EKSAKTA
Journal of Sciences and Data Analysis

E-ISSN: 2720-9326
P-ISSN: 2716-0459

DOI : 10.20885/EKSAKTA.vol6.iss2.art5

Deep Learning for Lung Disease Diagnosis: A CNN-Based Radiographic Approach

Danang Bagus Wibowo[1] , Achmad Fauzan[2,*]

[1,2] Statistics Study Program, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia, Indonesia.

[*] *Corresponding author: achmadfauzan@uii.ac.id*

Abstract: Various types of lung diseases affect the human respiratory system, with pneumonia, tuberculosis, and Covid-19 being among the most common. Early detection plays a crucial role in improving treatment outcomes and reducing mortality rates. Chest X-ray imaging is one of the most widely used diagnostic methods; however, it typically relies on manual interpretation by medical professionals, which can be time-consuming and prone to inconsistencies. This study aims to apply the Convolutional Neural Network (CNN) method as an automated approach to classify chest X-ray images of lung conditions. The dataset consists of 460 X-ray images for each category: normal, pneumonia, tuberculosis, and Covid-19. The CNN model was trained using an input shape of 224×224 pixels, a 3×3 filter size, and 5 epochs. Evaluation results showed that the model achieved 97% accuracy on the validation and 93% on the testing data. These findings highlight the potential of CNN in supporting automated diagnosis of lung diseases. In the future, this technology is expected to assist healthcare professionals in delivering faster and more accurate diagnoses, particularly in areas with limited access to radiology experts. Moreover, this innovation aligns with Sustainable Development Goal (SDGs) 3: Good Health and Well-being, by promoting early detection, timely treatment, and more equitable access to quality healthcare services.

Keywords: X-ray Image, Convolutional Neural Network (CNN), Sustainable Development Goal (SDGs)

Introduction

Lung disease represents a critical public health concern that, if not diagnosed and treated promptly, may result in severe complications or even mortality. These respiratory conditions can lead to shortness of breath, reduced physical functioning, oxygen deficiency, and in advanced stages, death [1]. Among the most prevalent lung diseases are Pneumonia, Tuberculosis (TB), and Coronavirus Disease 2019 (COVID-19), which was declared a global pandemic by the World Health Organization (WHO) in March 2020. Pneumonia is an inflammation of the lung parenchyma, commonly caused by various infectious agents such as bacteria, viruses, fungi, and parasites, and in some cases by chemical or physical exposures like radiation or extreme temperatures [2]. Covid-19 is a viral respiratory illness caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), which attacks the lungs and can lead to acute respiratory distress [3].

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

Tuberculosis, caused by *Mycobacterium tuberculosis*, primarily affects the lungs but may also spread to other organs. It is transmitted through airborne droplets released by infected individuals [4].

Early detection plays a pivotal role in mitigating the progression and transmission of lung diseases. One of the most commonly used diagnostic tools is chest X-ray imaging, which employs electromagnetic radiation to visualize internal thoracic structures and detect signs of abnormality in the lungs [5]. However, conventional interpretation of chest X-ray results typically relies on manual visual assessment by medical professionals, a process that is time-consuming and subject to inter-observer variability [6]. Advancements in computational intelligence, particularly in deep learning, offer promising alternatives to automate and enhance diagnostic processes. Among these, the Convolutional Neural Network (CNN) has shown exceptional performance in image classification tasks by learning hierarchical representations from raw pixel data [7]. CNNs are particularly advantageous for analyzing complex and high-resolution medical images, such as chest X-rays, due to their ability to detect subtle and abstract visual patterns [8], [9].

Given this background, the present study aims to develop a CNN-based model to classify lung conditions using chest X-ray images into four categories: normal, pneumonia, tuberculosis, and COVID-19. The goal is to support healthcare professionals in accelerating and improving diagnostic accuracy through automated image analysis. Importantly, this research aligns with the United Nations Sustainable Development Goals (SDGs), particularly Goal 3: Good Health and Well-being, by contributing to improved early detection of disease, facilitating faster clinical decision-making, and expanding access to affordable, technology-driven healthcare services. Through the application of artificial intelligence in medical imaging, this study seeks to bridge the gap between innovation and healthcare equity in the effort to reduce preventable deaths from respiratory illnesses.

Despite its widespread use, manual interpretation of chest X-rays poses significant challenges, including inter-observer variability, time consumption, and the risk of misdiagnosis, particularly when pneumonia, TB, and COVID-19 share overlapping visual patterns. Previous studies have attempted to overcome these limitations through computational approaches. Conventional methods such as Gray-Level Co-occurrence Matrix (GLCM) combined with classifiers like Support Vector Machines (SVM) have achieved moderate diagnostic performance, yet they rely heavily on handcrafted feature extraction, which limits their generalizability across diverse datasets. In contrast, more recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated superior capability by automatically learning hierarchical features from raw image data, thereby capturing subtle and abstract visual characteristics. These advantages make CNN a more convincing choice for developing robust and scalable diagnostic systems in medical imaging.

E-ISSN: 2720-9326
P-ISSN: 2716-0459

DOI :

RESEARCH METHOD

Data

The dataset used in this study consists of chest X-ray images of the human lungs, comprising four categories: (1) normal lung images, (2) images of lungs affected by pneumonia, (3) images of lungs with tuberculosis, and (4) images of lungs infected with COVID-19. A total of 1,840 images were included in the analysis. The data are secondary in nature and were obtained from the publicly accessible repository Kaggle.com.

Research Method

The stages of the research process are presented in Figure 1. Based on Figure 1,the first stage of this study involves data preprocessing. In this phase, the dataset is initially split using the *hold-out* method, allocating 90% for training data and 10% for testing. Subsequently, the training data is further divided into training and validation sets with the same proportion (90:10) [10]. This stratified partitioning aims to systematically assess model performance throughout the training and evaluation phases while minimizing the risk of overfitting. To enhance model generalization, data augmentation techniques such as random rotation, horizontal and vertical flipping, zooming, and shifting were applied during preprocessing. These augmentations were chosen to simulate real-world variability in chest X-ray images and improve robustness.

The second stage focuses on designing and training the Convolutional Neural Network (CNN) architecture. Convolutional Neural Network (CNN) is a special type of neural network used to process data with a mesh or grid-like topology. The name "convolutional neural network" indicates that this network uses a mathematical operation called convolution. Convolution is the multiplication operation between the input image values and the filter values. Thus, a convolutional network is a type of neural network that uses convolution in at least one of its layers [11]. CNN starts the process by filtering each input image through the convolution layer, then proceeds to the pooling operation, and finally enters the fully connected layer. Afterward, the softmax function is used to perform object classification by generating a probability value between zero and one [12]. The parts of CNN include the following: (a) convolution layer, (b) stride, (c) pooling layer, (d) fully connected layer, (e) activation function, (f) flatten layers.

In comparison with prior studies, CNN has consistently demonstrated superior performance in medical image classification tasks, often outperforming traditional machine learning techniques such as Support Vector Machines (SVM) and Gray-Level Co-Occurrence Matrix (GLCM) feature extraction [7], [9]. However, its performance is highly dependent on the quality and size of the training dataset, as well as the computational resources available, which remain a challenge in large-scale implementations. The training process in this study was conducted with X epochs, a batch size of Y, and a learning rate of Z. These parameters were selected after a series of tuning experiments and literature benchmarking to ensure convergence without overfitting. The

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

batch size was adjusted to balance computational efficiency with gradient stability, while the learning rate was optimized for reliable convergence. Convolution will first define the input values against the pixels in the form of numbers contained in the image. After that, it calculates the pixel values in the image using a filter or kernel. The output result is obtained through a matrix multiplication operation between the filter and the pixels being processed.

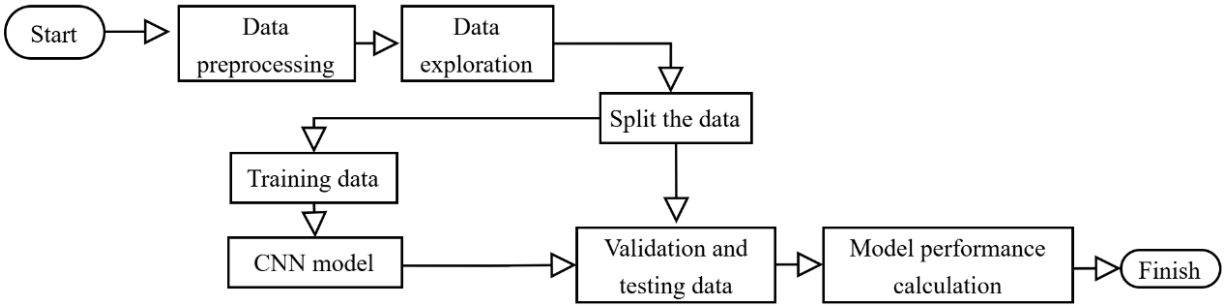


Figure 1. Research Flowchart

In this step, the filter is shifted to the next position in the image according to the set stride length. The calculation process continues repetitively until the entire image passes through the filtering process and produces a feature map [13]. Convolution has the following equation 1 [14].

$$Output = \frac{W - F + 2P}{S} + 1 \quad (1)$$

W = Image Size, F = Filter Size, P = Padding Value Used, and S = Shift Size (*Stride*). Stride is a parameter that sets the amount of filter shift. If the stride value is one, the filter will shift one pixel horizontally and vertically. Using a smaller stride will provide more detailed information from the input, but requires more intensive computation compared to a larger stride [15]. Pooling layer reduces the output size of the pooled feature map, and also reduces the number of parameters used. This helps reduce the risk of overfitting the model to the training data. In general, the pooling layer works by calculating the average value of the feature map called average pooling, or by taking the highest value called max pooling [16]. Fully connected layer is the last layer in a CNN. Each neuron in the previous layer has a direct connection (connected) with each neuron in the fully connected layer. In a CNN architecture, there are usually one or more fully connected layers. This layer uses a non-linear activation function, such as softmax, to generate class prediction probabilities [17].

This function forms the non-linear nature of neural networks, as without an activation function, neural networks are only capable of performing linear functions. The use of non-linear functions is important for solving complex problems as they are able to generate a variety of different patterns simultaneously, and are therefore highly desirable in more complex cases [18]. Some commonly used activation functions include: Rectified linear unit (ReLU) and Softmax.ReLU activation will declare the value of 0 for negative inputs and declare the input value if the input is positive [19]. Softmax is an activation

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

function used to classify linearly by taking into account the probability value of a class. The output value of each class has a range of 0 to 1 and the sum of the outputs of all classes is equal to 1 [20]. Flatten is a process that converts a 2-dimensional matrix or array resulting from convolution and pooling layers into a one-dimensional vector. By flattening, the matrix is converted into a long vector. The resulting vector from the flatten process will then be used as input at the fully connected layer stage [21].

The training process is conducted iteratively until an optimal level of accuracy is achieved for both training and validation datasets. Once the model demonstrates satisfactory performance, it is evaluated using unseen test data to assess its generalization capabilities. To further validate the model's predictive reliability, additional chest X-ray images—excluded from the training process—are used to examine the model's ability to correctly classify labels based on learned patterns. After performing predictions on all observations within a dataset with known class labels, a cross-tabulation between the actual and predicted classes—referred to as a confusion matrix—can be constructed. An illustration of the confusion matrix is presented in Table 1. From the confusion matrix, we calculate evaluation metrics including accuracy, precision, recall (sensitivity), and F1 score, as shown in Equations (2) and (3) [22].

Tabel 1. Confusion Matrix

		Predicted Value	
		Positive	Negative
Actual value	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

$$\text{Accuracy} = \frac{\text{TP+TF}}{\text{TP+FP+TN+FN}}, \ \text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}}, \ F1 - Score = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision+Sensitivity}} \tag{3}$$

The final stage of this research encompasses result interpretation, conclusion formulation, and recommendations. This study is expected to contribute to the development of intelligent diagnostic systems based on radiographic imagery. Moreover, the outcomes align with the Sustainable Development Goals (SDG) Goal 3: Good Health and Well-being, particularly in enhancing early detection, enabling faster and more accurate medical decision-making, and expanding access to technology-driven healthcare services.

RESULTS AND ANALYSIS

Preprocessing CNN

After the dataset input process, the preprocessing stage is then carried out. Image data is divided into two parts, namely training data and testing data, which are useful for building models and testing the quality of the models that have been formed. In addition

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

to training and testing data, there is also validation data used to monitor the performance of the model during training. Data division is done with a scenario of 90% training data and 10% testing data. From the training data, a further split was performed into training and validation subsets with a proportion of 90% and 10%, respectively. The illustration of the number of training, validation, and testing data is presented in Figure 2.

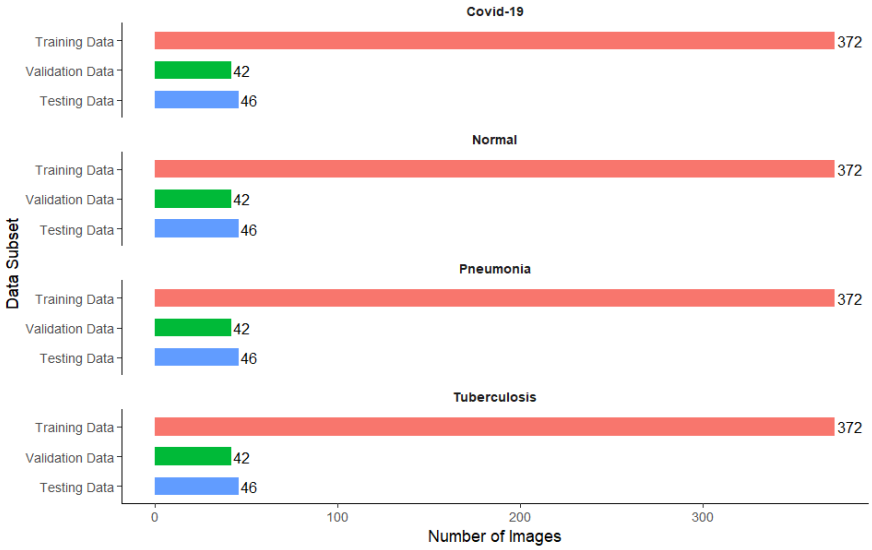


Figure 2. Distribution of Image Subsets by Lung Disease Category

Based on Figure 2, a CNN model was constructed using the training data. The first step involved building a data generator, which aims to produce a continuous data flow to ensure efficient training and testing processes. Subsequently, the image data were resized to a target size of 224×224 pixels. The purpose of this resizing process is to adjust the image dimensions by reducing both horizontal and vertical sizes. The model consists of several types of layers, including convolutional layers (Conv2D), pooling layers (MaxPooling2D), dropout layers (Dropout), a flatten layer (Flatten), and dense layers (Dense). The convolution operation is applied three times using Conv2D layers, each with a 3×3 kernel. The ReLU activation function is used to accelerate the training process and introduce non-linearity into the model. After each convolutional layer, a pooling layer with a 2×2 window is applied to reduce the spatial dimensions gradually. This step helps avoid drastic reductions in input size, thereby preserving essential features from the input image that are critical for accurate classification.

A dropout layer is included to reduce overfitting by randomly deactivating a fraction of the neurons during training. Then, the flatten layer transforms the multidimensional output into a one-dimensional vector. This is followed by a dense layer with 128 units and the ReLU activation function. Finally, an output dense layer with 4 units—corresponding to the number of target classes—is added, using the softmax activation function to perform multiclass classification. An overview of the CNN architecture is presented in Table 2, with a visual illustration provided in Figure 3.

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

Table 2. Summary of CNN Model

Layer	Output Shape	Parameter
Conv2D	(None, 222, 222, 16)	160
MaxPooling2D	(None, 111, 111, 16)	0
Conv2D	(None, 109, 109, 32)	4640
MaxPooling2D	(None, 54, 54, 32)	0
Conv2D	(None, 52, 52, 64)	18496
MaxPooling2D	(None, 26, 26, 64)	0
Flatten	(None, 43264)	0
Dense(128)	(None, 128)	5537920
Dense(4)	(None, 4)	516

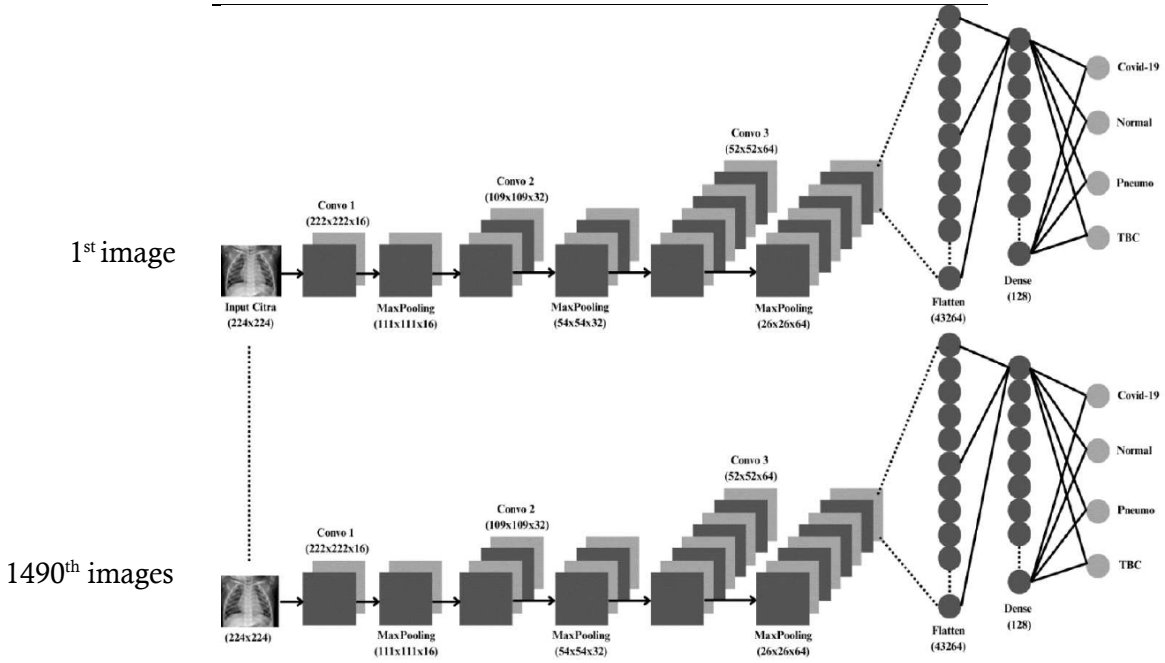
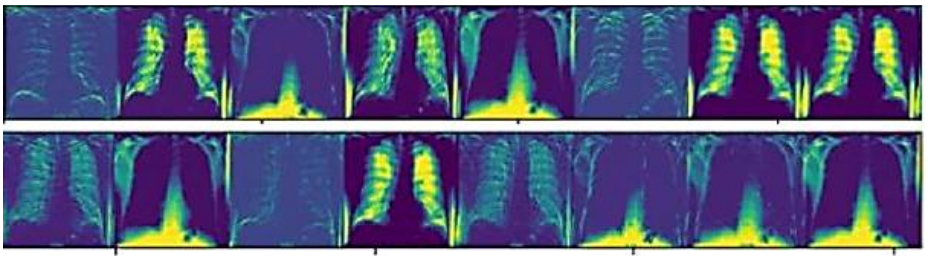


Figure 3. CNN model illustration

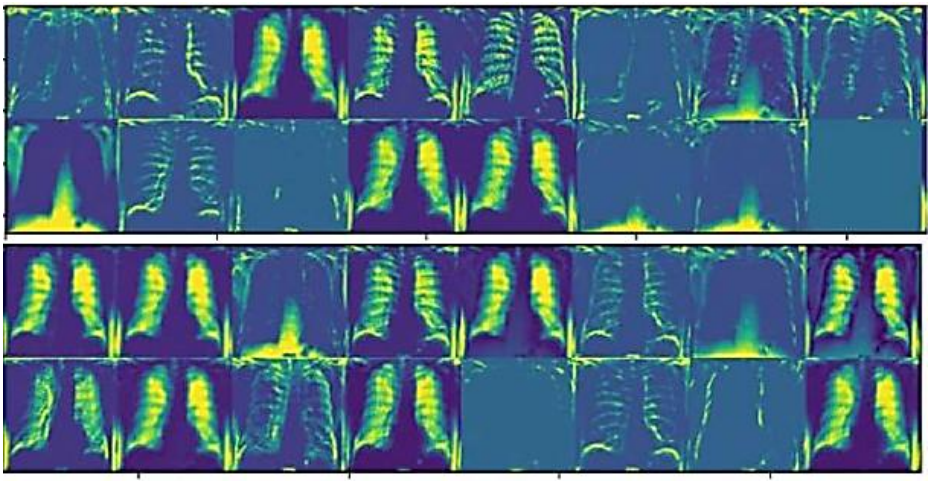
Based on Figure 3, the first step is the convolution layer process. In this step, the pixel values of the input image are defined numerically. The pixel values are then computed using a filter or kernel. The output is generated through matrix multiplication between the filter and the corresponding pixel region. The filter is shifted across the image according to the defined stride length, and this calculation continues until the entire image has been processed, resulting in a feature map [13]. Since the input image has a resolution of 224×224 pixels, only selected pixels are used as samples during the convolution process. Each pixel value represents image color intensity, ranging from 0 (black) to 255 (white). An illustration of the first, second, and third convolution layer processes is shown in Figure 4.

E-ISSN: 2720-9326
P-ISSN: 2716-0459

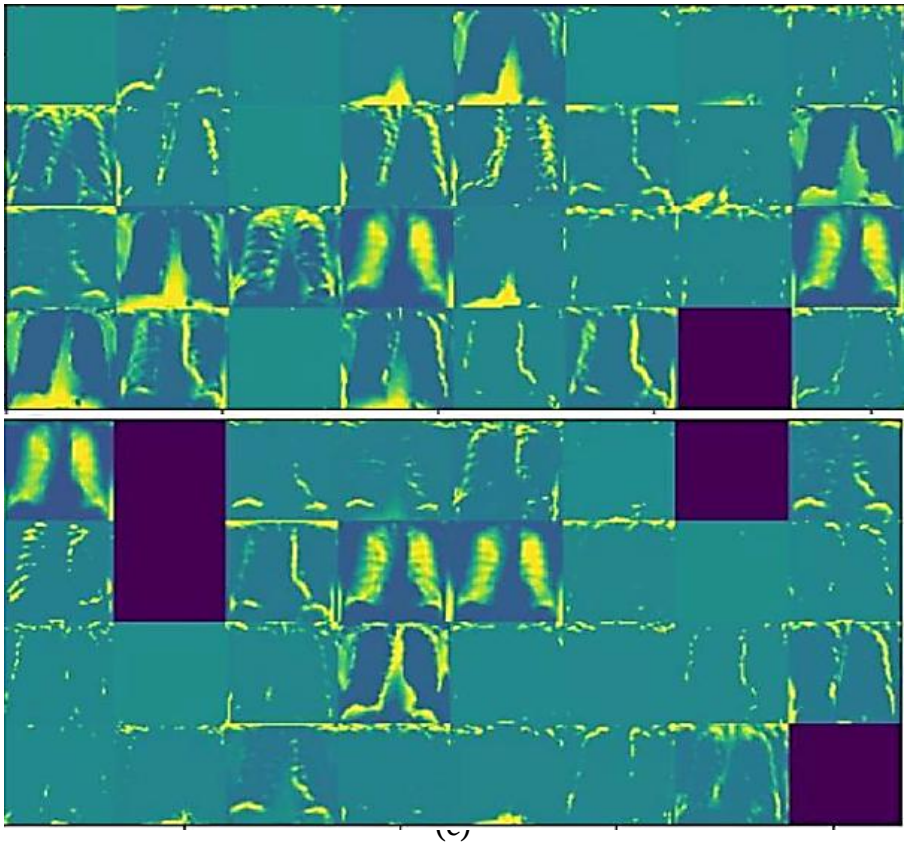
DOI :



(a)



(b)



(c)

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

Figure 4. Illustration of the convolution layer outputs: (a) first layer, (b) second layer, and (c) third layer.

After the convolution layer process, the next step is pooling. The pooling layer serves to reduce the dimensionality of the input matrix by performing spatial reduction. There are two common types of operations in the pooling layer: max pooling and average pooling. The output matrix in this layer is calculated based on the defined filter size and stride [14]. In max pooling, the maximum value within the region defined by the filter and stride is selected. In contrast, average pooling calculates the average value within that region. Following the pooling layer, the process continues with the fully connected layer. In this stage, a flattening process takes place, where the output from the previous layer is converted into a one-dimensional vector. An illustration of the fully connected layer stage is presented in Figure 5.

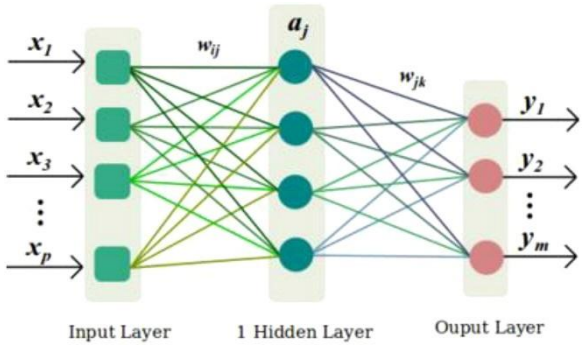


Figure 5. Illustration of the fully connected layer [23]

Based on Figure 5, the value of each neuron (x) is multiplied by a weight (w), which determines the value passed to the next neuron. The calculation for each hidden layer is illustrated by the following equation 4.

$$\sum_{i=1}^{p} x_i * w_{ij} = a_j \tag{4}$$

$x_1, x_2, x_3, \cdots, x_p$ represent the output values from the flattening process, which are then fed into the fully connected layer. Below is an example of how to calculate the value of a_j, where $j = 4$ (in line with the context of this study, which classifies into 4 categories). Suppose the values are: $x_1 = 10, x_2 = 0, x_3 = 8$, and $x_4 = 6$. Then, the values of a_1 through a_4 are calculated as shown in Equation (5).

$$\begin{aligned} a_1 &= (10 \times 0.1) + (0 \times 0.2) + (8 \times 0.3) + (6 \times 0.4) = 5.8 \\ a_2 &= (10 \times 0.3) + (0 \times 0.3) + (8 \times 0.4) + (6 \times 0.4) = 8.6 \\ a_3 &= (10 \times 0.2) + (0 \times 0.5) + (8 \times 0.4) + (6 \times 0.2) = 6.4 \\ a_4 &= (10 \times 0.3) + (0 \times 0.5) + (8 \times 0.4) + (6 \times 0.2) = 7.4 \end{aligned} \tag{5}$$

Each neuron output a_1, a_2, a_3, and a_4 is then multiplied by different weights to produce the final output values y_1, y_2, y_3, and y_4. This process is illustrated as shown in Equation 6.

$$y_1 = (5.8 \times 0.2) + (8.6 \times 0.2) + (6.4 \times 0.3) + (7.4 \times 0.3) = 7.02 \tag{6}$$

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

$$y_2 = (5.8 \times 0.5) + (8.6 \times 0.1) + (6.4 \times 0.2) + (7.4 \times 0.2) = 6.52$$
$$y_3 = (5.8 \times 0.4) + (8.6 \times 0.2) + (6.4 \times 0.3) + (7.4 \times 0.2) = 7.44$$
$$y_4 = (5.8 \times 0.4) + (8.6 \times 0.3) + (6.4 \times 0.5) + (7.4 \times 0.1) = 8.84$$

The values of y_1 to y_4 are then processed using the softmax function, denoted as $(S(y_i))$, which serves to convert the outputs into probabilities for classification predictions. The softmax function is defined by Equation 7 as follows [24]:

$$S(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{n} e^{y_i}} \tag{7}$$

$S(y_1)$: covid-19, $S(y_2)$: normal, $S(y_3)$: pneumonia, $S(y_4)$: tuberculosis. Based on the equation above, the values of $S(y_1)$ to $S(y_4)$ are presented in Equation 8 as follows

$$S(y_1) = \frac{e^{y_1}}{e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4}} = \frac{e^{7.02}}{e^{7.02} + e^{6.52} + e^{7.44} + e^{8.84}} = 0.11$$
$$S(y_2) = \frac{e^{y_2}}{e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4}} = \frac{e^{7.02}}{e^{7.02} + e^{6.52} + e^{7.44} + e^{8.84}} = 0.07$$
$$S(y_3) = \frac{e^{y_3}}{e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4}} = \frac{e^{6.52}}{e^{7.02} + e^{6.52} + e^{7.44} + e^{8.84}} = 0.16 \tag{8}$$
$$S(y_4) = \frac{e^{y_4}}{e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4}} = \frac{e^{8.84}}{e^{7.02} + e^{6.52} + e^{7.44} + e^{8.84}} = 0.66$$

$S(y_1) + S(y_2) + S(y_3) + S(y_4) = 1$, which indicates that the total probability across all classes equals 1. Based on the calculation results, the highest value is obtained for $S(y_4)$ which is 0.66 or 66%. Therefore, the model prediction points to class y_4, which corresponds to the Tuberculosis category.

Training result

After undergoing a series of processes in the Convolutional Neural Network (CNN) algorithm, the results of training and validation are obtained. Figure 6 presents a comparison between the accuracy and loss values obtained during the training process.

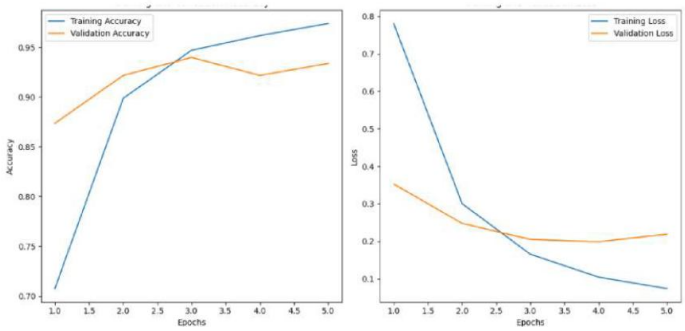


Figure 6. Comparison of Accuracy and Loss Values During the Training Process

Based on Figure 6, the accuracy value represents how well the model can correctly recognize objects, while the loss reflects the degree of error that occurs during classification

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

in each epoch. The loss value serves as an important indicator in evaluating the quality of a machine learning model—the lower the loss, the better the model is considered to perform, and vice versa. During training, the model achieved a loss of 0.073, while the loss on the validation data was 0.219. This indicates that the training loss is lower than the validation loss. Both values are relatively low, suggesting that the developed model performs well. This is further supported by the high accuracy rates achieved—97% on the training data and 93% on the validation data. The accuracy and loss results for both training and validation data are presented in Table 3. Accuracy is a measure of success in recognizing objects, while loss indicates the error rate that occurs when performing classification at each epoch. The loss value can be used as an indicator to evaluate the quality of the machine learning model used. The lower the loss value, the better the machine learning model, and vice versa.

Table 3. Training and Validation Results

Epoch	Train		Validation	
	Accuracy	Loss	Accuracy	Loss
1	0.7074	0.7808	0.8735	0.3525
2	0.8987	0.3003	0.9217	0.2477
3	0.9470	0.1651	0.9398	0.2046
4	0.9617	0.1041	0.9217	0.1986
5	0.9738	0.0734	0.9337	0.2189

The loss value obtained for the training data is 0.0734. The loss value in this training data is lower than the validation data, which is 0.2189. Both values can be considered quite low, which indicates the good quality of the model that has been obtained. This can also be reinforced by the high level of accuracy, reaching 97% for the training data and 93% for the validation data.

Confusion Matrix

To see and evaluate the performance of the trained model, several parameters such as accuracy, recall, and precision can be measured. In calculating these parameters, a matrix that is often referred to as a confusion matrix is used [25]. The confusion matrix obtained from the results of testing lung disease x-ray image data (testing data) can be seen as in Figure 7.

E-ISSN: 2720-9326
P-ISSN: 2716-0459

DOI :

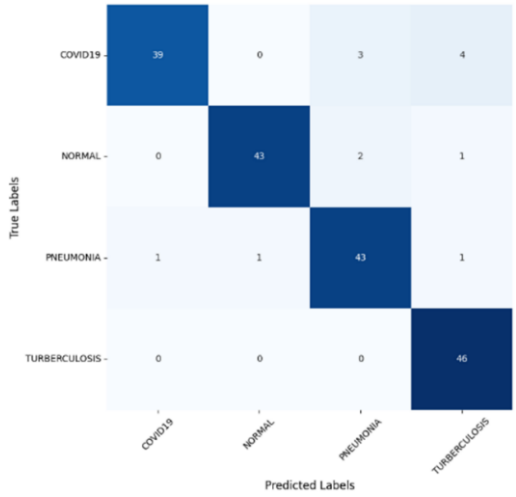


Figure 7. Confusion Matrix

Based on the confusion matrix in Figure 7, the results show that the predicted x-ray image of Covid-19 lungs and it is true that there are 39 images of Covid-19. For images that are predicted to be Covid-19 lungs, there are 1 image that is wrong. Meanwhile, there are 7 images that are predicted not to be Covid-19 lungs but are in fact Covid-19 lungs, namely 3 images predicted as pneumonia lungs and 4 images predicted as tuberculosis lungs. The results of accuracy, precision, recall, and f1-score are as follows.

Table 4. Accuracy, Precision, Recall and F1-Score Results

	Precision	Recall	F1-Score
Covid-19	0.97	0.85	0.91
Normal	0.98	0.93	0.96
Pneumonia	0.90	0.93	0.91
TBC	0.88	1.00	0.94
accuracy = 0.93			

Prediction of New Data

Next, the model is tested by inserting a random x-ray image to perform image detection. The purpose of this step is to identify unknown x-ray images by looking at image patterns. Then from this, the results of the probability percentage and the name label of the lung disease will be seen.

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

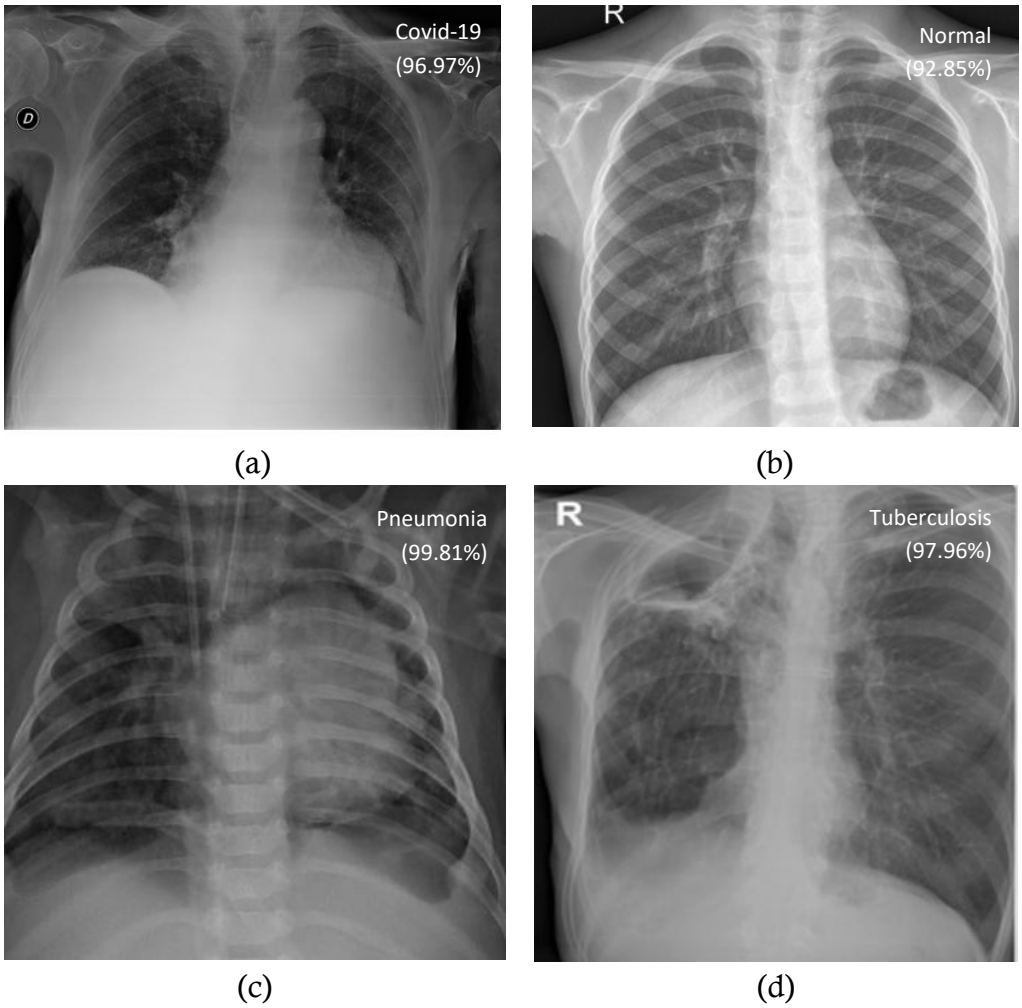


Figure 8. Prediction Result, (a) Covid-19, (b) Normal, (c) Pneumonia, (d) Tuberculosis

Based on the figure 8 above, the lung image detection results show that the image predicted as COVID-19 has a probability score of 96.97%. For the normal lung image, the model provides a prediction with a probability of 92.85%. Meanwhile, the image detected as pneumonia has a probability of 99.81%, and the image predicted as tuberculosis shows a probability of 97.96%. Among these four prediction results, it is evident that the model demonstrates a high level of confidence in each classification. Therefore, it can be concluded that the detection results are accurate and reliable, as reflected by the high probability values across all categories.

Conclusion

Based on the research findings, a Convolutional Neural Network (CNN) model was successfully developed using an input shape of 224×224 pixels, a 3×3 filter/kernel, and trained over 5 epochs. The dataset was composed of 1,490 training images, 166 validation images, and 184 testing images. The evaluation results demonstrate that the model achieved a high classification accuracy of 97% on the training set and 93% on the

E-ISSN: 2720-9326
P-ISSN: 2716-0459
DOI :

validation set. Moreover, the model attained an accuracy of 93% on the testing set, indicating its robust performance in classifying lung X-ray images.

However, this study also has several limitations, including the relatively small dataset size, possible class imbalance, and the absence of validation using real-world clinical data, which may affect its broader applicability. Furthermore, comparisons with other advanced architectures, such as ResNet or EfficientNet, were not conducted, leaving room for future benchmarking studies. For future work, it is recommended to utilize computing devices with higher hardware specifications to further enhance model performance, including machines equipped with larger Random Access Memory (RAM) and the integration of Graphics Processing Units (GPUs) to accelerate the training process.

Expanding the dataset, applying more diverse data augmentation strategies, and incorporating clinical evaluations are also essential to strengthen the robustness, reliability, and diagnostic accuracy of the model. Such enhancements are expected to improve training efficiency and enable the exploration of more complex datasets and deeper architectures, thereby advancing the practical use of CNN-based systems in detecting lung diseases and supporting global healthcare objectives.

References

[1] A. Saputra, "Sistem Pakar Identifikasi Penyakit Paru-Paru Pada Manusia Menggunakan Pemrograman Visual Basic 6.0," *Jurnal Teknologi dan Informatika (Teknomatika)*, vol. 1, no. 3, pp. 202–222, 2011.
[2] D. Djojodibroto, *Respirologi (Respiratory Medicine)*. Jakarta: EGC, 2014.
[3] Yuliana, "Corona virus diseases (Covid-19): Sebuah Tinjauan Literatur," *Wellness and Healty Magazine*, vol. 2, no. 1, p. 187, Feb. 2020, [Online]. Available: https://wellness.journalpress.id/wellness
[4] Dinas Kesehatan Provinsi Bali, "Profil Kesehatan Provinsi Bali Tahun 2016," Bali, 2017.
[5] I. Septiyanti, M. A. Khalif, and E. D. Anwar, "Analisis Dosis Paparan Radiasi Pada General X-Ray II Di Instalasi Radiologi Rumah Sakit Muhammadiyah Semarang," *Jurnal Imejing Diagnostik (JImeD)*, vol. 6, no. 2, pp. 96–102, Jul. 2020, doi: 10.31983/jimed.v6i2.5858.
[6] A. H. Fauziyah, "Deteksi Pneumonia Pada Anak-Anak Dari Citra X-Ray Berbasis Convolutional Neural Network," Institut Teknologi Sepuluh Nopember, 2020.
[7] I. Nurcahyati, T. H. Saragih, A. Farmadi, D. Kartini, and M. Muliadi, "Classification of Lung Disease in X-Ray Images Using Gray Level Co-Occurrence Matrix Method and Convolutional Neural Network," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 4, pp. 332–342, Aug. 2024, doi: 10.35882/jeeemi.v6i4.457.
[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[9] K. Xu, D. Feng, and H. Mi, "Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image," *Molecules*, vol. 22, no. 12, p. 2054, Nov. 2017, doi: 10.3390/molecules22122054.
[10] G. Mezzadri, T. Laloë, F. Mathy, and P. Reynaud-Bouret, "Hold-out strategy for selecting learning models: Application to categorization subjected to presentation orders," *J Math Psychol*, vol. 109, p. 102691, 2022.
[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

E-ISSN: 2720-9326
P-ISSN: 2716-0459

DOI :

[12] P. K. Choudhary and H. N. Nagaraja, *Measuring Agreement*. Wiley, 2017. doi: 10.1002/9781118553282.
[13] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep Learning Applications in Medical Image Analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018, doi: 10.1109/ACCESS.2017.2788044.
[14] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Jan. 2018. [Online]. Available: http://arxiv.org/abs/1603.07285
[15] T. Shafira, "Implementasi Convolutional Neural Networks untuk Klasifikasi Citra Tomat Menggunakan Keras," Universitas Islam Indonesia, Yogyakarta, 2018.
[16] J. Brownlee, *Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. Machine Learning Mastery, 2017.
[17] K. P. Danukusumo, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU," Universitas Atma Jaya Yogyakarta, Yogyakarta, 2017.
[18] K. R. Bokka, S. Hora, T. Jain, and M. Wambugu, *Deep Learning for Natural Language Processing: Solve your natural language processing problems with smart deep neural networks*, 1st ed. Packt Publishing Ltd., 2019.
[19] L. S. Ramba, "Design of a Voice Controlled Home Automation System using Deep Learning Convolutional Neural Network (DL-CNN)," *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 8, no. 1, pp. 57–73, Jun. 2020, doi: 10.34010/telekontran.v8i1.3078.
[20] C. E. Nwankpa, W. I. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," in *2nd International Conference on Computational Sciences and Technologies*, MUET Jamshoro, Dec. 2020. doi: 10.48550/arXiv.1811.03378.
[21] S. Sumahasan, "Object Detection using Deep Learning Algorithm CNN," *Int J Res Appl Sci Eng Technol*, vol. 8, no. 7, pp. 1578–1584, Jul. 2020, doi: 10.22214/ijraset.2020.30594.
[22] M. Muntean and F.-D. Militaru, "Metrics for Evaluating Classification Algorithms," 2023, pp. 307–317. doi: 10.1007/978-981-19-6755-9_24.
[23] N. S. Akbar, T. Zamir, J. Akram, T. Noor, and T. Muhammad, "Simulation of hybrid boiling nano fluid flow with convective boundary conditions through a porous stretching sheet through Levenberg Marquardt artificial neural networks approach," *Int J Heat Mass Transf*, vol. 228, p. 125615, Aug. 2024, doi: 10.1016/j.ijheatmasstransfer.2024.125615.
[24] M. Vakalopoulou, S. Christodoulidis, N. Burgos, O. Colliot, and V. Lepetit, "Deep Learning: Basics and Convolutional Neural Networks (CNNs)," 2023, pp. 77–115. doi: 10.1007/978-1-0716-3195-9_3.
[25] M. Fluorida Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 9, pp. 3112–3123, Sep. 2018, [Online]. Available: http://j-ptiik.ub.ac.id