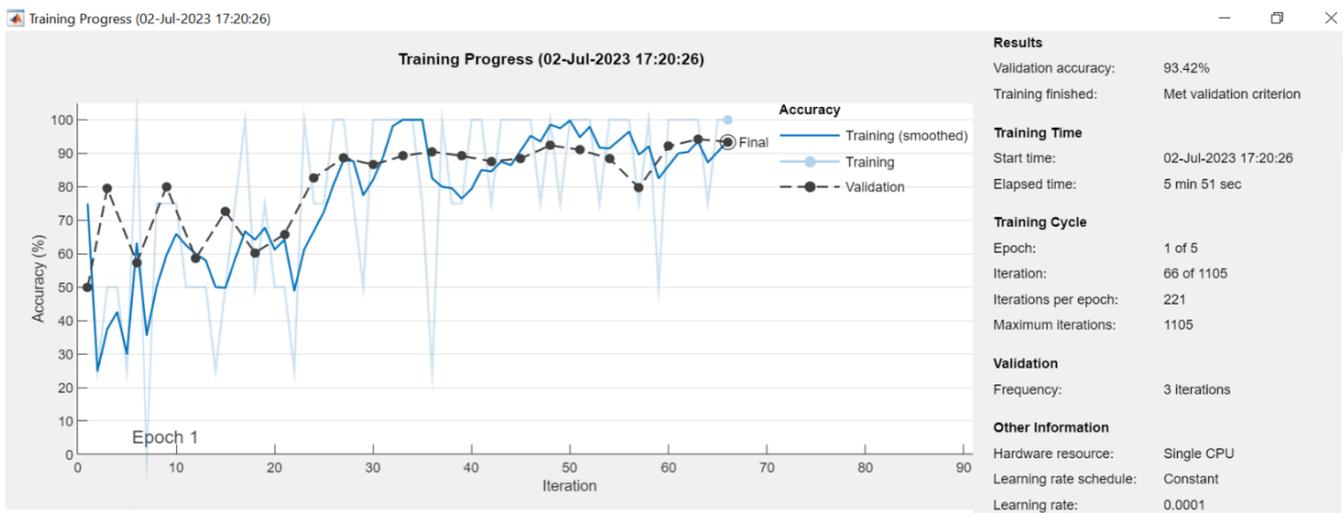


Figure 10. Graph of AlexNet scenario test results

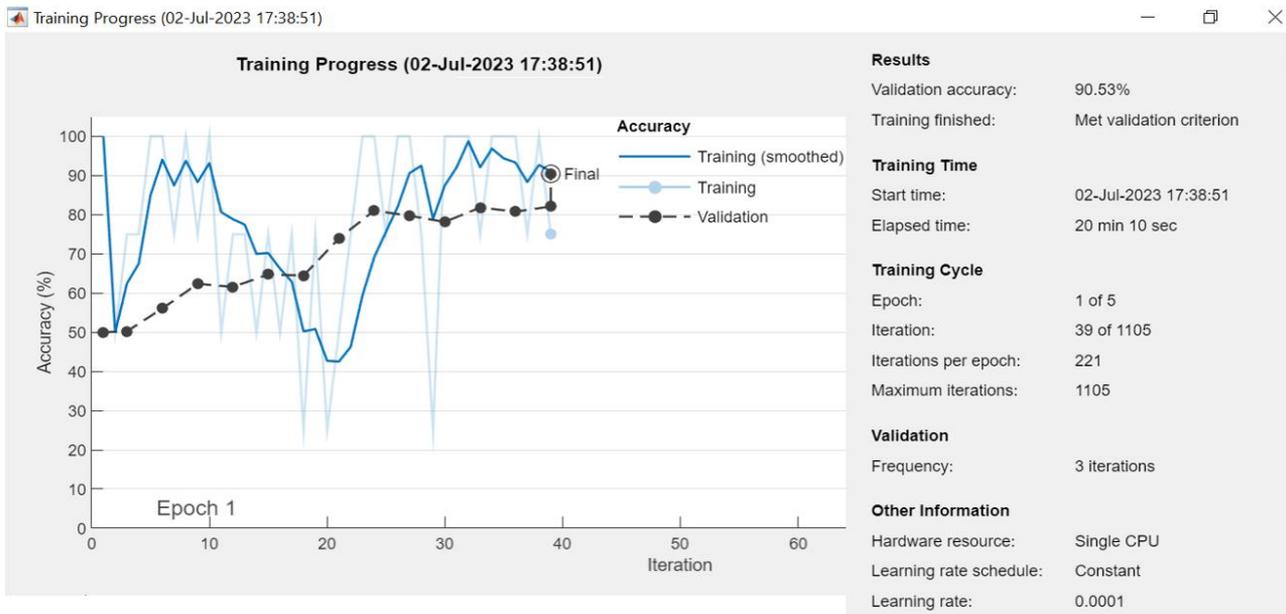


Figure 11. Graph of ResNet-50 scenario test results

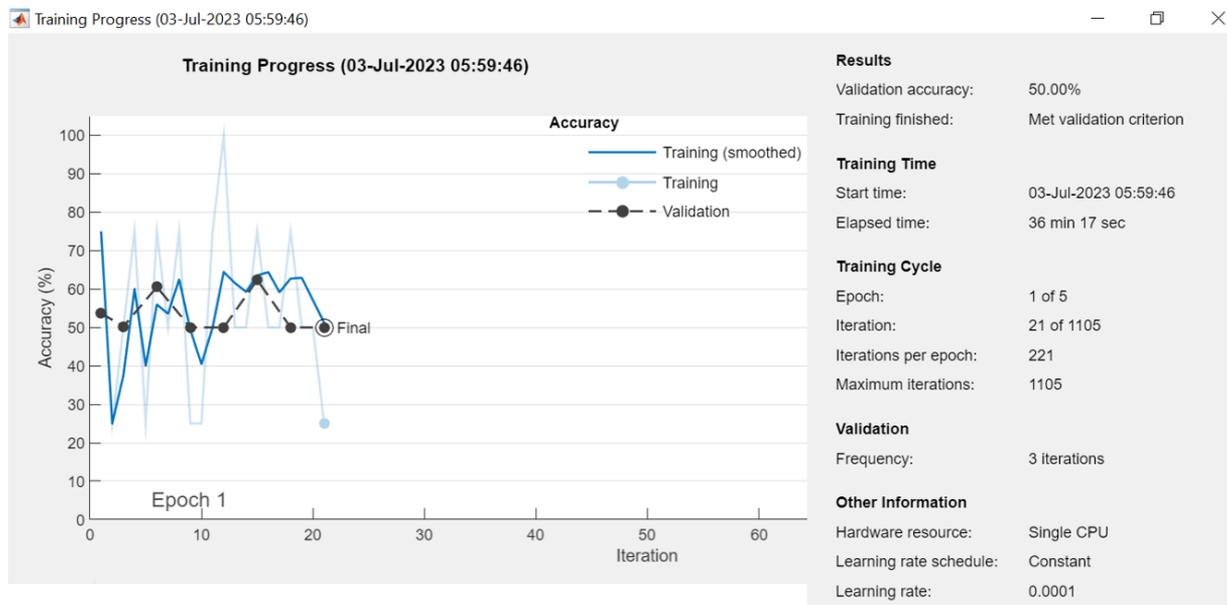


Figure 12. Graph of VGG-16 scenario test results

Furthermore, Figure 10 demonstrates higher performance with an accuracy reaching 93.42% and a testing time of 5 minutes and 51 seconds. These results indicate that AlexNet has remarkable capabilities in classifying images with a very high accuracy rate and relatively short computational time. This makes AlexNet a prominent choice for image classification tasks with high requirements in terms of accuracy and time efficiency.

Meanwhile, Figure 11 shows an accuracy of 90.53% with a testing time of 20 minutes and 10 seconds. Although its accuracy is slightly lower compared to AlexNet, ResNet-50 still demonstrates excellent performance and a reasonable computational time.

However, Figure 12 shows lower performance compared to the other three architectures, with an accuracy of only 50.00% and a testing time of 36 minutes and 17 seconds. These results indicate that VGG-16 may not be the optimal choice for image classification tasks with high requirements in terms of

accuracy and time efficiency.

Table 2 presents the results of scenario testing that compares the performance of different CNN architectures in terms of accuracy and time required for each architecture to process data.

From the results of tests carried out with several CNN models, the results obtained were that ResNet-18 achieved 86.32% accuracy, AlexNet achieved 93.42% accuracy, ResNet-50 achieved 90.53% accuracy and VGG-16 achieved 50.00% accuracy. while the time needed to carry out this test is ResNet-18 takes 11 minutes 23 seconds, AlexNet takes 5 minutes 51 seconds, ResNet-50 takes 20 minutes 10 seconds, VGG-16 takes 36 minutes 17 seconds.

In this test, the highest accuracy results were obtained in the second scenario, namely using the AlexNet architecture with an accuracy of 93.42% and a time of 5 minutes 51 seconds. The following are the results of the scenario test contained in Table 2.

Table 2. Scenario test results

Architecture	Accuracy	Time
ResNet-18	86.32%	11 min 23 sec
AlexNet	93.42%	5 min 51 sec
ResNet-50	90.53%	20 min 10 sec
VGG-16	50.00%	36 min 17 sec

4. CONCLUSION

From the experiments conducted, the results show that the AlexNet architecture has a better classification accuracy rate compared to ResNet-18, ResNet-50, and VGG-16 in the classification of Mycobacterium tuberculosis. The results achieved the best validation accuracy of 93.42% with the fastest testing time of 5 minutes and 52 seconds using the AlexNet architecture. Future research will involve several other Deep Convolutional Neural Network architectures and combining them with various other methods, as well as making changes to parameters to achieve the best results.

REFERENCES

- [1] World Health Organization. (2019). Global tuberculosis report 2019. Geneva, Switzerland: World Health Organization, pp. 1-297.
- [2] Rachmad, A.E.R.I., Chamidah, N., Rulaningtyas, R.I.R. I.E.S. (2020). Mycobacterium tuberculosis identification based on colour feature extraction using expert system. *Ann. Biol*, 36: 196-202.
- [3] Puttagunta, M., Ravi, S. (2021). Medical image analysis based on deep learning approach. *Multimedia Tools and Applications*, 80: 24365-24398. <https://doi.org/10.1007/s11042-021-10707-4>
- [4] Rachmad, A., Chamidah, N., Rulaningtyas, R. (2021). Classification of mycobacterium tuberculosis based on color feature extraction using adaptive boosting method. In *AIP Conference Proceedings*. AIP Publishing, 2329(1). <https://doi.org/10.1063/5.0042283>
- [5] Rochman, E.M.S., Suprajitno, H., Kamilah, I., Rachmad, A., Santosa, I. (2023). Tuberculosis classification using random forest with K-prototype as a method to overcome missing value. *Commun. Communications in Mathematical Biology and Neuroscience*, 2023. <https://doi.org/10.28919/cmbn/7873>
- [6] Mithra, K.S., EMMANUEL, W.S. (2018). FHDT: Fuzzy and Hyco-entropy-based decision tree classifier for tuberculosis diagnosis from sputum images. *Sādhanā*, 43: 1-15. <https://doi.org/10.1007/s12046-018-0878-y>
- [7] Purwanti, E., Widiyanti, P. (2012). Using learning vector quantization method for automated identification of mycobacterium tuberculosis. *Indonesian Journal of Tropical and Infectious Disease*, 3(1): 26-29.
- [8] Arisgraha, F., Widiyanti, P., Apsari, R. (2015). Digital detection system design of Mycobacterium tuberculosis through extraction of sputum image using neural network method. *Indonesian Journal of Tropical and Infectious Disease*, 3(1): 35-38.
- [9] Dinesh Jackson Samuel, R., Rajesh Kanna, B. (2019). Tuberculosis (TB) detection system using deep neural networks. *Neural Computing and Applications*, 31: 1533-1545. <https://doi.org/10.1007/s00521-018-3564-4>
- [10] Ker, J., Wang, L., Rao, J., Lim, T. (2017). Deep learning applications in medical image analysis. *IEEE Access*, 6: 9375-9389. <https://doi.org/10.1109/ACCESS.2017.2788044>
- [11] Munir, K., Elahi, H., Ayub, A., Frezza, F., Rizzi, A. (2019). Cancer diagnosis using deep learning: A bibliographic review. *Cancers*, 11(9): 1235. <https://doi.org/10.3390/cancers11091235>
- [12] He, K., Zhang, X., Ren, S., 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>
- [13] Rachmad, A., Chamidah, N., Rulaningtyas, R. (2019). Image enhancement sputum containing Mycobacterium tuberculosis using a spatial domain filter. In *IOP Conference Series: Materials Science and Engineering*. IOP Publishing, 546(5): 052061. <https://doi.org/10.1088/1757-899X/546/5/052061>
- [14] Yuwono, B. (2015). Image smoothing menggunakan mean filtering, median filtering, modus filtering dan gaussian filtering. *Telematika: Jurnal Informatika dan Teknologi Informatika*, 7(1): 65-75. <https://doi.org/10.31315/telematika.v7i1.416>
- [15] 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>
- [16] 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>
- [17] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553): 436-444. <https://doi.org/10.1038/nature14539>
- [18] Sudiarmika, I.B.K. (2018). Indonesian traditional shadow puppet image classification: A deep learning approach. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, Bali, Indonesia, pp. 130-135. <https://doi.org/10.1109/ICITEED.2018.8534776>
- [19] Rachmad, A., Chamidah, N., Rulaningtyas, R. (2020). Mycobacterium tuberculosis images classification based on combining of convolutional neural network and support vector machine. *Communications in Mathematical Biology and Neuroscience*. <https://doi.org/10.28919/cmbn/5035>
- [20] Gonzalez, T.F. (2007). Handbook of approximation algorithms and metaheuristics. Chapman & Hall/Crc Computer and Information Science Series. https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Handbook%20of%20Approximation%20Algorithms%20and%20Metaheuristics%20%5BGonzalez%202007-01-05%5D.pdf.
- [21] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv Preprint arXiv: 1409.1556*. <https://doi.org/10.48550/arXiv.1409.1556>
- [22] Rachmad, A., Fuad, M., Rochman, E.M.S. (2023). Convolutional neural network-based classification model of corn leaf disease. *Mathematical Modelling of Engineering Problems*, 10(2): 530-536. <https://doi.org/10.18280/mmep.100220>
- [23] Gunawan, R.J., Irawan, B., Setianingsih, C. (2021).

- Pengenalan ekspresi wajah berbasis convolutional neural network dengan model arsitektur vgg16. *eProceedings of Engineering*, 8(5): 6442-6454.
- [24] Damayanti, F., Muntasa, A., Herawati, S., Yusuf, M., Rachmad, A. (2020). Identification of Madura tobacco leaf disease using gray-level Co-occurrence matrix, color moments and Naïve Bayes. In *Journal of Physics: Conference Series*. IOP Publishing, 1477(5): 052054. <https://doi.org/10.1088/1742-6596/1477/5/052054>
- [25] Rachmad, A., Syarief, M., Rifka, S., Sonata, F., Setiawan, W., Rochman, E.M.S. (2022). Corn leaf disease classification using local binary patterns (LBP) feature extraction. In *Journal of Physics: Conference Series*. IOP Publishing, 2406(1): 012020. <https://doi.org/10.1088/1742-6596/2406/1/012020>
- [26] Ubaidillah, A., Rochman, E.M.S., Fatah, D.A., Rachmad, A. (2022). Classification of corn diseases using random forest, neural network, and naive bayes methods. In *Journal of Physics: Conference Series*. IOP Publishing, 2406(1): 012023. <https://doi.org/10.1088/1742-6596/2406/1/012023>
- [27] Solihin, F. (2023). Comparison of support vector machine (SVM), K-Nearest neighbor (K-NN), and Stochastic gradient descent (SGD) for classifying corn leaf disease based on Histogram of oriented gradients (HOG) feature extraction. *Elinvo (Electronics, Informatics, and Vocational Education)*, 8(1). <https://doi.org/10.21831/elinvo.v8i1.55759>