RESEARCH ARTICLE **OPEN ACCESS**

Manuscript received September 9, 2024; October 12, 2024; October 14, 2024; date of publication November 27, 2024
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i1.572

How to cite: Qurrata A'yuni, Nasaruddin Nasaruddin, Muhammad Irhamsyah, Mulkan Azhary, and Roslidar Roslidar, "Intelligent Tuberculosis Detection System with Continuous Learning on X-ray Images", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 1, pp. 130-141, January 2025.

Intelligent Tuberculosis Detection System with Continuous Learning on X-ray Images

Qurrata A'yuni[1], Nasaruddin Nasaruddin[1], Muhammad Irhamsyah[1], Mulkan Azhary[2], and Roslidar Roslidar[1]

[1] Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia
[2] Faculty of Medicine, Universitas Syiah Kuala, Banda Aceh, Indonesia

Corresponding author: Roslidar Roslidar, email: roslidar@usk.ac.id.

This work was supported by the Institute for Research and Community Services (LPPM) of Universitas Syiah Kuala, under Grant No. 427/UN11.2.1/PG.01.03/SPK/PTNBH/2024.

ABSTRACT Tuberculosis (TB) has become a global health threat with millions of cases each year. Therefore, rapid and accurate detection is needed to control its spread. The application of artificial intelligence, especially Deep Learning (DL), has shown great potential in improving the accuracy of TB detection through DL-based X-ray image analysis. Although many studies have developed X-ray image classification models, very few have integrated them into web or mobile platforms. In addition, the models integrated into these platforms generally do not apply continuous learning methods so that model performance cannot be updated. Thus, it is necessary to build an intelligent system based on a web application that integrates the ResNet-101 model for TB detection in X-ray images. This system utilizes continuous learning methods, allowing the model to automatically update itself with new data, thereby improving detection performance over time. The results showed that before continuous learning, the model successfully classified all TB images correctly, but was only able to classify two normal images correctly, resulting in an accuracy of 62.5%. After manual continuous learning, the model showed an increase in accuracy to 71.4%, with better ability to recognize normal images, although there was a slight decrease in performance in detecting TB.

INDEX TERMS Tuberculosis, intelligent system, deep learning, ResNet-101, continuous learning.

I. INTRODUCTION

Tuberculosis (TB) is a pulmonary disease transmitted through the air and caused by Mycobacterium tuberculosis [1]. Despite control efforts, TB remains a leading cause of death from infectious diseases worldwide [2]. In 2021, nearly half of the global TB cases were reported in Southeast Asia, with 4.82 million cases accounting for approximately 45.4%. Eight countries contributed to about 66% of the total global cases, with Indonesia (9.2%) ranking second after India. Additionally, Indonesia had 969,000 TB cases in 2021, according to the Global TB Report 2022 [3]. The high incidence of tuberculosis is influenced by various socio-economic factors that continue to play a role in its spread within communities. These factors include malnutrition, inadequate housing conditions, financial limitations, and restricted access to healthcare services. Additionally, other aspects, such as food security and smoking habits, also serve as important determinants in the transmission of TB [4].

Therefore, accurate and rapid TB detection is crucial to prevent its progression and control its spread [5]. Several studies have explored the application of artificial intelligence (AI) in TB detection, employing various methodologies, including machine learning and deep learning models. These studies have demonstrated promising results, yet there remains a need for continuous adaptation of models to better reflect real-world conditions, especially in dynamic and evolving healthcare environments. However, with the rapid advancement of technology in various fields, including healthcare, there is a promising potential to significantly impact the quality of services in hospitals. This technological transformation can comprehensively change the landscape of medical services, enhance TB detection capabilities, and accelerate the response to identified cases, offering a hopeful outlook for the future of healthcare [6].

One of the challenges in today's healthcare system is achieving a balance between the increasing amount of documented data on one hand and demographic changes and an aging population on the other. Although this puts pressure on healthcare services, there is significant potential through the utilization of big data and artificial intelligence in medicine [7]. The use of artificial intelligence (AI) in the medical field opens up huge opportunities to overcome the challenges of the modern health system. AI is able to process and analyze an ever-increasing amount of medical data, such as electronic medical records, test results, medical images, and sensor data, thereby facilitating pattern identification, disease prediction, and faster and more accurate diagnosis. In this way, AI can reduce operational burdens, improve the accuracy of medical decisions, and provide more efficient care, even amidst pressures from changing demographics and data usage [8].

The implementation of this technology has the potential to alleviate the burden on doctors, considering the increasing complexity of current healthcare services and the information overload that must be managed in patient care [7]. Over the past decade, research in the fields of medicine and biomedicine, along with a significant volume of publications, has seen substantial growth. These research outcomes have created promising advances in the development of artificial intelligence, particularly in machine learning algorithms in general [9]. However, a more in-depth examination of recent AI applications in TB detection reveals limitations in existing methodologies. Studies such as those by S. Lakhani and S. Sundaram [10], F. Pasa et al. [11], and S. I. Lopes and J. F. Valiati [12] have employed deep learning models, including Convolutional Neural Networks (CNNs) and ResNet-50, to achieve high accuracy in TB detection using chest X-ray images. While these models have demonstrated strong performance, they rely on static approaches that cannot adapt to new data or evolving disease patterns, limiting their long-term effectiveness. By addressing these gaps, the current study aims to enhance the robustness and applicability of AI in TB detection through continuous learning approaches.

Recent studies have explored deep learning approaches for detecting TB from chest X-rays. Convolutional Neural Networks (CNNs) are widely used for this task, with various architectures such as VGG-16, AlexNet, and GoogLeNet showing promising results [13], [11]. These models have achieved high accuracy, with one study reporting 99.76% accuracy using VGG-16 [13] and another study achieving an AUC of 0.99 using a combination of AlexNet and GoogLeNet [11]. Another study using a CNN model with Adam optimization reached 84% accuracy on a test dataset consisting of 663 images [14]. However, most of these approaches rely on static models that do not account for new data or evolving disease patterns. This fact may cause significant limitations in terms of their real-world application, as these models are not equipped for continuous adaptation to changing environments. While the results achieved by these various models are quite promising, most of these studies have significant limitations. The approaches used generally produce static models and do not take into account the concept of continuous learning. As a result, these models cannot adapt to the addition of new data or changes in disease patterns over time, which can affect overall detection accuracy.

In this context, our study addresses this gap by deploying a deep learning model for TB detection, specifically ResNet101, which was trained on X-ray images and achieved an accuracy rate of 99.2%. Unlike previous static models, our approach incorporates a novel continuous learning method, allowing the model to be regularly updated with new image data to maintain high accuracy as data evolves. This contribution fills the gap in existing research, which primarily focuses on static models that do not adapt to new data over time. Our hypothesis is that continuous learning will allow the system to outperform static models by consistently integrating new information, thus providing more reliable and adaptive performance in TB detection. Following are the key contributions of our work:

1. Presenting the development of an intelligent web-based system of tuberculosis detection using X-ray images.
2. Describing the process of deploying the ResNet-101 model on an intelligent web-based system using the DigitalOcean and Docker cloud platforms.
3. Explaining the implementation of a novel continuous learning method that allows the model to be periodically updated with new image data from users.
4. Exposing the performance of the system's practical usage on real x-ray images to ensure the potential impact on public health.

The rest of this article describes our work and findings. Section II presents a related works of tuberculosis. Section III discusses the proposed methods, including the design, tools, datasets, and implementation of continuous learning. Finally, Section IV highlights the research findings, including the performance and accuracy of the implemented system.

II. RELATED WORKS

Previous research has shown the effectiveness of deep learning algorithms of convolutional neural networks (CNNs) in detecting tuberculosis (TB) from chest X-ray images. Puttagunta and Ravi [15] demonstrated that models such as AlexNet, GoogleNet, ResNet, and DenseNet achieved high accuracy in detecting TB using transfer learning techniques on public datasets like TBX11K [16] and JSRT [17]. Their research results highlight the significant potential of deep learning in computer-aided diagnosis (CAD) systems for improving TB detection in chest radiographs.

In addition, Nafisah and Muhammad [18] developed an automatic TB detection system using a deep learning model equipped with image segmentation and explainable artificial intelligence (XAI). CNN models like EfficientNetB3 achieved up to 99.1% accuracy on several public datasets. Their research results show that using segmented X-ray images provides better performance compared to raw images.

In another study, Akbari et al. [19] developed a CNN model that achieved 97% accuracy in detecting TB from X-ray images. This model was trained on a comprehensive dataset from Qatar University and the University of Dhaka, and used data augmentation to improve the model's robustness. The study highlights the potential of AI to enhance diagnostic accuracy and speed, and assist in early detection and treatment of TB.

Research by Guo et al. [20] demonstrated improved accuracy in detecting TB-related abnormalities and specific manifestations through techniques such as fine-tuning models and ensemble methods. They used class activation mapping techniques for visual interpretation, which helped in localizing abnormalities in chest X-ray images. This approach provided significant improvements in TB detection by leveraging the ability of CNNs to identify and visualize suspicious areas. This research emphasizes the importance of developing more advanced deep learning models for more accurate TB diagnosis.

From this related work, it is clear that existing methods have several limitations. To address these limitations, a continuous learning-based approach is proposed to improve the performance of tuberculosis detection from X-ray images, as many models have been built but have not considered continuous adaptation and updating of the model.

III. METHOD

The proposed intelligent system was developed using an experimental design as illustrated in FIGURE 1, which shows a flowchart of the process of building a web-based intelligent system. The first step is to develop a ResNet-101 model, which is used as the inference model to perform image classification in intelligent systems. After the ResNet-101 model is built, the next step is to create a continuous learning system that allows the model to be continuously improved over time. Next, we build a web application that serves as a user interface for this intelligent system. The model and continuous learning system are then deployed to the cloud to take advantage of greater scalability and flexibility. Last, we integrated the continuous learning systems, models, and web applications and proceeded with the testing to ensure that all components worked well. When testing is successful, then the system automatically proceeds with continuous learning to improve its performance independently based on the new data.

A. DATASET

Tuberculosis is one of the oldest diseases known to affect humans and remains a leading cause of death worldwide. It is caused by the bacterium Mycobacterium tuberculosis [21]. Tuberculosis is primarily transmitted through the air. Additionally, the disease can spread through direct contact with an active patient, especially via small droplets produced when the patient coughs or sneezes [22]. Generally, chest X-rays (CXR), as shown in FIGURE 2, will show white spots in the lungs, which indicate the presence of TB, and are useful for assessing the extent of disease spread [23].

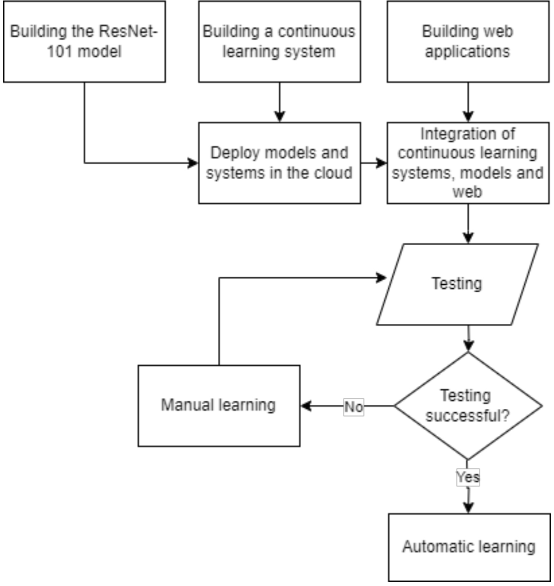

Building the ResNet-101 model
Building a continuous learning system
Building web applications
Deploy models and systems in the cloud
Integration of continuous learning systems, models and web
Testing
Testing successful?
No
Manual learning
Yes
Automatic learning

FIGURE 1. **Flowchart of the process of developing a web-based intelligent system.**

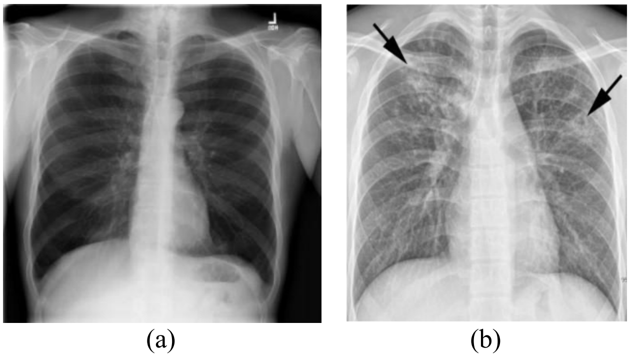


FIGURE 2. **Chest X-ray visualizations (a) normal** [24] **and (b) tuberculosis** [25].

The dataset used in the study to build the model consists of 326 CXR images for each class, TB and normal, taken from the Montgomery [24], Belarus [26], and RSNA [27] to develop the inference model. The choice of 326 images per class was made to maintain consistency across research stages, as this number was used in a prior study. For training the ResNet-101 model in the previous study, we used 200 epochs with a batch size of 4, a learning rate of 0.001, and the SGD optimizer with a momentum of 0.9. Meanwhile, in developing the intelligent system, we used secondary datasets from a non-public source consisting of 11 TB images and 12 normal images. These datasets were obtained and utilized in compliance with ethical guidelines. Given the nature of the medical data, particular attention was paid to ensuring that all images were anonymized to protect the privacy of the individuals involved. Informed consent had been obtained from all subjects as part of the data collection process for the original study. Of the total dataset, 8 images per class were used in the continuous learning process, while the remaining

3 TB images and 4 normal images were reserved for system testing.

In this work, the image preprocessing process was carried out to adjust the data with the convolutional network architectures. First, all images are resized to 224×224 pixels using the ImageDataGenerator function from TensorFlow to ensure uniformity in input dimensions for the convolutional neural network. Additionally, the pixel values are normalized by dividing each pixel value by 255, changing the value range from 0-255 to 0-1. This step ensures that the input to the model is within a normalized range, which is commonly used in deep learning models. All images are processed in RGB format with three color channels, consistent with the input requirements of the ResNet-101 model, which is initialized using weights from ImageNet. The dataset is divided into three parts: 60% for training, 20% for validation, and 20% for testing, with each set consisting of two classes: normal and TB. This division is explicitly carried out using the appropriate directory structure through the flow_from_directory function in ImageDataGenerator.

B. MODEL DEVELOPMENT

The ResNet-101 network consists of 101 layers and is a deep convolutional neural network as shown in TABLE 1. Its design is based on the VGG-19 model and is one of the most complex structures proposed for the ImageNet competition. In a convolutional neural network (CNN), multiple layers are interconnected and trained to perform various tasks. ResNet-N learns different levels of features through its layers. Typically, the convolutional layers of the model have 33 filters. One of the aspects that makes ResNet stand out is that each layer has the same number of filters to maintain the same size of the output feature maps. If the size of the output feature maps is reduced by half, the number of filters is doubled, thus ensuring that the temporal complexity of each layer remains constant. Downsampling is directly performed by combining two layers with two steps [28].

The learning parameters used in the continuous learning process include several important hyperparameter settings. The model is trained for 100 epochs, meaning it goes through the training dataset 100 times.

TABLE 1
ResNet-101 Architecture [29]

Layer name	Output shape	101-layers
*conv*1	112×112	7×7, 64 stride 2
*conv*2_*x*	56×56	3×3 max pool, stride 2 $\begin{bmatrix}1\times1 & 64\\ 3\times3 & 64\\ 1\times1 & 256\end{bmatrix}\times 3$
conv_x	28×28	$\begin{bmatrix}1\times1 & 128\\ 3\times3 & 128\\ 1\times1 & 512\end{bmatrix}\times 4$
conv1_x	14×14	$\begin{bmatrix}1\times1 & 256\\ 3\times3 & 256\\ 1\times1 & 1024\end{bmatrix}\times 23$
conv1_x	7×7	$\begin{bmatrix}1\times1 & 512\\ 3\times3 & 512\\ 1\times1 & 2048\end{bmatrix}\times 3$
	1×1	3×3 average pool, 1000-d fc

A learning rate of 0.001 is used to control the size of weight adjustments in each iteration, ensuring stability during training. The optimizer is Stochastic Gradient Descent (SGD) with a momentum of 0.9, which accelerates convergence and reduces oscillations in the gradients.

C. WEB SYSTEM DEVELOPMENT

The development of the proposed intelligent web-based system is done using React and Next.js, JavaScript frameworks that enable the creation of interactive and dynamic user interfaces (UI) and support server-side rendering for better performance. The website design is implemented with Tailwind CSS, a framework that allows efficient styling of HTML elements, ensuring a modern and responsive appearance. The deployment process is carried out using Vercel, a hosting platform that provides ease and speed in the deployment process, as well as supporting continuous deployment for automatic updates. Additionally, we utilize Vercel Analytics for performance monitoring. The backend technology used is Node.js.

We use FastAPI as the main framework. The selection of FastAPI is based on various reasons, including its ability to handle requests quickly and its strong support for data types, which helps in automatic input validation. FastAPI also allows for fast and efficient development, as well as supporting async programming, which is very useful for improving application performance. This API plays an important role as an intermediary for communication between the website and the cloud, allowing the website to send data to the cloud and receive responses from the deployed ResNet-101 model, enabling real-time predictions.

D. DEPLOYMENT OF RESNET-101 ON CLOUD

In this work, we utilized the DigitalOcean cloud platform to deploy the ResNet-101 model. Initially, we created a droplet that functions as a virtual server on DigitalOcean. SSH (Secure Shell) is applied to access the virtual server. Once connected to the virtual server, we installed the required libraries. Subsequently, the model was uploaded to the server and is now ready to be used on the website.

E. IMPLEMENTATION OF CONTINUOUS LEARNING

This section describes the continuous learning method used to update the previously built model. The process of continuous learning is as follows. FIGURE 3 illustrates the workflow in the system used in this research. The process begins with inputting X-ray images through the web interface. The images are then stored in the server's storage. The system uses a virtual machine to process the stored images. This virtual machine runs deep learning algorithms using the ResNet-101 model. The ResNet-101 model will be updated periodically based on the new images received. The final outcomes of this process include the updated ResNet-101 model and a learning curve graph showing the model's performance during training.

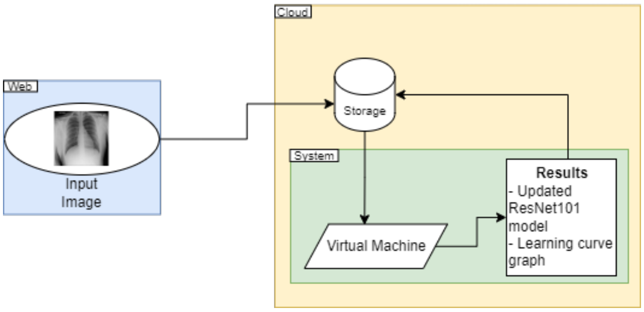


FIGURE 3. **Continuous learning process.**

FIGURE 3 illustrates the workflow of the novel continuous learning of the proposed intelligent system. The process begins with feeding X-ray images through the web interface. The images are then stored in the server's storage. The system uses a virtual machine to process the stored images. This virtual machine runs deep learning algorithms using the ResNet-101 model. The ResNet-101 model will be updated periodically based on the new images received. The final outcomes of this process include the updated ResNet-101 model and a learning curve graph showing the model's performance during training.

Continuous Learning is defined by several key principles. First, online learning allows the learning process to occur at any time without fixed tasks or permanent datasets, and without clear boundaries between different tasks. Second, there is forward and backward transfer, where the model can use learning from previous tasks to help with new tasks, and conversely, new tasks can also improve the model's performance on earlier tasks. Additionally, the model must be resilient to catastrophic forgetting, meaning that learning on new tasks should not reduce the model's performance on previous data. Lastly, even though the model does not have direct access to previous tasks, it must still be able to retain knowledge from those tasks [30].

In a Continuous Learning setting, an infinite sequence of data is considered, where at each time step t, the network receives new data $\{x_t, y_t\}$ drawn from a distribution that is not independent and identically distributed. This distribution P can change, either rapidly or gradually. The main goal is to learn a function governed by parameters θ that can minimize a pre-defined loss $\mathcal{L}$ on the new data, without disrupting the previously learned tasks, and with the possibility of improving performance on tasks that have already been learned [31].

$$\theta^t = \underset{\theta,\xi}{argmin}\ \mathcal{L}\left(F(X_t;\theta), y_t\right) + \sum \xi\, i \tag{1}$$

Such that:

$$\mathcal{L}(F(x_t;\theta), y_t) \leq \mathcal{L}(F(x_t;\theta^{t-1}), y_t) + \xi_i, \tag{2}$$

$$\xi_i \geq 0\ ;\ \forall i\ \in [0..t-1] \tag{3}$$

Where x_t is the input, y_t is the output, and $\xi = \{\xi_i\}$ represents slack variables that allow for some constraints to be violated, such as a slight increase in the loss from previous tasks [31].

IV. RESULT

This research utilizes a laptop with 8 GB RAM and an Intel i3-5005U CPU running at 2.00 GHz with 4 cores (4 CPUs) for website creation, development, and report writing. Key software tools include Visual Studio Code, Python, and JavaScript for web development, model training, and overall website creation. Additionally, DigitalOcean cloud services, specifically a droplet with 8 GB of RAM and 4 vCPUs, are crucial for model development and deployment.

The following subsections comprehensively describe the simulation results. First, we illustrate the performance of the trained model integrated into the proposed intelligent system. Then, we explain the integration process following the functional system testing results. Finally, we demonstrate the continuous learning process and its impact on the performance of the inference model.

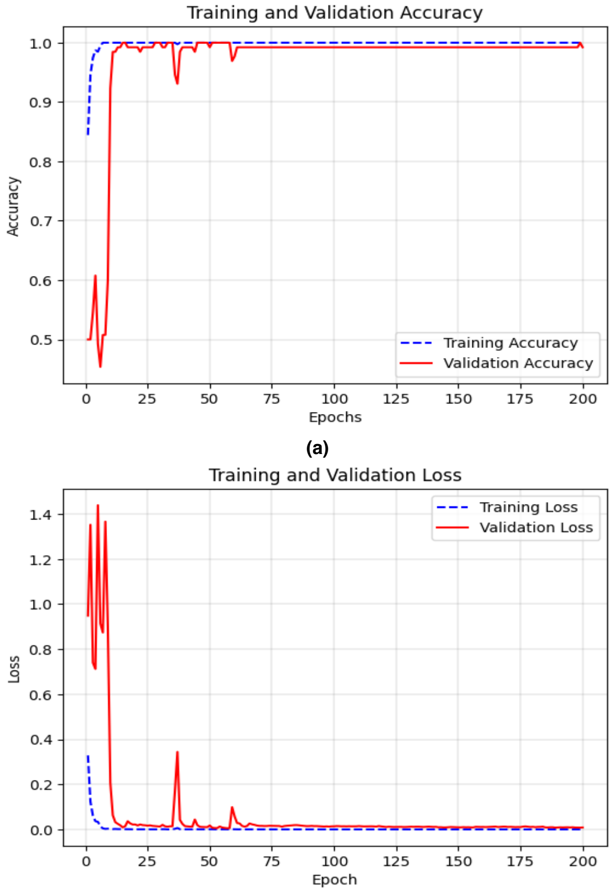
A. CLASSIFIER MODEL PERFORMANCE

The training results of learning curves are shown in FIGURE 4. FIGURE 4 (a) shows the training accuracy graph reaches an almost perfect value close to 1 after approximately 25 epochs and remains stable until the end of training. Meanwhile, the validation accuracy initially fluctuates but then improves to near-perfect values after around 25 epochs and remains stable until the end of training. This indicates that the model learns well from the training data and is able to generalize these patterns to the validation data without significant overfitting.

In FIGURE 4 (b), the training loss graph quickly decreases to near zero after approximately 25 epochs and remains stable until the end of training. Meanwhile, the validation loss initially experiences significant fluctuations and reaches a peak of around 1.4 before decreasing drastically and stabilizing close to zero after about 25 epochs. The rapid decrease and stability achieved by both losses indicate that the model is effective in reducing prediction errors for both training and validation data.

Overall, these two graphs indicate that the trained model performs well. High training and validation accuracy, along with low and stable loss, show that the model not only learns effectively from the training data but also generalizes well to the validation data. The initial fluctuations in both graphs reflect the model's early adjustment phase in finding optimal parameters. Still, after approximately 25 epochs, both accuracy and loss reach stability, indicating consistent and reliable model performance. Moreover, the testing process resulted in a 99.2% accuracy rate with a .h5 model size of 327 MB and a .tflite model size of 162 MB. Thus, this inference model can be implemented into a web-based intelligent system and allow self-learning using CXR images fed by the users to improve the accuracy rate.

This section describes the integration process following the results of functional system testing. In addition, we also show the continuous learning process and its impact on the performance of the inference model.



(a)

(b)

FIGURE 4. Learning curve of training ResNet-101 model (a) accuracy and (b) loss.

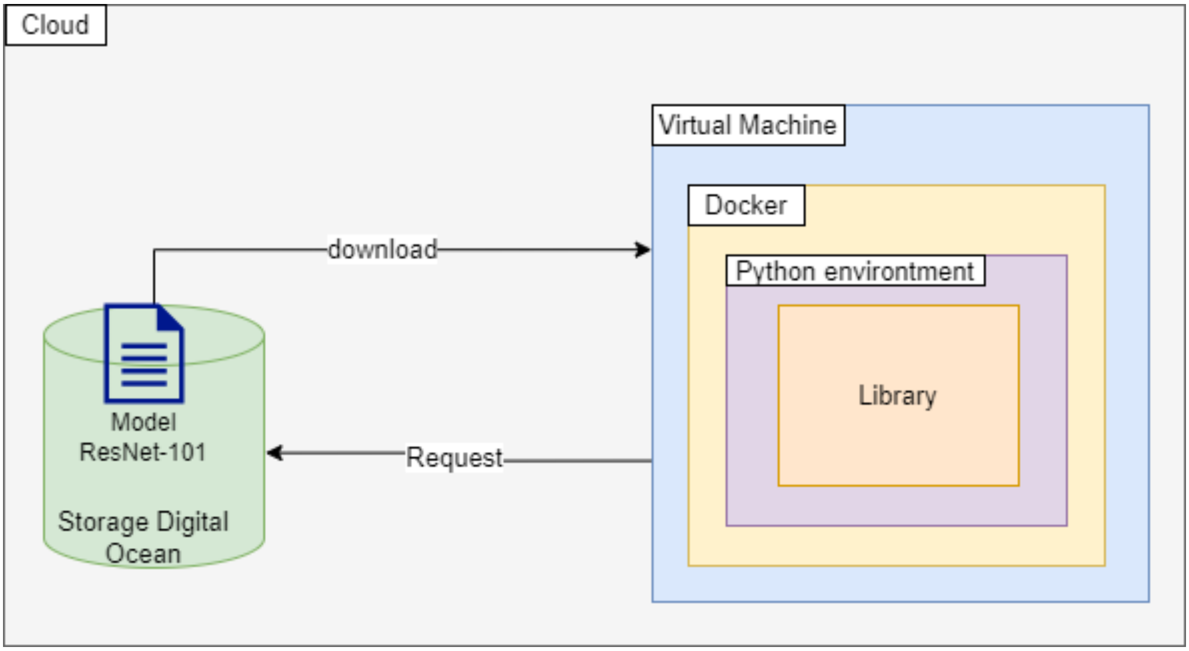


FIGURE 5. Integration of ResNet-101 model into the intelligent system.

B. INTEGRATION OF RESNET-101 MODEL

The inference model of ResNet-101 was successfully deployed into the web and integrated with a continuous learning system. Thus, the model can continue to learn from new incoming data of chest X-ray images, thereby increasing its reliability over time. FIGURE 5 illustrates the integration of the ResNet-101 model into an intelligent continuous learning system, where this system is integrated into the cloud. The process begins with data storage on DigitalOcean, where the latest data and models are kept. The virtual machine sends requests to the storage to download the necessary data for training and updating the model. After the update, the optimized model is re-uploaded to the storage. Thus, this system supports continuous learning, allowing the model to evolve as new data is added. However, a more thorough comparative analysis with existing tuberculosis detection models is required to contextualize the results, and will be addressed in future research.

FIGURE 6 illustrates the integration of the ResNet-101 model into a web-based application connected to the cloud via FastAPI. Data is stored on DigitalOcean and can be accessed by the application through download requests. The application platform is built using Docker to provide a consistent and scalable environment. Inside Docker, FastAPI is used to handle requests and provide fast responses. Users upload images through the web interface, and these images are processed by the ResNet-101 model deployed on cloud. The prediction results, showing the probabilities of TB or normal, are displayed back to the user in the web interface.

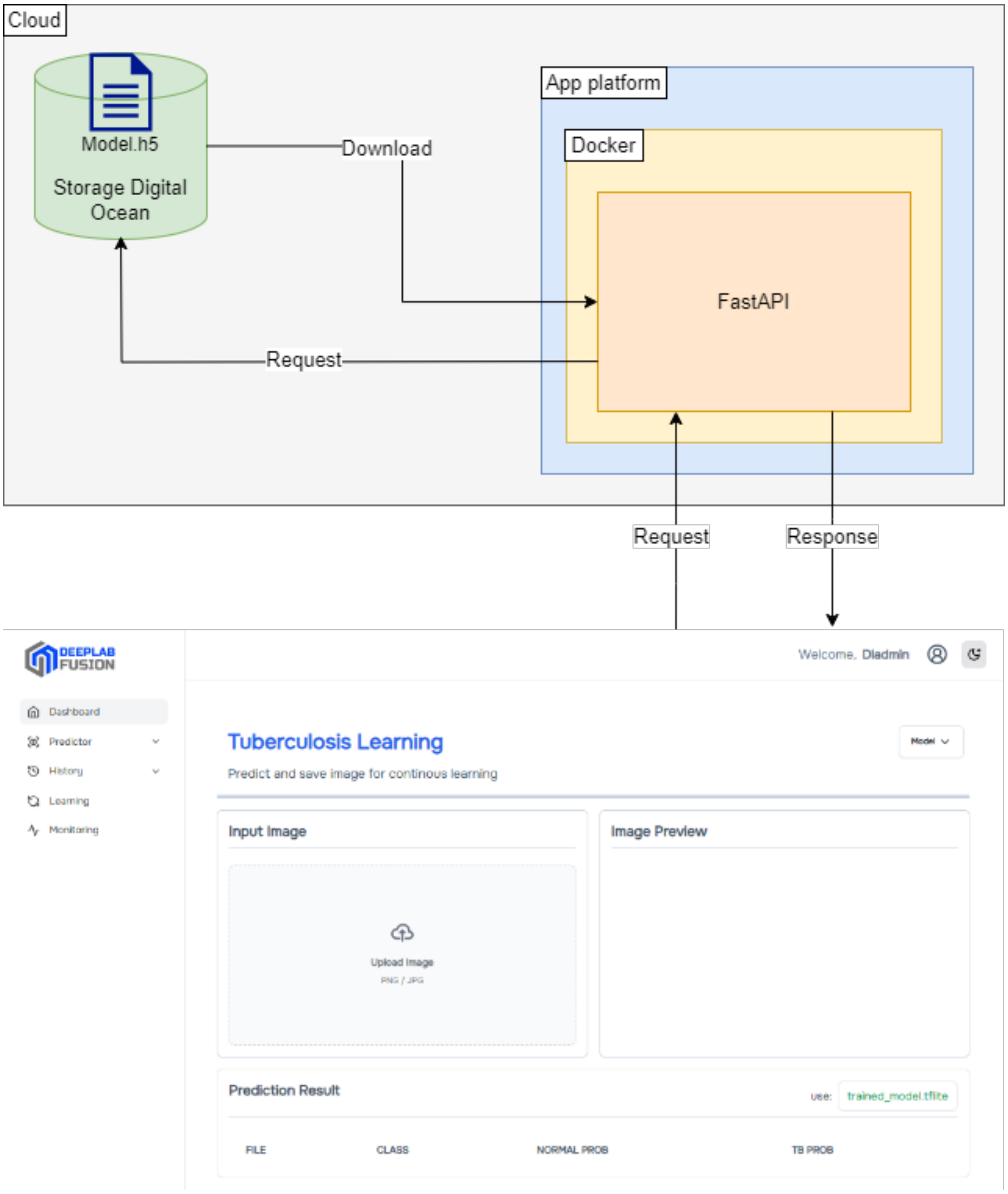
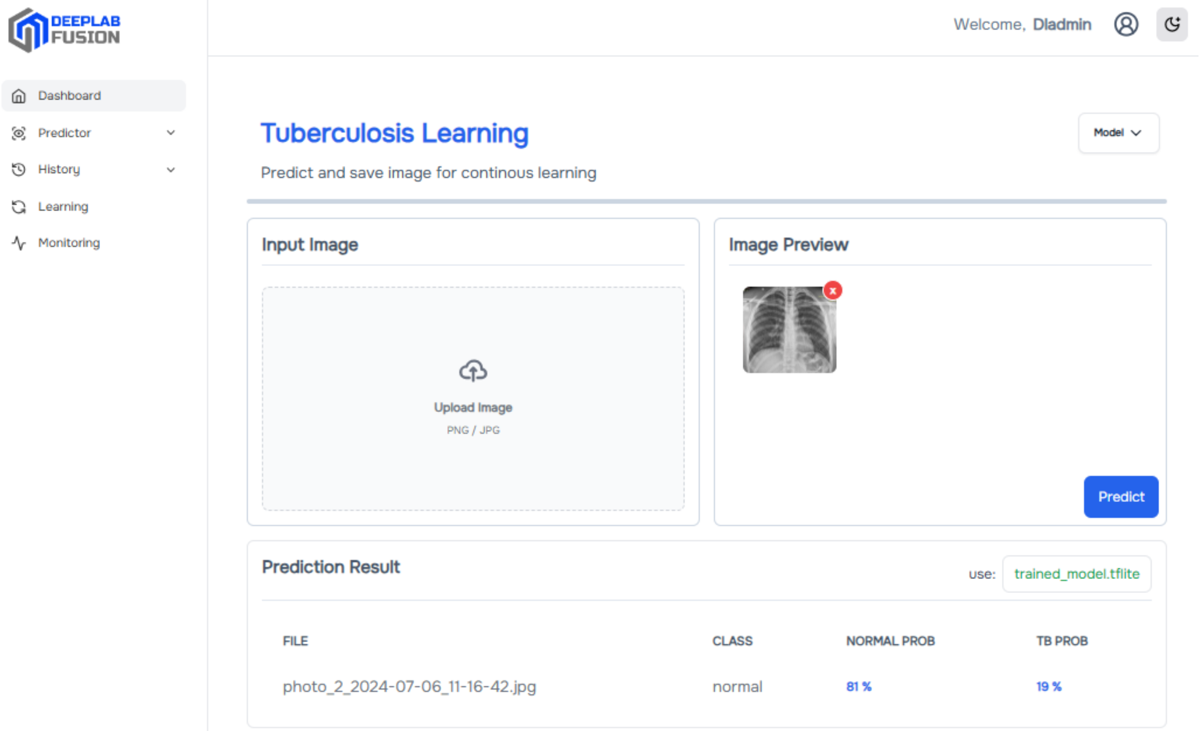


FIGURE 6. **Integration of ResNet-101 model into web-based application.**

C. FUNCTIONAL SYSTEM

FIGURE 7 shows the learning page, which includes the file name, image class, and a visual representation of the prediction with a percentage bar indicating the likelihood that the X-ray image is normal or has TB. Additionally, there is a model version identifier (trained_model.tflite) that indicates the version of the deep learning model used for the prediction. This interface demonstrates that the uploaded image has been successfully analysed, providing users with a simple and efficient way to check for the presence of tuberculosis using deep learning technology.

FIGURE 8 shows page view of the monitoring interface in the web-based intelligent system. At the top, information about the saved dataset is displayed, showing 13 normal images and 22 TB images. The main section of this interface presents the models resulting from retraining. Several models are listed with the model name, file size, last modified date, and model version. Additionally, there is a "learning curve" section displaying the learning curve or training result graph, as well as a "System process" section showing ongoing system processes with a "Loading..." status. This interface provides a clear overview of the performance and updates of the models used in the system for tuberculosis detection, while also monitoring the continuous learning process being applied.



Tuberculosis Learning

Predict and save image for continous learning

Input Image

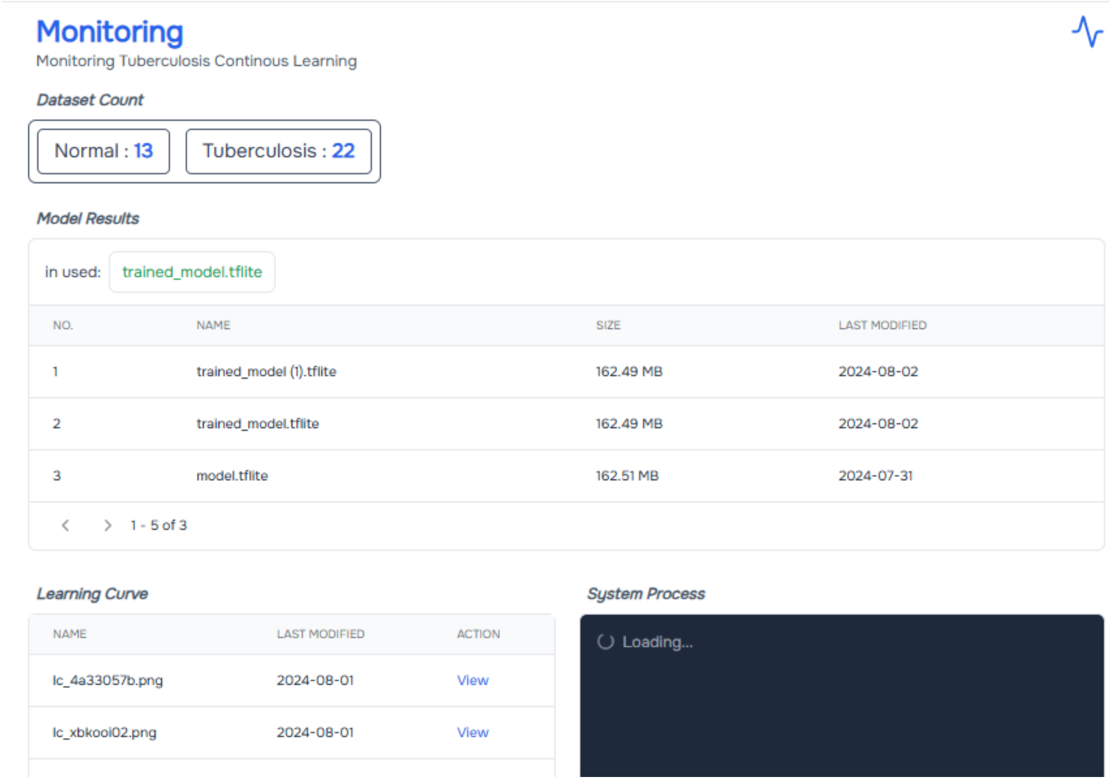
Image Preview

Prediction Result

use: trained_model.tflite

FILE	CLASS	NORMAL PROB	TB PROB
photo_2_2024-07-06_11-16-42.jpg	normal	81 %	19 %

FIGURE 7. Learning page view from web-based application.



Monitoring

Monitoring Tuberculosis Continous Learning

Dataset Count

Normal : 13 Tuberculosis : 22

Model Results

in used: trained_model.tflite

NO.	NAME	SIZE	LAST MODIFIED
1	trained_model (1).tflite	162.49 MB	2024-08-02
2	trained_model.tflite	162.49 MB	2024-08-02
3	model.tflite	162.51 MB	2024-07-31

1 - 5 of 3

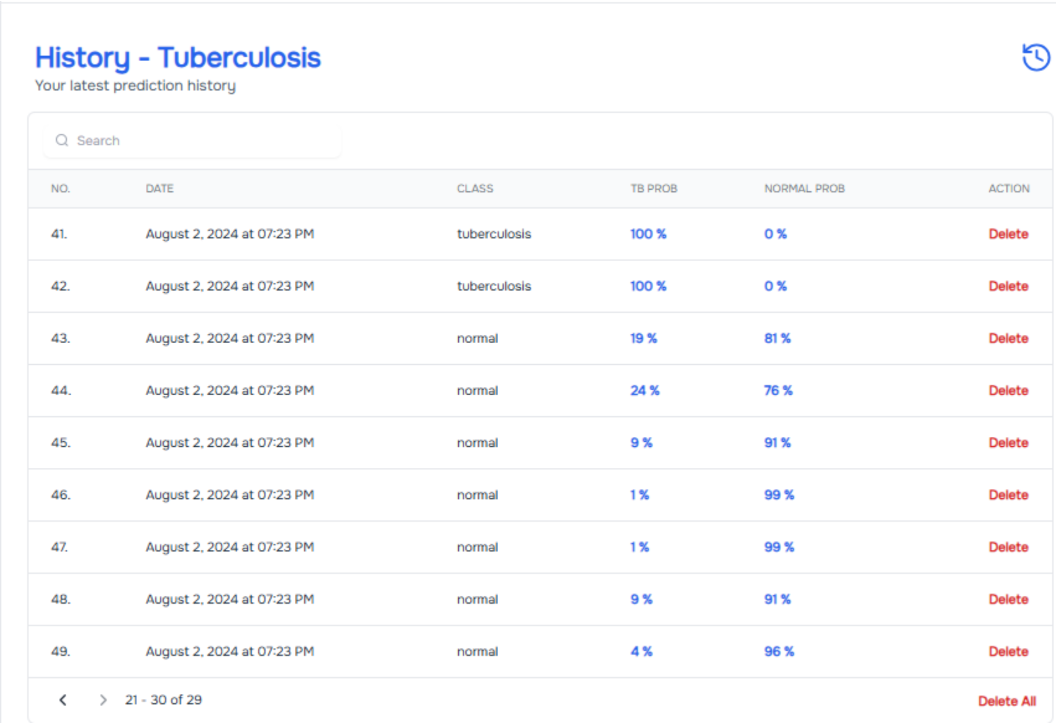
Learning Curve

NAME	LAST MODIFIED	ACTION
lc_4a33057b.png	2024-08-01	View
lc_xbkooi02.png	2024-08-01	View

System Process

Loading...

FIGURE 8. Monitoring page view from web-based application.



History - Tuberculosis
Your latest prediction history

Search

NO.	DATE	CLASS	TB PROB	NORMAL PROB	ACTION
41.	August 2, 2024 at 07:23 PM	tuberculosis	100 %	0 %	Delete
42.	August 2, 2024 at 07:23 PM	tuberculosis	100 %	0 %	Delete
43.	August 2, 2024 at 07:23 PM	normal	19 %	81 %	Delete
44.	August 2, 2024 at 07:23 PM	normal	24 %	76 %	Delete
45.	August 2, 2024 at 07:23 PM	normal	9 %	91 %	Delete
46.	August 2, 2024 at 07:23 PM	normal	1 %	99 %	Delete
47.	August 2, 2024 at 07:23 PM	normal	1 %	99 %	Delete
48.	August 2, 2024 at 07:23 PM	normal	9 %	91 %	Delete
49.	August 2, 2024 at 07:23 PM	normal	4 %	96 %	Delete

21 - 30 of 29 Delete All

FIGURE 9. History page view from web-based application.

FIGURE 9 shows the "History - Tuberculosis" page. The main screen displays a tuberculosis prediction history table, which includes several columns such as number, prediction date and time, predicted class (normal or tuberculosis), model confidence percentage in the prediction, correct class, confidence percentage for the correct class, and an option to delete the entry. The table lists the last 9 predictions with details such as July 3, 2024, at 08:37 AM, where the model predicted "normal" with 91% confidence, while the correct class was "tuberculosis" with 95% confidence. This feature allows users to track and manage the history of tuberculosis predictions made by the model, providing insights into the model's performance and accuracy in predicting health conditions.

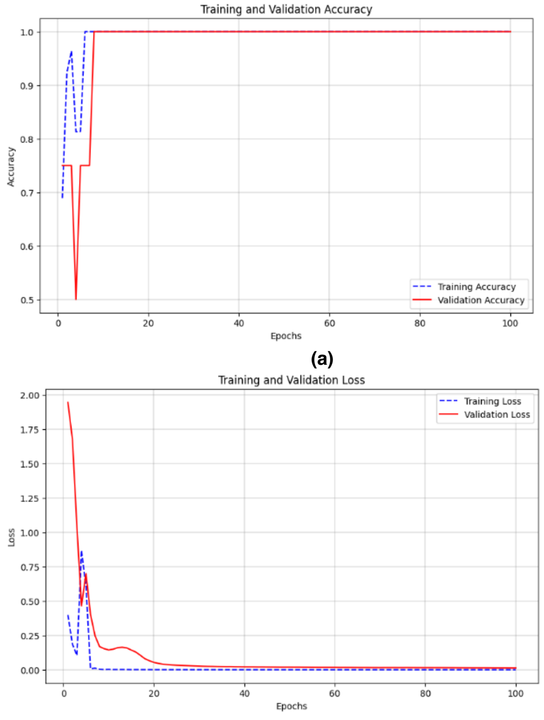
D. THE INTELLIGENT SYSTEM PERFORMANCE

We proceeded with the initial testing of the proposed intelligent system using eight images for each class of normal and tuberculosis to identify the accuracy of the model's prediction. Both classes were tested using real X-ray images from a secondary source. The testing results in TABLE 2 shows that out of 8 normal images tested, only 2 images were successfully detected correctly by the model. Meanwhile, all tuberculosis images were successfully detected correctly, indicating better model performance in detecting TB images compared to normal images. The finding shows that even though the model was 99.2% accurate prior to being deployed into the system when implemented to the real chest x-ray images, the model performance decreased. It may happen because the model was trained on different sources of chest X-ray images, which may feature different characteristics. Thus, continuous learning is needed to maintain the model's performance during practical usage.

The continuous learning system is set to be able to update the model if there are 5 images each in the normal and tuberculosis classes. However, as the testing results show that the system is unable to recognize the normal dataset correctly, as shown in TABLE 2, the predicted results can not be used as it was not correctly labeled the image. Thus, we performed a manual model update using 8 images for each class, namely normal and TB, with a data division of 80:20 for training and validation. The training process was carried out with 100 epochs and a batch size of 1.

TABLE 2
Initial testing results

Dataset	Class	Prediction results		
		Normal	TB	Probability
photo_2	Normal	-	74%	TB
photo_3	Normal	-	99%	TB
photo_4	Normal	-	60%	TB
photo_5	Normal	-	75%	TB
photo_6	Normal	100%	-	Normal
photo_7	Normal	98%	-	Normal
photo_8	Normal	-	89%	TB
photo_9	Normal	-	81%	TB
photo_1	TB	-	92%	TB
photo_2	TB	-	99%	TB
photo_3	TB	-	96%	TB
photo_4	TB	-	95%	TB
photo_5	TB	-	95%	TB
photo_6	TB	-	100%	TB
photo_7	TB	-	92%	TB
photo_8	TB	-	96%	TB



(a)

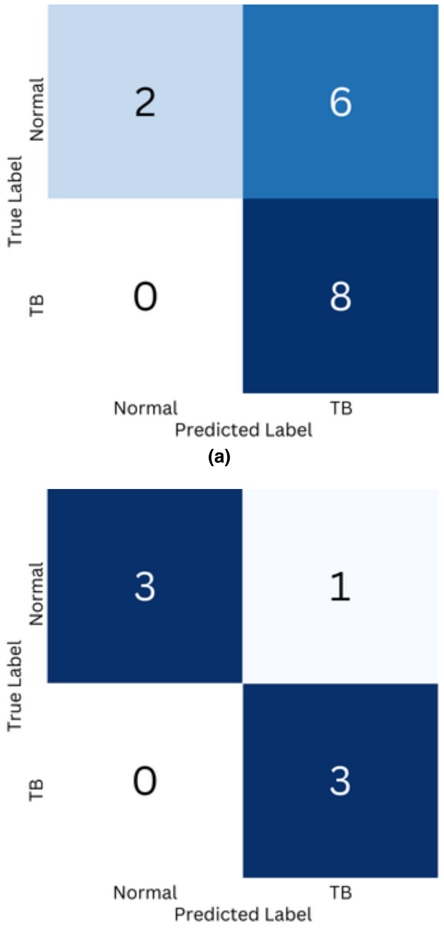
(b)

FIGURE 10. Learning curve of manual training (a) accuracy and (b) loss.

FIGURE 10 shows the learning curve of the model training process. In FIGURE 10 (a), it was observed that the training accuracy experienced initial fluctuations but quickly reached a very high and stable value of 100% after the first few epochs. On the other hand, the validation accuracy showed a steadily increasing trend from the beginning of training, with a slight decrease in some early epochs, before eventually reaching and maintaining a value of 100% until the end of training. It indicated that the model was able to learn well from the training data and also maintain good performance on the validation data. In FIGURE 10 (b), the loss decreases in both training and validation data as the number of epochs increases. The training loss decreases significantly from the beginning of training to a very low value. Although the validation loss fluctuates, overall it still shows a downward trend, indicating that the model does not experience overfitting and is able to learn data patterns well. These results show that even though the dataset used is very small, the model is able to learn well and provide fairly accurate prediction results.

V. DISCUSSION

Based on the results of this study, the test results were obtained in the form of a confusion matrix as seen in FIGURE 11. Confusion matrix in FIGURE 11 (a) shows the results of the tuberculosis (TB) detection model on X-ray images before retraining. In this matrix, the system was tested with 8 TB patient images and 8 normal images. Out of the 8 normal images, the system correctly identified only 2 images as normal, while 6 images were misclassified as TB, resulting in an accuracy rate of 25% for normal images. Conversely, out of the 8 TB images, the system correctly identified all of them as TB, resulting in an accuracy rate of 100% for TB images. Overall, out of 16 tested images, the system correctly classified 10 images (2 normal images and 8 TB images), with an overall accuracy of 62.5%.



(a)

(b)

FIGURE 11. Confusion matrix (a) before retraining and (b) after retraining.

These results indicate that the system is highly effective in recognizing TB images with a perfect accuracy rate, but its performance in recognizing normal images needs significant improvement. The higher accuracy for TB images compared to normal images suggests that this model is better at detecting the presence of TB than identifying normal conditions. However, it is important to acknowledge the limitations of the dataset used in this study, which may contribute to potential biases in the model's performance. This imbalance in the dataset underscores the need for further validation using more comprehensive and balanced data to ensure the robustness and generalizability of the findings.

FIGURE 11 (b) showed the confusion matrix from the model testing after manual continuous learning. The model was tested using 4 normal images and 3 TB images. Out of the 4 normal images, the model correctly identified 3 images as normal but misclassified 1 normal image as TB. On the other hand, from the 3 TB images, the model correctly identified all of them as TB, with no misclassifications. Overall, out of the 7 images tested, the model correctly classified 6 images (3 normal images and 3 TB images), resulting in an overall accuracy of 86%. The sensitivity for TB images was 100%, and the specificity for normal images was 75%. These results indicated that the model performed well in identifying TB images but needed improvement in classifying normal images to enhance specificity and overall accuracy.

This research developed an intelligent system based on a web application for TB detection using continuous learning, which has the potential to assist healthcare professionals, particularly as a tool for early TB detection. This system can be integrated with hospital information systems, allowing medical personnel to upload images and quickly receive prediction results.

VI. CONCLUSION

This research has successfully developed an intelligent system based on a web application for X-ray image classification, specifically utilizing the ResNet-101 model for tuberculosis detection. Although the model initially achieved an accuracy of 99.2%, its performance declined when tested on real X-ray images, indicating discrepancies between the training data and real-world data. Initial tests on 16 images (8 normal and 8 TB) revealed that the model correctly classified all 8 TB images but misclassified 6 normal images, resulting in an overall accuracy of 62.5%. Then, the model was retrained with new data to improve the classifier performance. Subsequent testing after manual continuous learning showed marked improvement, with the model correctly identifying 3 out of 4 normal images and all 3 TB images, achieving an overall accuracy of 86%. However, challenges remain, such as the lack of datasets needed to proceed with the automatic continuous learning process and the need to enhance the model's ability to more accurately recognize images. The sensitivity for TB images remained at 100%, but the specificity for normal images only improved to 75%. The continuous learning system was designed to update the model as more data became available, but initial testing highlighted the need for further refinement to correctly identify normal images.

These findings indicate that while the model performs well in detecting TB images, there is room for improvement in classifying normal images to enhance overall accuracy and specificity. The development of an intelligent system for tuberculosis detection is an important step in utilizing technology to assist healthcare services in the fight against tuberculosis. This work contributes to the broader effort to combat tuberculosis by developing a web-based intelligent system that enables automated TB prediction and training. Such a system could improve early detection and support healthcare professionals in making rapid decisions. As tuberculosis remains a leading cause of death worldwide, particularly in low- and middle-income countries, a reliable and efficient TB detection system is essential to reduce delays in diagnosis and improve patient outcomes. The success of this initiative relies on collaboration between researchers and clinicians. Researchers must refine AI models, while clinicians should provide real-world feedback to improve these systems. Through collaborative action, we can maximize AI's potential in combating tuberculosis. Future work will focus on addressing the dataset imbalance, the region of interest of the TB area, enhance the image quality, and refining the continuous learning system to improve the model's robustness and accuracy, particularly for normal image classification.

REFERENCES

[1] A. Pant, B. Das, and G. A. Arimbasseri, "Host microbiome in tuberculosis: disease, treatment, and immunity perspectives," *Front. Microbiol.*, vol. 14, p. 1236348, Sep. 2023.

[2] B. Frascella *et al.*, "Subclinical Tuberculosis Disease—A Review and Analysis of Prevalence Surveys to Inform Definitions, Burden, Associations, and Screening Methodology," *Clinical Infectious Diseases*, vol. 73, no. 3, pp. e830–e841, Aug. 2021.

[3] "Laporan Tahunan Program TBC 2022," TBC Indonesia. Accessed: Oct. 01, 2024. [Online]. Available: https://tbindonesia.or.id/pustaka-tbc/laporan-program/

[4] Universitas Pertamina, A. Kusumaningrum, G. Wulandari, Universitas Pertamina, A. Kautsar, and Universitas Pertamina, "Tuberkulosis di Indonesia: Apakah Status Sosial-Ekonomi dan Faktor Lingkungan Penting?," *JEPI*, vol. 23, no. 1, pp. 1–14, Jan. 2023, doi: 10.21002/jepi.2023.01.

[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.

[6] K. T. Kadhim, A. M. Alsahlany, S. M. Wadi, and H. T. Kadhum, "An Overview of Patient's Health Status Monitoring System Based on Internet of Things (IoT)," *Wireless Pers Commun*, vol. 114, no. 3, pp. 2235–2262, Oct. 2020.

[7] L. N. Sanchez-Pinto, Y. Luo, and M. M. Churpek, "Big Data and Data Science in Critical Care," *Chest*, vol. 154, no. 5, pp. 1239–1248, Nov. 2018.

[8] R. Kejriwal and Mohana, "Artificial Intelligence (AI) in Medicine and Modern Healthcare Systems," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Trichy, India: IEEE, Nov. 2022, pp. 25–31. doi: 10.1109/ICAISS55157.2022.10010939.

[9] Y. Guo, Z. Hao, S. Zhao, J. Gong, and F. Yang, "Artificial Intelligence in Health Care: Bibliometric Analysis," *J Med Internet Res*, vol. 22, no. 7, p. e18228, Jul. 2020.

[10] P. Lakhani and B. Sundaram, "Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks," *Radiology*, vol. 284, no. 2, pp. 574–582, Aug. 2017.

[11] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization," *Sci Rep*, vol. 9, no. 1, p. 6268, Apr. 2019, doi: 10.1038/s41598-019-42557-4.

[12] U. K. Lopes and J. F. Valiati, "Pre-trained convolutional neural networks as feature extractors for tuberculosis detection," *Computers in Biology and Medicine*, vol. 89, pp. 135–143, Oct. 2017, doi: 10.1016/j.compbiomed.2017.08.001.

[13] S. Aulia and S. Hadiyoso, "Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 2, p. 290, Jul. 2022.

[14] 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, p. 012020, Mar. 2020.

[15] M. K. Puttagunta and S. Ravi, “Detection of Tuberculosis based on Deep Learning based methods,” *J. Phys.: Conf. Ser.*, vol. 1767, no. 1, p. 012004, Feb. 2021.

[16] Y. Liu, Y.-H. Wu, Y. Ban, H. Wang, and M.-M. Cheng, “Rethinking Computer-Aided Tuberculosis Diagnosis”.

[17] “JSRT Database | Japanese Society of Radiological Technology.” Accessed: Aug. 07, 2024. [Online]. Available: http://db.jsrt.or.jp/eng.php

[18] S. I. Nafisah and G. Muhammad, “Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence,” *Neural Comput & Applic*, vol. 36, no. 1, pp. 111–131, Jan. 2024.

[19] Department of Information Systems, Faculty of Computer Science, Kabul University, Kabul, Afghanistan., M. N. Akbari, and A. Azizi, “Building a Convolutional Neural Network Model for Tuberculosis Detection Using Chest X-Ray Images,” *ghalib*, vol. 1, no. 1, pp. 21–26, Jan. 2023.

[20] R. Guo, K. Passi, and C. K. Jain, “Tuberculosis Diagnostics and Localization in Chest X-Rays via Deep Learning Models,” *Front. Artif. Intell.*, vol. 3, p. 583427, Oct. 2020, doi: 10.3389/frai.2020.583427.

[21] Y. Lee, M. C. Raviglione, and A. Flahault, “Use of Digital Technology to Enhance Tuberculosis Control: Scoping Review,” *J Med Internet Res*, vol. 22, no. 2, p. e15727, Feb. 2020.

[22] C. Prasitpuriprecha *et al.*, “Drug-Resistant Tuberculosis Treatment Recommendation, and Multi-Class Tuberculosis Detection and Classification Using Ensemble Deep Learning-Based System,” *Pharmaceuticals*, vol. 16, no. 1, p. 13, Dec. 2022.

[23] S. Bello *et al.*, “Empirical evidence of delays in diagnosis and treatment of pulmonary tuberculosis: systematic review and meta-regression analysis,” *BMC Public Health*, vol. 19, no. 1, p. 820, Dec. 2019.

[24] “Tuberculosis Chest X-rays (Montgomery).” Accessed: Jan. 16, 2024. [Online]. Available: https://www.kaggle.com/datasets/raddar/tuberculosis-chest-xrays-montgomery

[25] W.-J. Koh *et al.*, “Chest Radiographic Findings in Primary Pulmonary Tuberculosis: Observations from High School Outbreaks,” *Korean J Radiol*, vol. 11, no. 6, p. 612, 2010.

[26] “Drug resistant tuberculosis X-rays.” [Online]. Available: https://www.kaggle.com/datasets/raddar/drug-resistant-tuberculosis-xrays

[27] “RSNA Pneumonia Detection Challenge.” Accessed: Jul. 15, 2024. [Online]. Available: https://kaggle.com/competitions/rsna-pneumonia-detection-challenge

[28] S. P. Dash, J. Ramadevi, R. Amat, P. K. Sethy, S. K. Behera, and S. Mallick, “Wafer Defect Identification with Optimal Hyper-Parameter Tuning of Support Vector Machine using the Deep Feature of ResNet 101,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1291, no. 1, p. 012048, Sep. 2023.

[29] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, “Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer,” *Procedia Computer Science*, vol. 179, pp. 423–431, 2021.

[30] B. Salami, K. Haataja, and P. Toivanen, “State-of-the-Art Techniques in Artificial Intelligence for Continual Learning: A Review,” presented at the 16th Conference on Computer Science and Intelligence Systems, Sep. 2021, pp. 23–32. doi: 10.15439/2021F12.

[31] R. Aljundi, “Continual Learning in Neural Networks,” Oct. 18, 2019, *arXiv*: arXiv:1910.02718. Accessed: Aug. 26, 2024. [Online]. Available: http://arxiv.org/abs/1910.02718

Qurrata A’yuni was born in Banda Aceh, September 14, 2001. She received a Bachelor's degree in Computer Engineering from the Department of Computer Engineering, Syiah Kuala University, in 2023. Her undergraduate research focused on multimedia technology, specifically deep learning, with research interests in the same field. She is currently pursuing a Master's degree in Electrical Engineering at Syiah Kuala University, with research interests in deep learning and a thesis focus on information technology.

Nasaruddin Nasaruddin was born in Dayah Leubue, Pidie Jaya, on April 2, 1974. He received his B.Eng. degree in Electrical Engineering from the Sepuluh Nopember Institute of Technology, Surabaya, Indonesia in 1997. Subsequently, he received his M. Eng and D. Eng in Physical Electronics and Informatics, the Graduate School of Engineering, Osaka City University, Japan, in 2006 and 2009, respectively. He is a full professor at Electrical Engineering Department, Syiah Kuala University.

Muhammad Irhamsyah was born in Banda Aceh, July 18, 1972. He is currently a lecturer in the Department of Electrical Engineering, Faculty of Engineering, Syiah Kuala University. Joined Syiah Kuala University since 2001 in Telecommunication Engineering. Graduated from S1 Department of Electrical Engineering ITS Surabaya in 1998 and S2 from the Department of Electrical Engineering University of Indonesia in 2008 with the field of Telekomunikas Engineering expertise. Has a research interest in wireless telecommunication and antenna.

Mulkan Azhary is a teaching staff working at Department of Anatomy and Histology, Fakultas Kedokteran Universitas Syiah Kuala (USK) in Banda Aceh - Indonesia. He did the undergraduate and professional medical training at Fakultas Kedokteran Universitas Syiah Kuala (USK) in 2008. Then, in 2011 he was graduated from the post-graduate study (M.Sc.) in Clinical Anatomy at School of Medical Sciences – Universiti Sains Malaysia (USM) in Kota Bharu - Malaysia. In 2015, he graduated the Ph.D. study in Medical Sciences at Fakultas Kedokteran Universitas Airlangga (Unair) in Surabaya - Indonesia. He also had the clinical qualification of Pulmonologist that was graduated from Fakultas Kedokteran - Universitas Indonesia (FKUI) in 2020.

Roslidar Roslidar received her Bachelor Degree in Electrical Engineering in 2001 from Universitas Syiah Kuala. In 2009 she graduated from the Master program in Telecommunication Engineering, University of Arkansas, USA, under Fulbright scholarship. In January 2022, she received her PhD in Doctoral School of Engineering Science, Universitas Syiah Kuala.

Since 2001 she has been the lecturer and researcher at the Department of Electrical and Computer Engineering in Universitas Syiah Kuala. Her research interest is developing the e-health monitoring system based on thermal imaging. She is also active in any research related with electrical engineering and deep learning.