Journal of Electrical Engineering and Computer (JEECOM)
Vol. 8, No. 1 (2026), DOI: 10.33650/jeecom.v8i1.17318
p-ISSN: 2715-0410; e-ISSN: 2715-6427

Evaluation of Google Teachable Machine for Tuberculosis Screening Using Chest X-Ray Images

Vira Meldimira[1], Agus Wantoro[2*], Kraugusteeliana[3], Zulkifli[4]
[1]Department of Medicine, Faculty of Medicine, Aisyah University, Indonesia
[2]Department of Computer Sciences, Faculty of Technology and Informatics, Aisyah University, Indonesia
[3]Information System Department, Universitas Pembangunan Nasional Veteran Jakarta, Indonesia
[4]Department of Informatics Engineering, Faculty of Technology and Informatics, Aisyah University, Indonesia

Article Info

Article history:

Received March 20, 2026
Revised April 25, 2026
Accepted April 29, 2026

Keywords:

Classification Image;
Chest X-ray;
Teachable Machine;
Tuberculosis;

ABSTRACT

Tuberculosis (TB) remains a leading cause of death from infectious diseases worldwide, with Indonesia ranking second. Conventional TB diagnosis, which relies on the manual interpretation of chest X-rays, faces challenges due to a shortage of radiologists, particularly in resource-limited healthcare settings. This study aimed to develop a TB image classification model using Google Teachable Machine a no-code machine learning platform to create an easily deployable screening tool that requires no programming expertise. The model was developed using a transfer learning approach based on the Mobile Net architecture, trained on a dataset of chest X-rays labeled as either TB-positive or normal, and evaluated for accuracy across multiple training epochs. The results showed that both training and testing accuracies converged toward 1.0 after the 10th epoch; the curves moved in parallel without significant performance discrepancies, indicating good model generalization capabilities. Optimal results were achieved with an epoch count of 60, a batch size of 64, and a learning rate of 0.003. These findings demonstrate that the ease of implementation offered by no-code platforms does not necessarily compromise classification performance, making this a potentially practical solution to support early TB detection. Further research is recommended to conduct external validation and assess sensitivity and specificity to confirm the model's suitability as a clinical pre-screening instrument.

Corresponding Author:
Agus Wantoro,
Department of Computer Sciences, Faculty of Technology and Informatics, Aisyah University, Indonesia
Email: aguswantoro@aisyahuniversity.ac.id; Orcid ID : https://orcid.org/0000-0002-0314-7492

1 INTRODUCTION

Tuberculosis (TB) remains one of the most serious global public health threats [1]. The WHO Global Tuberculosis Report 2025 notes that the number of incident TB cases globally declined in 2024 for the first time since 2020, falling from 10.8 million cases in 2023 to 10.7 million cases in 2024 [2]. However, this decline has not been uniformed across the globe. TB remains the leading cause of death from infectious disease worldwide surpassing even Covid-19 with an estimated two deaths occurring every minute [3]. Indonesia ranks second globally in terms of TB burden, after India, accounting for approximately 10% of the total global case load; this underscores the fact that the issue is both highly localized and a matter of urgent national health concern [4].

TB screening generally relies on chest X-ray imaging interpreted manually by trained medical professionals, such as radiologists or pulmonologists [5]. This process has significant limitations because interpretation is subjective, requires specialized expertise, and is prone to error due to the variability of TB radiographic patterns, which often resemble other lung diseases such as pneumonia [6]. In some regions,

particularly those with limited healthcare resources, the availability of radiologists is extremely scarce, resulting in slow and uneven TB screening processes [7].

Advancements in artificial intelligence specifically deep learning based on Convolutional Neural Networks (CNN) have demonstrated significant potential for automating medical image classification with high accuracy [8]. However, implementing conventional deep learning models typically requires programming skills, adequate computing infrastructure, and deep technical expertise posing a genuine challenge for healthcare facilities with limited human resources and IT infrastructure. It is in this context that no-code or low-code machine learning platforms, such as Google Teachable Machine, warrant exploration, as they enable the development of image classification models without the need for complex programming [9].

The urgency of this research is grounded in several crucial facts: the cumulative global incidence of TB has declined by only 8.3% since 2015 far short of the 50% reduction target set for 2025 while the TB mortality rate has decreased by only 23% against a target of 75%. [10]. This indicates that TB elimination strategies still face significant obstacles, particularly regarding early detection and diagnostic coverage [11]. Furthermore, millions of TB cases remain undiagnosed globally each year a gap likely exacerbated by limited access to radiologists in primary healthcare facilities, especially in rural areas or regions with constrained health resources. Furthermore, the economic burden imposed by TB remains substantial, with nearly half of affected households incurring catastrophic costs due to the disease a situation exacerbated by delayed diagnosis [12]. In this context, there is an urgent need to develop and scientifically validate image-based screening tools that are affordable, accessible, and do not require advanced programming expertise, particularly to serve as pre-screening solutions in resource-limited healthcare facilities prior to referral for definitive diagnostic testing.

A review of the literature indicates that deep learning-based tuberculosis (TB) image classification research has advanced rapidly; however, the majority of studies focus on complex CNN architectures requiring advanced programming expertise such as Res Net, Dense Net, MobileNetV2, and the hybrid Ghost Net-Mobile ViT model which, despite achieving high accuracy (>95%), are difficult for non-technical healthcare workers in primary care facilities to replicate [13]. Several studies have explored Google Teachable Machine for TB detection, one study evaluated the ability of Teachable Machine as a deep neural network-based tool to predict TB probability from chest X-ray images, using 348 TB images and 3806 normal images, and achieved an Area Under the Curve (AUC) of 0.951–0.975 in external validation, comparable to pulmonologist interpretation [14]. Another study within the Indonesian context has also utilized Teachable Machine to develop an Android-based application for classifying TB X-ray images; however, it focused on the application development aspect without thoroughly evaluating the model's suitability for real-world implementation in resource-limited healthcare facilities [15].

Based on the review above, three key gaps have been identified: (a) limited research specifically positioning and evaluating Teachable Machine as a practical screening tool for primary healthcare settings in regions with shortages of radiologists and computing infrastructure; (b) a scarcity of studies testing Teachable Machine model performance on TB image datasets within the epidemiological context of high-TB-burden countries like Indonesia; and (c) a lack of comprehensive discussion regarding the trade-off between ease of implementation (no-code) and model accuracy, sensitivity, and specificity—factors that are crucial considerations before recommending this tool for use as a pre-screening instrument in the field.

This study has several objectives: (1) To implement a tuberculosis image classification model using the Google Teachable Machine platform based on chest X-ray images. (2) To evaluate the performance of the resulting model using metrics such as accuracy, sensitivity, specificity, and area under the curve (AUC) to assess its suitability as a screening tool. (3) To analyze the potential and limitations of applying Teachable Machine as an image-based TB screening aid in resource-limited healthcare settings. (4) To provide practical recommendations regarding the feasibility of implementing this no-code/low-code AI model to support early TB detection efforts, particularly in areas with limited access to radiologists.

2 METHOD

2.1 Research Design

This study employs a quantitative experimental approach with an applied research design, aiming to develop and evaluate a tuberculosis (TB) image classification model based on the Google Teachable Machine platform. The research process follows standard Machine Learning model development stages data collection, preprocessing, model training, testing, and performance evaluation, and subsequently analyzes the model's feasibility as a pre-screening tool. The research design is illustrated in Figure 1.

ISSN: 2715-6427

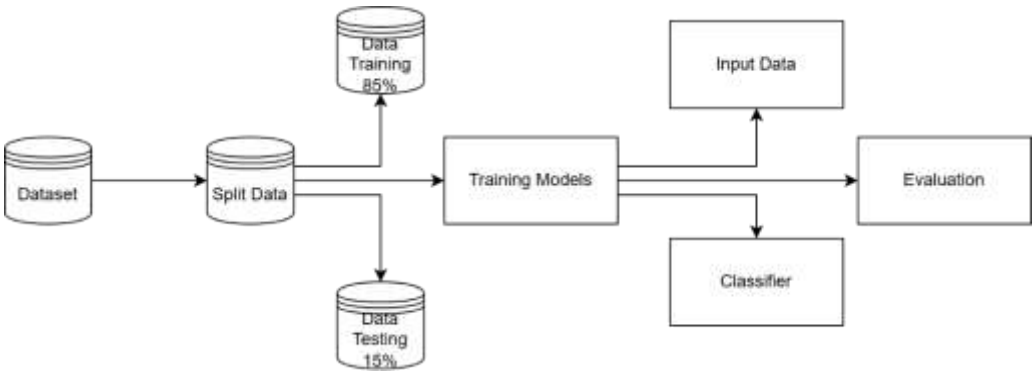
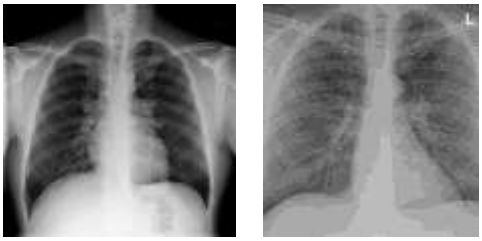


Figure 1. Research design

2.2 Data sources

The chest X-ray dataset used in this study is sourced from a medically validated public repository [16]. The dataset from the www.kaggle.com platform, which is widely used in similar studies. It comprises two main classes: TB-positive images and normal (non-TB) images [17]. The dataset contains a total of 1200 records, split equally into 600 TB-positive records and 600 normal records. Sample data are shown in Figure 2.



(a) Normal (b) TB-Positive

Figure 2. Sample dataset: (a) Normal (non-TB), and (b) TB-Positive

2.3 Split Data

Data splitting is the process of dividing a single dataset into two parts for the purpose of training and testing machine learning models [18]. The training data accounts for 75% of the data, while the testing data accounts for 15%. The dataset split is illustrated in Figure 3.

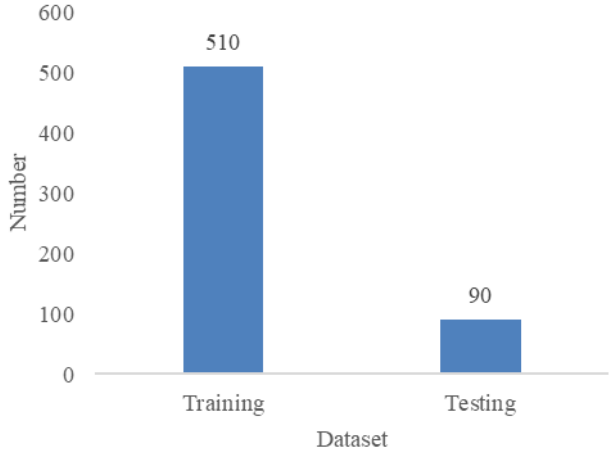


Figure 3. Number of data training, and data testing

2.4 Google Teachable Machine

Google Teachable Machine is a web-based, no-code machine learning platform that utilizes transfer learning from the Mobile Net architecture [19]. The model development stages include:

1. Upload the TB and normal image datasets to the Teachable Machine interface, grouped according to class labels.
2. Configure training parameters, including the number of epochs, batch size, and learning rate.
3. Train the model directly in the browser using TensorFlow.js, without the need for library installations or writing code.

2.5 Model Evaluation

An evaluation was conducted to determine the model performance associated with the optimal combination of parameters for this technique. This evaluation compared scenarios involving different numbers of epochs, batch sizes, and learning rates. Subsequently, the model underwent training, and its performance was assessed based on accuracy and epoch loss.

3 RESULTS AND DISCUSSION

The model used in this machine learning application is Mobile Net. The initial test employed a combination of 50 epochs, a batch size of 16, and a learning rate of 0.001. The confusion matrices for the model's performance evaluation are presented in Figure 4, and Figure 5.

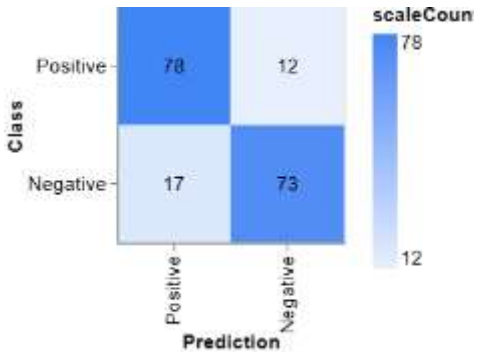


Figure 4. Confusion matrix for the first scenario

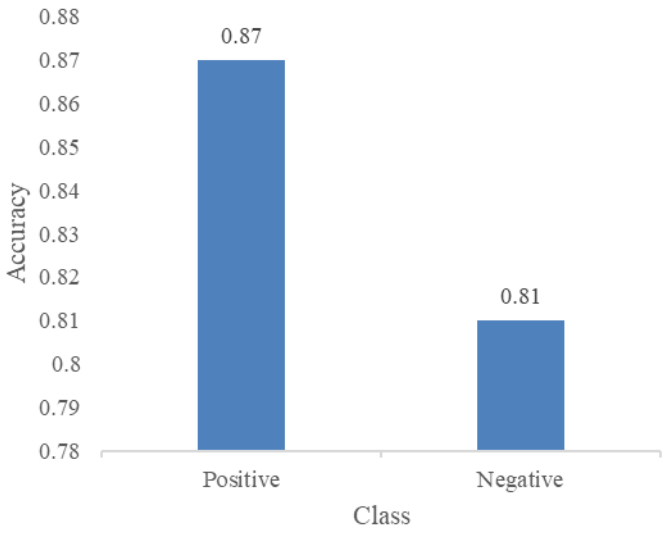


Figure 5. Performance for each class is based on the first scenario.

ISSN: 2715-6427

Figure 5 illustrates varying performance results across the different classes. The model demonstrates better predictive performance for the positive class compared to the negative class. The model's accuracy for this combination is 0.84, or 84%. We further compare the performance and loss per epoch, as presented in Figure 6.

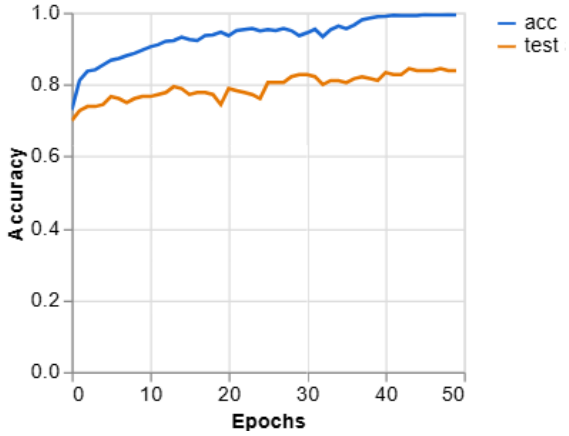


Figure 6. Accuracy per epoch in the first scenario

Figure 6 shows that training accuracy consistently increased from approximately 80% to nearly 100% by the final epoch. Testing accuracy also improved rising from around 70%-75% to approximately 84% despite some fluctuations during the training process. The widening gap between training and testing accuracy in the final epochs indicates potential overfitting. Consequently, increasing the number of epochs beyond the 30–40 range did not yield significant improvements in testing performance; therefore, implementing early stopping or regularization methods could be considered to enhance the model's generalization. The loss per epoch is shown in Figure 7.

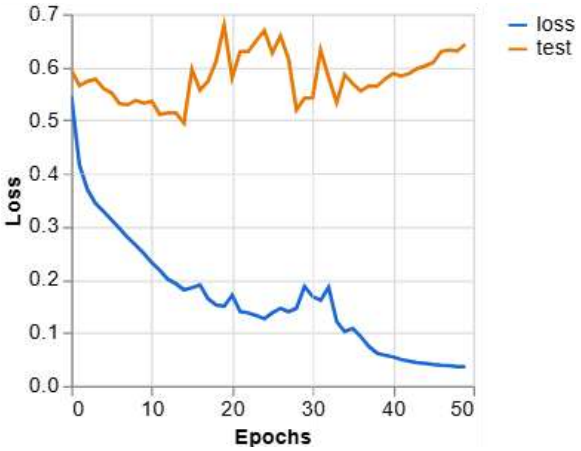


Figure 7. Loss per epoch in the first scenario

Based on the loss-versus-epoch graph (Figure 7), the training loss decreased consistently from approximately 0.55 at the start of training to around 0.04 by the 50th epoch. This indicates that the model was becoming increasingly adept at fitting the training data. Conversely, the testing loss tended to fluctuate within the 0.5–0.7 range, rising again to approximately 0.64 by the final epoch. The widening gap between training and testing loss suggests overfitting, as the model improved on the training data without a corresponding improvement in performance on the testing data. For subsequent testing, we utilized a configuration of 55 epochs, a batch size of 32, and a learning rate of 0.002. The confusion matrices illustrating the model's performance are presented in Figure 8 and Figure 9.

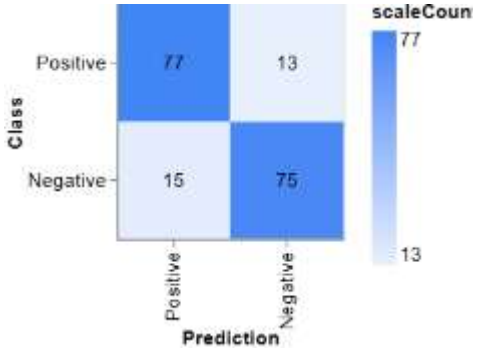


Figure 8. Confusion matrix for the second scenario

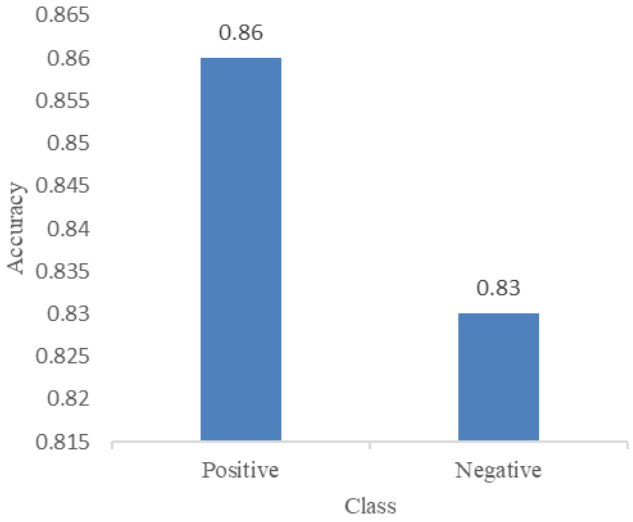


Figure 9. Performance for each class is based on the second scenario.

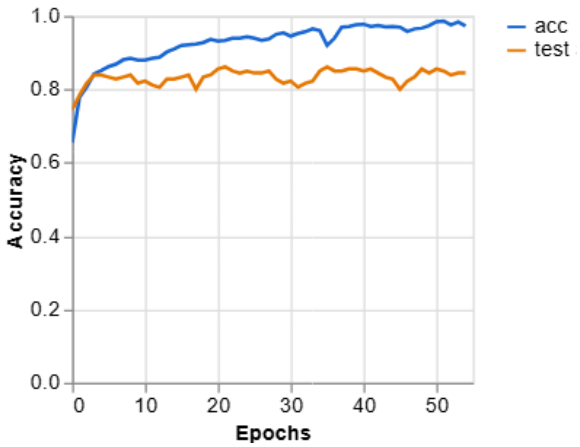


Figure 10. Accuracy per epoch in the second scenario

Figure 10 shows that training accuracy increased gradually from approximately 65% to nearly 98% by the end of training. Meanwhile, testing accuracy rose to around 83–85% but tended to fluctuate after several epochs. The widening gap between training and testing accuracy indicates overfitting, where the model excels at learning the training data but shows relatively limited performance improvement on the testing data. The loss per epoch for the second scenario is shown in Figure 11.

ISSN: 2715-6427

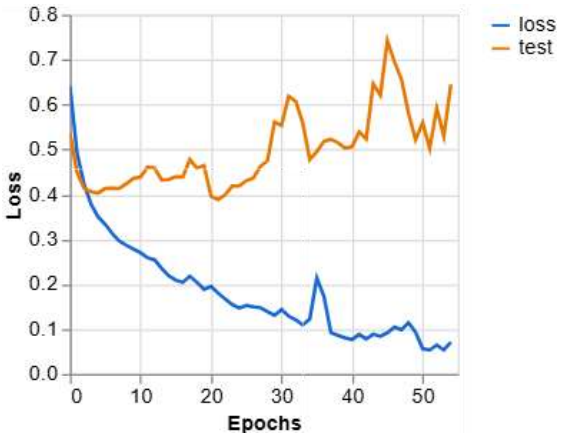


Figure 11. Loss per epoch in the second scenario

Figure 10 illustrates the progression of training loss and test loss values over 55 epochs. The training loss consistently decreased from approximately 0.65 at the start of training to less than 0.10 by the final epoch, indicating that the model was increasingly capable of learning patterns within the training data. However, a different pattern was observed for the test loss; after decreasing to around 0.40 during the initial epochs, it subsequently showed an increase and significant fluctuations, eventually reaching a value of approximately 0.74.

The diverging trends between training loss and test loss indicate overfitting a condition where the model improves at learning the training data but its ability to generalize to unseen data declines. Therefore, model selection should not be based on the lowest training loss, but rather on the epoch that yields the minimum validation or test loss. Techniques such as early stopping, regularization, dropout, and learning rate adjustment can be considered to enhance the model's generalization capability. Subsequently, we conducted testing (Scenario 3) using 60 epochs, a batch size of 64, and a learning rate of 0.003. The model's performance results are presented in the confusion matrix in Figure 12 and the performance metrics in Figure 13.

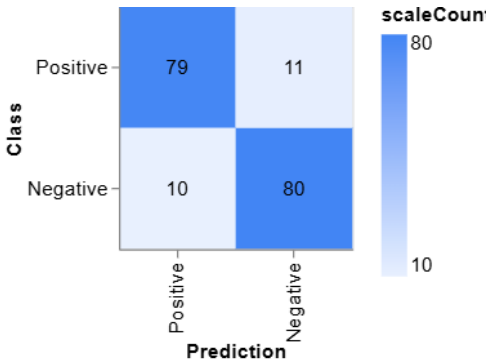


Figure 12. Confusion matrix for the third scenario

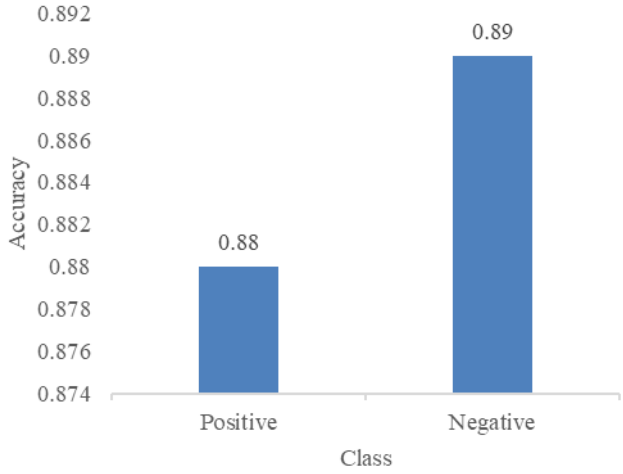


Figure 13. Performance for each class is based on the third scenario.

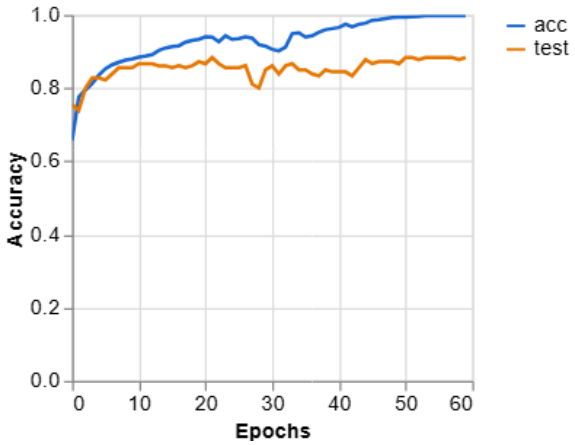


Figure 14. Accuracy per epoch in the third scenario

Figure 14 shows that training accuracy consistently increased from approximately 0.67 to nearly 1.00 by the end of the epochs. Meanwhile, test accuracy rose during the initial phase to around 0.85-0.88 but subsequently tended to stabilize and fluctuate within that range. The widening gap between training accuracy and test accuracy in the final epochs indicates potential overfitting, where the model performs exceptionally well on training data but shows relatively limited performance gains on test data. Overall, the model achieved reasonably stable test accuracy; however, regularization or early stopping could be considered to improve generalization capabilities. The loss per epoch for the third scenario is shown in Figure 15.

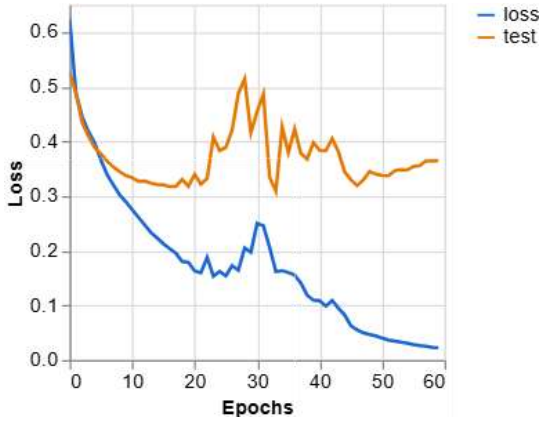


Figure 15. Loss per epoch in the third scenario

ISSN: 2715-6427

Figure 15 shows that the model learns effectively, but tends to overfit starting around epoch 20; thus, increasing the number of epochs does not always yield better results. In this case, the model at approximately epoch 18–20 is likely a more optimal candidate than the model at epoch 60. Subsequently, we conducted tests using 50 epochs, a batch size of 32, and a learning rate of 0.002. The model's performance results for this scenario are presented in the confusion matrix in Figure 16, and the performance metrics are shown in Figure 17.

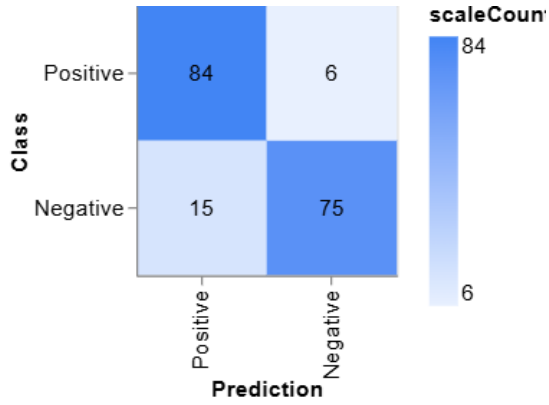


Figure 16. Confusion matrix for the fourth scenario

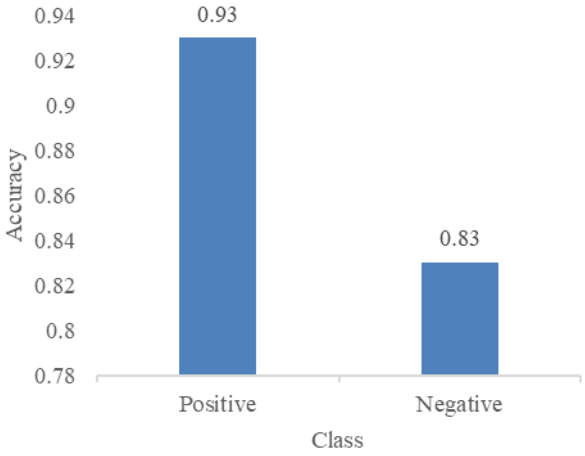


Figure 17. Performance for each class is based on the fourth scenario

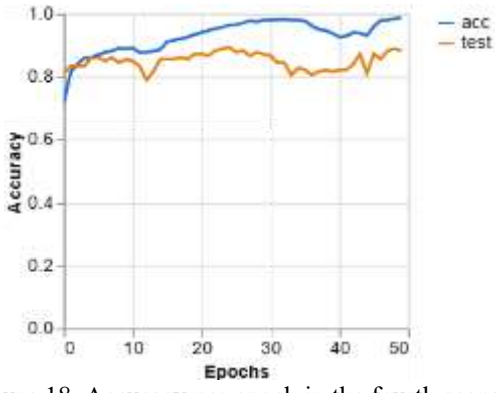


Figure 18. Accuracy per epoch in the fourth scenario

Figure 18 shows that, over the course of 50 epochs, training accuracy (acc) exhibits a consistent upward trend, rising from approximately 0.75 at the start of training to nearly 0.98–1.00 by the final epoch.

Meanwhile, validation/testing accuracy (test acc) tends to hover between 0.80 and 0.89, showing fluctuations throughout the process. The widening gap between training and testing accuracy in the final epochs indicates overfitting; the model improves at recognizing training data, yet its performance gains on unseen data remain relatively limited. Consequently, the model's performance on test data warrants attention, and techniques such as early stopping, dropout, or regularization could be considered to enhance the model's generalization. Loss per epoch show in Figure 19.

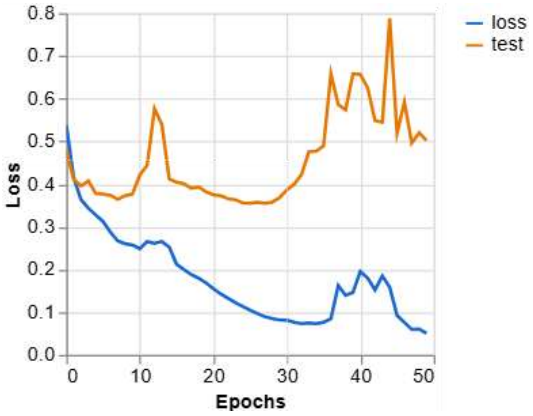


Figure 19. Loss per epoch in the fourth scenario

Figure 19 illustrates the training and test losses over 50 epochs; the training loss generally declined from approximately 0.53 to 0.05, indicating the model's increasing ability to fit the training data. Meanwhile, the test loss initially decreased to around 0.36 between epochs 25 and 30 but subsequently rose and fluctuated significantly, reaching approximately 0.78. The widening gap between training and test losses in the final epochs indicates overfitting, where the model performs increasingly well on training data while its performance on unseen data tends to deteriorate. Consequently, the model likely exhibited better generalization performance around epochs 25–30, prior to the rise in test loss. A comparison of the test scenarios is presented in Table 1.

Tabel 1. Performance comparison using several test scenarios

Epoch	Batch Size	Learning Rate	Accuracy (%)
50	16	0.001	84
55	32	0.002	84.5
60	64	0.003	88.5
50	32	0.002	88

The table shows that in this case, the best combination is Epoch = 60, Batch Size = 64, and Learning Rate = 0.003. This is consistent with previous research [20] that this combination can be an option when testing the model

4 CONCLUSION

This study aims to evaluate a tuberculosis (TB) image classification model using the no-code machine learning platform Google Teachable Machine, as an effort to provide a chest X-ray-based TB screening tool that is easily implementable in resource-limited healthcare facilities

Based on the model training and testing results, the following conclusions can be drawn: (a) A Teachable Machine-based TB classification model was successfully developed, demonstrating accuracy curves that converged near 1.0 for both training and test data after the 10th epoch; the curves moved in parallel without a significant performance gap up to the 50th epoch, indicating that the model did not overfit and possessed good generalization capabilities on the dataset used. (b) The Teachable Machine platform proved technically viable for developing TB image classification models without requiring programming expertise, aligning with the research objective of providing an accessible AI solution for non-technical healthcare workers in primary care facilities, particularly in regions with a shortage of radiologists. (c) The ease of implementation (no-code approach) did not compromise model performance (d) These findings support the argument that democratizing AI via no-code platforms offers a potential practical solution for

narrowing the TB diagnostic gap especially in high-burden countries like Indonesia by serving as an initial pre-screening tool prior to referral for definitive diagnostic testing.

Overall, this study provides preliminary evidence that Google Teachable Machine could be a viable alternative for supporting image-based early TB detection in resource-limited healthcare settings, while also paving the way for more comprehensive future research. Further evaluation using sensitivity, specificity, and confusion matrix metrics is required to ensure the model is not biased toward the majority class, given the significant clinical consequences of false-negative results in TB diagnosis.

ACKNOWLEDGEMENTS

The author extends gratitude to all parties who provided support, assistance, and contributions to the research titled "Evaluation of Google Teachable Machine for Tuberculosis Screening Using Chest X-Ray Images." Special thanks are conveyed to Universitas Aisyah Pringsewu, as well as to everyone who offered guidance, input, and support throughout the research process and the preparation of this article. The author also appreciates those who assisted in providing and managing the X-ray image data used in the study. Support from various parties was invaluable to the development of this research, particularly regarding the application of Machine Learning technology as an approach to facilitate X-ray image-based tuberculosis screening.

REFERENCES

[1] J. Chakaya *et al.*, "Global Tuberculosis Report 2020 – Reflections on the Global TB burden, treatment and prevention efforts," *Int. J. Infect. Dis.*, vol. 113, pp. S7–S12, 2021, doi: https://doi.org/10.1016/j.ijid.2021.02.107.

[2] H. Lee, J. Kim, J. Kim, and Y.-J. Park, "[Review of the Global Burden of Tuberculosis in 2023: Insights from the WHO Global Tuberculosis Report 2024].," *Jugan geon-gang gwa jilbyeong*, vol. 18, no. 11 Suppl, pp. 55–69, Mar. 2025, doi: 10.56786/PHWR.2025.18.11suppl.5.

[3] D. Zenner *et al.*, "State of the global tuberculosis epidemic: burden, progress towards WHO End TB Strategy targets and systemic challenges," *The Lancet Microbe*, Sep. 2026, doi: 10.1016/j.lanmic.2026.101433.

[4] T. Berida and C. W. Lindsley, "Move over COVID, Tuberculosis Is Once again the Leading Cause of Death from a Single Infectious Disease," *J. Med. Chem.*, vol. 67, no. 24, pp. 21633–21640, Dec. 2024, doi: 10.1021/acs.jmedchem.4c02876.

[5] I. Kontsevaya *et al.*, "Update on the diagnosis of tuberculosis," *Clin. Microbiol. Infect.*, vol. 30, no. 9, pp. 1115–1122, 2024, doi: https://doi.org/10.1016/j.cmi.2023.07.014.

[6] I. E. Dror, "Cognitive and Human Factors in Expert Decision Making: Six Fallacies and the Eight Sources of Bias," *Anal. Chem.*, vol. 92, no. 12, pp. 7998–8004, Jun. 2020, doi: 10.1021/acs.analchem.0c00704.

[7] A. O'Hagan, "Expert Knowledge Elicitation: Subjective but Scientific," *Am. Stat.*, vol. 73, no. sup1, pp. 69–81, Mar. 2019, doi: 10.1080/00031305.2018.1518265.

[8] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," 2025. doi: 10.3390/info16030195.

[9] M. Jibril, "Development of a No-Code Machine Learning Model Builder for Predictive Analytics in Education," *Comput. Appl. Eng. Educ.*, vol. 33, no. 6, p. e70088, Nov. 2025, doi: https://doi.org/10.1002/cae.70088.

[10] A. Bhargava, M. Bhargava, and A. Juneja, "Social determinants of tuberculosis: context, framework, and the way forward to ending TB in India," *Expert Rev. Respir. Med.*, vol. 15, no. 7, pp. 867–883, Jul. 2021, doi: 10.1080/17476348.2021.1832469.

[11] H. Shah, J. Patel, S. Rai, and A. Sen, "Advancing tuberculosis elimination in India: A qualitative review of current strategies and areas for improvement in tuberculosis preventive treatment," *IJID Reg.*, vol. 14, p. 100556, 2025, doi: https://doi.org/10.1016/j.ijregi.2024.100556.

[12] R. M. Ghazy *et al.*, "A systematic review and meta-analysis of the catastrophic costs incurred by tuberculosis patients," *Sci. Rep.*, vol. 12, no. 1, p. 558, 2022, doi: 10.1038/s41598-021-04345-x.

[13] M. Patel, A. Das, V. K. Pant, and M. J., "Detection of Tuberculosis in Radiographs using Deep Learning-based Ensemble Methods," in *2021 Smart Technologies, Communication and Robotics (STCR)*, 2021, pp. 1–7. doi: 10.1109/STCR51658.2021.9588936.

[14] C.-F. Chen *et al.*, "A deep learning-based algorithm for pulmonary tuberculosis detection in chest radiography," *Sci. Rep.*, vol. 14, no. 1, p. 14917, 2024, doi: 10.1038/s41598-024-65703-z.

[15] A. Mittal, B. Madke, A. Singh, A. Bagde, B. Ghewade, and H. V., "Prediction of Pulmonary Tuberculosis From Chest X-Ray Using a Machine Learning Model Trained With Google's Teachable

Machine Software," in *2025 3rd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIHEI)*, 2025, pp. 1–6. doi: 10.1109/IDICAIHEI65991.2025.11379085.

[16] S. Tabik *et al.*, "COVIDGR Dataset and COVID-SDNet Methodology for Predicting COVID-19 Based on Chest X-Ray Images," *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 12, pp. 3595–3605, 2020, doi: 10.1109/JBHI.2020.3037127.

[17] 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.

[18] A. A. Shujaaddeen, F. M. B. -Alwi, A. T. Zahary, and A. S. Alhegami, "A Model for Measuring the Effect of Splitting Data Method on the Efficiency of Machine Learning Models: A Comparative Study," in *2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2024, pp. 1–13. doi: 10.1109/eSmarTA62850.2024.10639022.

[19] A. K. Shrivastava, P. Tiwari, A. K. Dave, and G. Snadya, "AI - driven disease detection in guava plants using teachable machine learning models," *Plant Sci. Today*, vol. 13, no. 1, pp. 1–8, 2026, doi: https://doi.org/10.14719/pst.11037 eISSN.

[20] Z. Ma, Y. Xu, H. Xu, Z. Meng, L. Huang, and Y. Xue, "Adaptive Batch Size for Federated Learning in Resource-Constrained Edge Computing," *IEEE Trans. Mob. Comput.*, vol. 22, no. 1, pp. 37–53, 2023, doi: 10.1109/TMC.2021.3075291.