DOI: 10.15849/IJASCA.221128.01

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 3, November 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Covid-19 and Tuberculosis Detection in X-Ray of Lung Images with Deep Convolutional Neural Network

Firda Rahmatul Ummah, Dina Tri Utari

Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia
e-mail: 18611052@students.uii.ac.id

Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia
e-mail: dina.t.utari@uii.ac.id

Abstract

Tuberculosis is an infectious disease with symptoms similar to those of Covid-19, such as fever, cough, and shortness of breath. Based on the existing cases, these two diseases attack the lungs and can affect their shape. Detection of this disease can be done through a chest X-ray. In the X-ray images of Covid-19 and Tuberculosis, both have ground-glass opacity and consolidation, thus classifying the two diseases is tricky if done manually. One method that can be used for classification is Convolutional Neural Network (CNN). The results obtained from this research are the implementation of the CNN algorithm with four convolutions which are convolution - pooling and repeated four times. The best architecture for parameters epoch 50 with the optimizer ADAM, image size 100x100 pixels, kernel size 3x3, and in the data scenario 80%:20%. The results of the level of accuracy of the classification process in the test data are 85.4%. In addition, the labeling prediction obtained is that the Covid-19 label is predicted to be correct with a probability percentage of 95.85%, while the probability percentage for the Tuberculosis label is 98%.

Keywords: *Covid-19, Tuberculosis, Image, CNN*

1 Introduction

In recent years, the Covid-19 disease outbreak has been feared by many people worldwide. Covid-19 was classified as a global pandemic by the World Health Organization (WHO) on March 11, 2020. However, before the Covid-19 outbreak, a dangerous and quite deadly infectious disease was Tuberculosis. Tuberculosis is an

Received 30 June 2022; Accepted 22 October 2022

infectious disease caused by the bacterium Mycobacterium Tuberculosis with symptoms similar to Covid-19. Furthermore, Tuberculosis became the second-fastest transmission after Covid-19.

According to (Silva et al., 2020), early detection and diagnosis are necessary to suppress the spread of Covid-19 disease. Generally, the Covid-19 test is carried out with a Reverse-Transcription Polymerase Chain Reaction (RT-PCR) test, however this method is less efficient, and the resulting sensitivity is low. Meanwhile, a test to detect Tuberculosis, the sputum test or BTA, is also carried out according to information from halloSehat.com, which is under the auspices of the Indonesia Ministry of Health. The sputum test is not very efficient because currently, the accuracy of the test is only around 50-60%. Based on testing tests for Covid-19 and Tuberculosis, it is quite an obstacle to distinguish or detect the two diseases. Therefore, managing treatment for Covid-19 and Tuberculosis is done through detection by applying X-ray technology. Under (Rawat et al., 2017), X-ray technology is a technology that is entirely accurate for detecting diseases, especially lungs.

Both Covid-19 and Tuberculosis attack the lungs a lot and can affect the shape of the lungs. A chest X-ray showed that about 96% of Tuberculosis patients had ground-glass opacity (GGO) and consolidation in the patient's lungs, which affected the shape of the lungs (Sulaiman et al., 2018). The images produced on chest X-rays are mostly airspace opacities in consolidation and GGO with bilateral distribution, peripheral, and more in the lower part. In contrast, abnormal parenchymal images and pleural effusion rarely appear (Budi et al., 2020). In the X-ray images of Covid-19 and Tuberculosis, both GGO and consolidation are present, thus detecting the two diseases is quite tricky. Rapid detection of Covid-19 and Tuberculosis is vital in doing this. Because the treatment given is slow or even wrong, bad things will be experienced by people with Covid-19 and Tuberculosis. The authors will classify X-ray images on Covid-19 and Tuberculosis based on several existing problems. X-ray classification is necessary because it is complicated to detect manually on X-ray images, in order to need to be experts who are very competent in X-ray reading. Reading or interpreting chest X-ray requires understanding the anatomy and physiology of the chest organs and an understanding of the limitations of this radiological examination. Unfortunately, many doctors, even radiologists, still make mistakes in X-ray readings (Delrue et al., 2010). A study in 2007 showed errors in the interpretation of chest X-rays on an average of 3-5% per day. Taking an average of 4% of errors made daily, of the 1 billion chest X-rays performed annually, radiologists make 40 million errors in interpreting chest X-rays each year (Berlin, 2007; Brady, 2017). Whereas according to the 2012 Indonesian Doctor Competency Standards (SKDI), the interpretation of chest X-rays is at level 4a, which means that general practitioners should be able to perform these clinical skills independently and without supervision (Konsil Kedokteran Indonesia, 2012). Therefore, the authors classified the X-ray images using the Deep Convolutional Neural Network (CNN). We hope that the results of this study can be used by doctor particularly doctor or radiologists to detect the disease based on chest X-ray images. Early detection can help doctor to give the appropriate treatment. CNN is a part of deep learning. According to (Chauhan et al., 2018), CNN is a Neural Network that is commonly used in image data. In addition, it is often used to detect and identify objects in images that can outperform the old methods in computer vision and pattern identification. Furthermore, an accurate CNN model with a minimum number of parameters is required to be applied to a device or computer without burdening computation (Zebin et al., 2019).

2 Related Work

Deep learning has considerably increased typical medical applications, especially medical image-based diagnostics. Neural Network surpasses other traditional models and image analysis methods (Ahmad, 2021; Lundervold et al., 2019). Due to the encouraging results of CNN's analysis and classification of medical images, these are considered the actually standard in this field (Kayalibay et al., 2017; Li et al., 2014). Since the advent of Covid-19 and Tuberculosis, many researchers have conducted experiments on diagnosing, treating, and managing these diseases.

Researchers in (Suci Aulia et al., 2021) report the importance of the applicability of Deep learning Convolutional Neural Networks (CNN) in image analysis to detect cases of Covid-19 and Tuberculosis. One of the applications is the decision support system (DSS) that helps medical professionals, especially doctors, diagnose TB in grades one, 2+, and 3+ fast. Therefore, DSS is urgently needed and can be used in the long term to deal with Covid-19. This study proposes a rapid classification of normal lung, Tuberculosis lung, and Covid-19 lung based on chest radiograph (CXR) as the first step to implementing DSS. The proposed CNN image processing-based CXR classification achieved the highest accuracy of 88.37%. The proposed and developed system is intended to help doctors diagnose lung disease.

Researchers in (Ali H. Al-Timemy et al., n.d.) propose extraction of Deep Features (DF) from chest X-rays, techniques available in most hospitals, and subsequent classifications using machine learning methods that do not require large amounts of computational resources. They collected a data set of five classes of chest X-rays, including a balanced number of Covid-19, viral pneumonia, bacterial pneumonia, TB, and healthy cases. They compared the performance of a pipeline that combined 14 highly trained deep networks for DF extraction with a traditional machine learning classifier. The results show the potential benefits of using channels to detect Covid-19. In particular, it depends on accessible X-rays and can operate with limited computational resources so that it can be used in resource-constrained configurations.

Researchers in (Razief Moch Diar et al., 2022) used the MobileNet architecture to design an automated lung disease classification system based on CNN-based lung radiographs. They classify image data into four classes: viral pneumonia, Covid-19, normal, and Tuberculosis. The final results of this study show that the best optimizer is Adam. It uses CLAHE pre-processing on Epoch 50 and produces an accuracy value of 94.687 and a loss of 0.148. In addition, system performance results are 95% accuracy, 93% recall, and 94% F-1 score.

Based on the literature, we use the Deep CNN to classify X-ray of lung images of Covid-19 and Tuberculosis. Some previous studies only used one convolution, but in this study, we tried to use four convolutions to filter information and produce a more significant feature map. Based on the results, we also performed labeling predictions for the new Covid-19 and Tuberculosis X-ray images. Furthermore, the results of this study can be used by doctors or radiologists to help make it easier to diagnose Covid-19 and Tuberculosis based on X-ray of lung images. The results of this study are expected to facilitate the interpretation of X-ray of lung images to minimize errors.

3 Materials and Methods

In this study, we use X-ray image data of the lungs for Covid-19 and Tuberculosis. Based on this data was obtained from the website kaggle.com (Chowdhury et al., 2020; Rahman et al., 2020). The data used is 600 data, namely 300 Covid-19 X-ray image data and 300 Tuberculosis X-ray image data. The following figure is an X-ray of lung image for Covid-19 and Tuberculosis.

Fig. 1. X-ray of Lungs (A: Covid-19, B: Tuberculosis)

In this study, we consider using convolution and pooling, which are repeated four times. Since we hope that the classification results obtained are more accurate because the information from the filtered image is more comprehensive than only one convolution. Previous studies, as in (A. H. Al-Timemy et al., 2020; S. Aulia et al., 2021; R. M. Diar et al., 2022; Giełczyk et al., 2022; Mabrouk et al., 2022; Soffer et al., 2019; Sorić et al., 2020; Yusoff et al., 2021), one convolution and only completed the classification stage. In addition, this study also conducted several scenarios for training and test data and then carried out the training process with three different types of optimizers. In addition, in this study, the obtained model can then be used to predict an image with the probability that the image will be detected in a particular disease.

3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a method of popular neural network. CNN can process two-dimensional data, such as sound and images. In general, the way CNN works is almost the same as a neural network. However, there is a significant difference, namely using a 2-dimensional or high-dimensional kernel in the CNN layer unit that will perform

the convolution. The CNN kernel combines spatial features with a spatial form that resembles the input medium. Based on the use of CNN, various parameters are used to reduce the number of variables, making it easy to study (Khan et al., 2018).
The CNN layers are generally divided into two, namely the first feature extraction layer and the second classification layer. CNN can be trained to predict the features of an object. The following illustration of CNN is generally depicted in the architecture in **Fig. 2.**

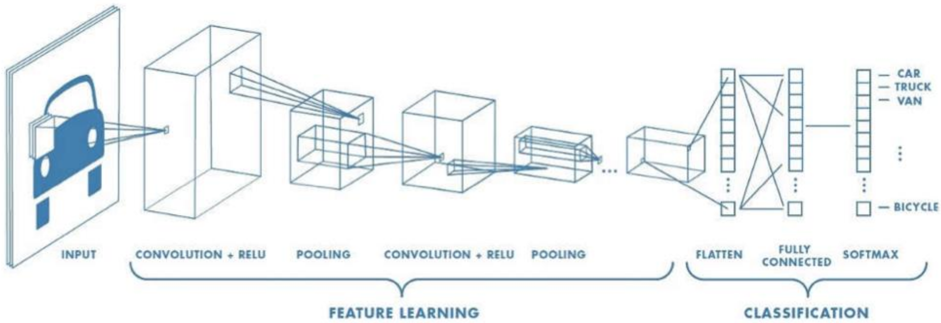


Fig. 2. CNN Architecture (Aliady et al., 2018)

Based on **Fig. 2.**, there are two layers of the CNN architecture. The first is feature learning which consists of a convolution layer and a pooling layer. Each of these layers will produce feature maps in the form of numbers so that it can represent images to be forwarded to the classification layer section. The second is the classification layer which consists of fully connected layers with other layers. This layer receives the input generated from the output layer for the feature learning section. Then in the feature learning section, it will be processed in flatten by adding to the hidden layer on a fully connection to produce output in the form of accuracy values for each classification class.

1. Convolution Layer

The convolution layer is a special kind of linear operation. In convolution networks, there are neural networks that use convolution only instead of applying the general matrix, using at least one layer. Convolution is one of the mathematical operations used for image processing. In the convolution operation, the output function is applied as feature maps of the input. The input and output can be seen as an argument with an accurate value. The calculation of feature maps that will be generated can be searched with the following formula (Aliady et al., 2018):

$$n_{out} = \left(\frac{n_{in}-k+2p}{s}\right) + 1 \quad (1)$$

where n_{out} is feature map size, n_{in} is input matrix size, k is filter matrix size, p is zero padding, and s is stride. Stride is a parameter that determines how many filter shifts (Aliady et al., 2018).
Based on the convolution layer, the layer consists of neurons that arranged to form a filter with length and height (pixels). For example, the first layer is to do a convolutional layer with a size of 6x6x3. In the description, 6 is the pixel length and pixel height. Then 3 is

the thickness or number according to the channel of the image. These three filters will be shifted overall on the part of the image. Each shift will operate in the form of a "dot" between the input and the value of the filter. This step produces an output or feature map. The following explains the convolution operation based on the figure below.

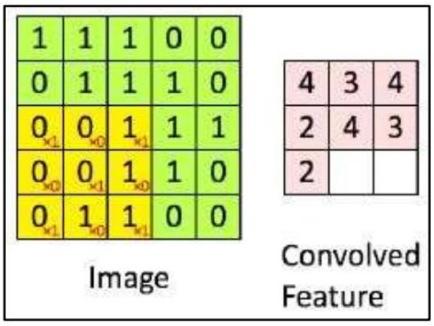


Fig. 3. Convolution Operation (Peryanto et al., 2020)

Based on **Fig. 3.**, all green squares are images that will perform convolution. Movements in the kernel will be executed from the top left corner to the bottom right. Then the results obtained can be seen in the image on the right. Convolution is used to extract features from the input image. Convolution can produce linear transformations obtained from input data according to specific information in the data. The layer weight will specify the convolution kernel to be used (Yamashita et al., 2018).

2. Polling Layer

The pooling layer is a layer in input and processing carried out by various statistical operations based on the feature map function's closest pixel value. The pooling layer is used to reduce the size obtained on feature maps. Based on the type of pooling that is often used is max pooling. In this case, the pooling layer process is the same as the convolution layer, namely by sliding the window over the entire surface of the image. However, this window is used as a reference in selecting the maximum value in a specific area. In this process, a feature maps matrix will be generated, which is filled with selected maximum values (Nagi et al., 2011).

In general, the pooling layer used is a 2x2 filter that will be applied in two steps and will operate on every slice of the input. This form will be able to reduce the feature maps to 75% of the original size. The following is an example of a pooling matrix using 2x2 max pooling with two strides so that at each filter shift, the maximum value in the 2x2 pixel area will be selected. The following is an example of a pooling layer figure.

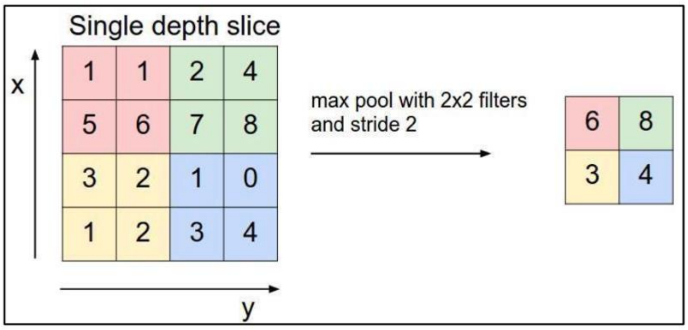


Fig. 4. Polling Layer Operation (Peryanto et al., 2020)

The use of the pooling layer aims to reduce the dimensions of the feature map. Therefore it can speed up computing because there are fewer parameters that need to be updated and can overcome overfitting (Aliady et al., 2018).

3. Flatten

Flatten is a process in the reshape feature map which will become a vector to be used as input from the fully connected layer (Szarvas et al., 2005; Szegedy et al., 2013). In the flattening process, the results of the flatten value on the layer are obtained by multiplying the last size obtained from the feature learning process with the filter used.

4. Fully Connected Layer

According to (Hijazi et al., 2015), the fully connected layer is a collection obtained from the convolution process. In accordance with (Albewi et al., 2017), the layer will get input from the previous process as feature determination, and this feature will be selected to see which class is most correlated.

In the fully connected layer, the application of Multilayer Perceptron (MLP) is commonly used in that layer. In addition, it also aims to perform dimensional transformations on the data so that the data can be classified linearly. Each neuron needs to transform into one-dimensional data based on the fully connected layer. Then new neurons can be inserted into a fully connected layer, and it can cause the data to lose spatial and irreversible information. Therefore, a fully connected layer is implemented at the last network (Basha et al., 2020).

5. Activation Function

The activation function is a non-linear function that allows the function of the neural network to change with the result that data can be fed to higher dimensions. Thus allowing simple traversing of paths for classification.

a) ReLu

ReLu or Rectified linear unit is the most common and basic way of introducing a non-linear artificial network. The ReLu activation function only has a $\max(0, x)$ value (Heaton, 2015). The following is the mathematical equation for the ReLu activation function,

$$\mathrm{A(x)} = \max(0, x) \tag{2}$$

The ReLu function will perform thresholding at a zero value which will be carried out on the pixel value in the input image. Based on this activation, all pixel values that are less than zero in an image will be assigned a value of 0.

b) Sigmoid
The sigmoid function is a function that transforms the range of values obtained from input x to 0 and 1. The following is the formula of the sigmoid activation function (Wang et al., 2020),

$$A = \left(\frac{1}{1+e^{-x}}\right) \quad (3)$$

6. Optimizer

An optimizer is an algorithm that aims to determine the optimal weight by minimizing errors and maximizing accuracy values. In the training process, the weights or parameters can be changed. It is used to minimize the loss function thus it can predict as accurately as possible (Bera et al., 2019).

a) Root Mean Square Propagation (RMSProp)
RMSProp is a particular version of Adagrad developed in the neural network class. RMSprop is a powerful optimizer in normalized gradients with the latest gradient magnitude values. Therefore, gradient normalization, i.e., keeping the moving average above the root mean square gradient, is called RMS. One of the advantages of RMSProp is that it can handle stochastic objectives well (Bera et al., 2019).

b) Stochastic Gradient Descent (SGD)
SGD is the most straightforward optimizer, but it takes a long time to detect convergence for the operations carried out. SGD is used to update parameters in the form of weights and biases. SGD also has the advantage that the memory contained in SGD can be minimized, which is used in need to find new weight values (Bera et al., 2019). In the SGD for the updating process, a parameter can be defined in the following equation:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} L(\theta_t) \quad (4)$$

where:
θ_{t+1} : fixed parameters (weight)
η : learning rate
$\nabla_{\theta_t} L(\theta_t)$: gradient against weight

c) Adaptive Moment Optimization (ADAM)
ADAM is an optimization algorithm based on first-order gradients which are computationally efficient and requires low memory. ADAM is one of the adaptive learning rate optimizations to combine RMSProp and momentum (Bera et al., 2019). The following is the equation for ADAM's optimization calculation,

$$\theta_{t+1} = \theta_t - \alpha \frac{n}{\sqrt{\widehat{v_t}}} \widehat{m_t} \quad (5)$$

where:
α : learning rate
$\widehat{m_t}$: momentum estimation
$\widehat{v_t}$: sub-gradient estimation

The ADAM optimizer has several advantages: it can perform computational efficiency, save memory, and is suitable for various non-convex optimization problems in the machine learning field (Bera et al., 2019).

7. Proposed Study

The proposed Deep CNN architecture will be seen from the results of the epoch, optimizer, and selected dataset. The following is **Fig. 5.**, which is a schematic of CNN architecture.

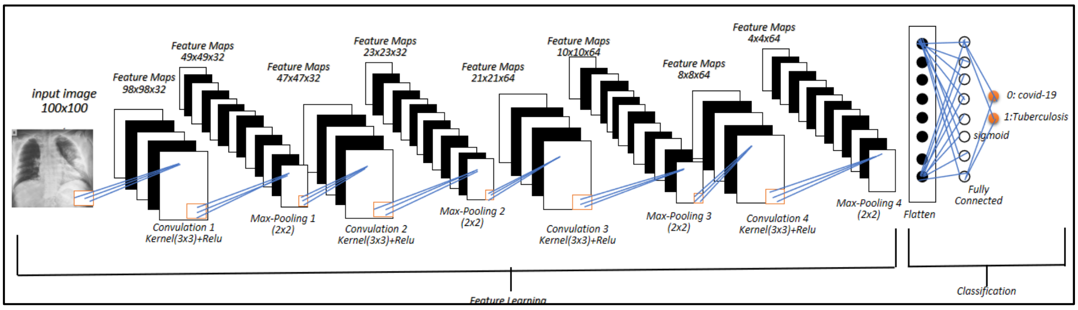


Fig. 5. Deep CNN Architecture Scheme

In this study, there were four convolutions, where the second, third, and fourth convolutions were performed after each pooling. The kernel size used is 3x3. It makes the size can be shifted to all parts of the pixel image that is inputted by 100x100. Therefore, the number of filters used is 32 for convolution layers 1 and 2 and 64 for convolutions 3 and 4. Then when the process is complete, an output called activation maps or features maps will be obtained, assisted by the ReLu activation function. ReLu activation changes the negative output to 0, and the positive value will be the activation value itself.

We use a max-pooling of 2x2 to calculate features with a size of 98x98. Therefore, the most considerable value for filter shift is taken in the 2x2 area.

Then after we get the output in the form of feature maps in the form of a multidimensional array, it will go to the next stage; the flatten stage. This step generates a vector for the input of the fully connector layer.

At this stage, the researcher only uses one hidden layer for the MLP network. Based on this process, the dropout value is used before the image classification or prediction process. Then it will be entered into the classification process with the help of sigmoid activation; thus, it is classified into targets, namely two categories of Covid-19 and Tuberculosis.

4 Results and Discussions

4.1 Pre-processing

At this stage, feature equalization is carried out because the elements of each image vary greatly and clarify the features of the X-ray image. We pre-processed the data in terms of resizing. The best size is chosen by considering the length of time in the training process.

Table 1. Processing Time Based on Size

Size	Training time
100x100	7s/step
200x200	41s/step
300x300	76s/step
400x400	92s/step
500x500	166 s/step

Based on the results in Table 1., we use a size of 100x100 because the fastest time is 7s/step. This result makes the analysis process work well. Then it will pre-process the dataset with a thresholding process, and this process is a process makes the image value 0-1 for black and 0-255 for white. Next, we use the grayscale color, producing a gray color that matches the X-ray image.

4.2 Data Scenario

The X-ray image data used are 300 Covid-19 data and 300 Tuberculosis data. In this discussion, the researcher has divided the dataset into several ratios. It is split into training and validation data. Furthermore, 10% of the data will be taken from the training data used for the testing process.

Table 2. Dataset Scenario

Scenario	Training Data	Validation Data	Testing Data
70%:30%	378	180	42
80%:20%	432	120	48
90%:10%	486	60	64

4.2 Deep CNN Model Design

At this stage, we will conduct several experiments to compare the Deep CNN model for the best classification results. The best Deep CNN architecture is based on the parameters used. This study compares epoch values, optimizer, and dataset scenarios.

1. Comparing Epoch and Optimizer

We will see the results of accuracy and loss obtained from the train and validation data. We will use this comparison to get the best Deep CNN model to produce a reasonably high accuracy value. We use the epoch values of 25, 50, 100, 150, and 200 and the RMSProp, SDG, and ADAM optimizers.

Table 3. Accuracy and Loss of Training Process

Optimizer	Epoch	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
RMSProp	25	0.92	0.21	0.86	0.42
	50	0.94	0.18	0.88	0.38
	100	0.94	0.19	0.87	0.40
	150	0.90	0.22	0.86	0.48
	200	0.88	0.29	0.83	0.41

Optimizer	Epoch	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
SGD	25	0.50	0.68	0.50	0.68
	50	0.86	0.44	0.73	0.52
	100	0.90	0.36	0.82	0.22
	150	0.87	0.44	0.75	0.50
	200	0.90	0.46	0.81	0.38
ADAM	25	0.92	0.21	0.88	0.36
	50	**0.95**	**0.15**	**0.92**	**0.26**
	100	0.90	0.23	0.87	0.38
	150	0.93	0.21	0.84	0.42
	200	0.94	0.18	0.89	0.33

Based on Table 3., the best optimizer is ADAM with an epoch value of 50. This combination provides the highest accuracy value and the lowest loss for training and validation data.

2. Comparing Scenario Dataset

Afterward got the best epoch value and optimizer type, our next step will be to compare dataset scenarios based on the value of accuracy and loss in the training and validation data.

Table 4. Accuracy and Loss of Training Process for Each Scenario

Dataset	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
70%:30%	0.94	0.17	0.83	0.56
80%:20%	0.95	0.15	0.92	0.26
90%:10%	0.92	0.26	0.90	0.26

Table 4. shows the results of the accuracy and loss values are obtained based on the comparison of datasets. In the three comparisons, the highest accuracy value of the training and validation data is in the 80%:20% comparison scenario. Therefore, it can be concluded that by using the 80%:20% dataset scenario, the machine used can classify images well. The best model is obtained with a total number of parameters of 204,513 with details in Table 5.

Table 5. Best Model Parameter

Name	Size	Parameter
Conv2d_1	(None, 98, 98, 32)	320
MaxPool_1	(None, 49, 49, 32)	0
Conv2d_2	(None, 47, 47, 32)	9,248
MaxPool_2	(None, 23, 23, 32)	0
Conv2d_3	(None, 21, 21, 64)	18,496
MaxPool_3	(None, 10, 10, 64)	0
Conv2d_4	(None, 8, 8, 64)	36,928
MaxPool_4	(None, 4, 4, 64)	0

Name	Size	Parameter
Flatten	(None, 1024)	0
Dense_1	(None, 128)	131,200
Dense_2	(None,64)	8,256
Dense_3	(None, 1)	65
Total Param: 204,513		
Trainable Param: 204,513		
Non-Trainable Param:0		

The convolution process will cause the size of the image to be smaller. In Table 5. the final size is obtained before entering into a fully connected, namely at the size of 4x4 pixels, using 64 filter parameters. Then the following process is to change the matrix data into vectors so that it can enter the fully connection process thus the results obtained are 1,024 neurons that can be forwarded. Furthermore, the process will be added with a dropout function with the result that it can classify images into two categories of Covid-19 and Tuberculosis.

4.2 Classification Result Based on Best Model

The confusion matrix compares the actual target values in the Deep CNN model obtained. A table will be displayed stating the amount of test data on the actual and false target classifications. The following is the result of the confusion matrix.

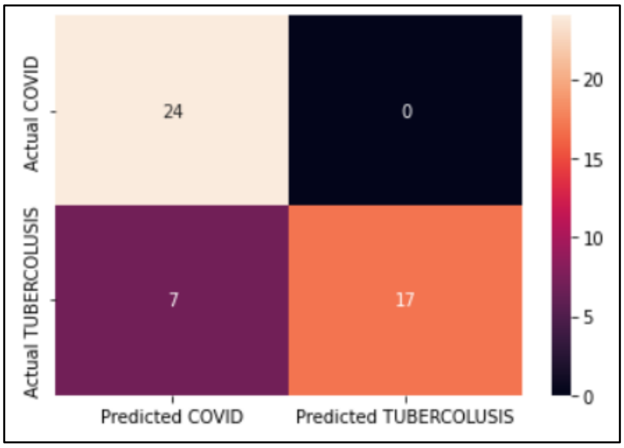
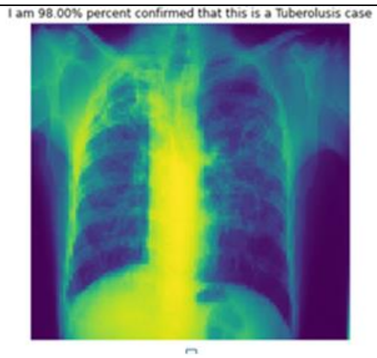


Fig. 6. Confusion Matrix

Based on the confusion matrix results, the Precision and Recall values for each category can be calculated, 0.77 and 1.00 for Covid-19. Then for Tuberculosis of 1.00 and 0.71. Furthermore, for the results obtained, the accuracy value obtained from the classification with the best Deep CNN model on testing data is 0.854. Based on this accuracy, the image can be predicted correctly: 24 Covid-19 and 17 Tuberculosis data.
The results of the best Deep CNN model are also used to determine the labelling of new data based on the pattern of the image. It will be seen from the probability percentage and

the name label of the disease affect the shape of the lungs. The following table shows the results of the predictions carried out in the Covid-19 and Tuberculosis labels.

Table 6. Prediction result

	Covid-19	Tuberculosis
Probability percentage	95.85%	98.00%
		

Based on the Covid-19 label, it is predicted to be Covid-19 with a probability percentage of 95.85%. The Tuberculosis label is predicted to be Tuberculosis with a 98.00% probability percentage. Therefore, it can be concluded that the predictive results for the overall label on the classification of lung shapes affected by Covid-19 and Tuberculosis are well-predicted, evidenced by the high probability percentage.

5 Conclusion

The best implementation of the Deep CNN algorithm that can classify the shape of the lungs affected by Covid-19 and Tuberculosis is to find the best architecture by comparing several parameters. Four convolutions were performed in this study, followed by second, third, and fourth convolutions after each pooling. The best model is obtained for the parameter epoch 50 with ADAM optimizer, image size of 100x100 pixels, kernel size of 3x3, and the 80%:20% data scenario, with an accuracy of 85.4%.

The prediction results of the overall label on the classification of the shape of the lungs affected by Covid-19 and Tuberculosis are predictably good based on a high percentage of probability.

References

1. Ahmad, M. (2021). Ground truth labeling and samples selection for hyperspectral image classification. *Optik*, *230*.
2. Albewi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy*, *19*(6), 242.
3. Aliady, H., & Utari, D. T. (2018). GPU Based Image Classification using Convolutional Neural Network Chicken Dishes Classification. *Int. J. Advance Soft Compu. Appl*, *10*(2), 1–13.

4. Al-Timemy, A. H., Khushaba, R. N., Mosa, Z. M., & Escudero, J. (2020). *An Efficient Mixture of Deep and Machine Learning Models for COVID-19 and Tuberculosis Detection Using X-Ray Images in Resource Limited Settings*.
5. Al-Timemy, Ali H., Khushaba, R. N., Mosa, Z. M., & Escudero, J. (n.d.). *An Efficient Mixture of Deep and Machine Learning Models for COVID-19 and Tuberculosis Detection Using X-Ray Images in Resource Limited Settings*.
6. Aulia, S., Hadiyoso, S., Mengko, T. L. E. R., & Suksmono, A. B. (2021). Covid-19 and Tuberculosis Classification Based on Chest X-Ray Using Convolutional Neural Network. *Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics* , 407–420.
7. Aulia, Suci, Hadiyoso, S., Mengko, T. L. E. R., & Suksmono, A. B. (2021). Covid-19 and Tuberculosis Classification Based on Chest X-Ray Using Convolutional Neural Network. *Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics* , 407–420.
8. Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, *378*, 112–119.
9. Bera, S., & Shrivastava, V. K. (2019). Analysis of various optimizers on deepconvolutional neural network model in theapplication of hyperspectral remote sensing imageclassification. *International Journal of Remote Sensing*, *41*(7), 2664–2683.
10. Berlin, L. (2007). Radiologic errors and malpractice : a blurry distinction. *The Practice of Radiology*, *189*(3), 517–522.
11. Brady, A. P. (2017). Error and discrepancy in radiology: inevitable or avoidable? *Insights Imaging*, *8*(1), 171–182.
12. Budi, Y., & Hayatun, U. (2020). Peran pemeriksaan radiologis pada diagnosis Coronavirus disease 2019. *Jurnal Kedokteran Syiah Kuala*, *20*(1).
13. Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. *First International Conference on Secure Cyber Computing and Communication*.
14. Chowdhury, M. E. H., Rahman, T., Khandakar, A., & Mazhar, R. (2020). Can AI help in screening Viral and COVID-19 pneumonia? *IEEE Access*, *8*, 132665–132676.
15. Delrue, L., Gosselin, R., Ilsen, B., Landeghem, A. van, de Mey, J., & Duyck, P. (2010). Difficulties in the Interpretation of Chest Radiography. In Medical Radiology (pp. 27–49).
16. Diar, R. M., Fu'Adah, R. Y. N., & Usman, K. (2022). Klasifikasi Penyakit Paru-paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network. *EProceedings of Engineering*, *9*(2).
17. Diar, Razief Moch, Fu'Adah, R. Y. N., & Usman, K. (2022). Klasifikasi Penyakit Paru-paru Berbasis Pengolahan Citra X Ray Menggunakan Convolutional Neural Network . *EProceedings of Engineering*, *9*(2).
18. Giełczyk, A., Marciniak, A., Tarczewska, M., & Lutowski, Z. (2022). Pre-processing methods in chest X-ray image classification. *PLoS ONE*, *17*(4).
19. Heaton, J. (2015). *Artificial Intelligence for Humans: Deep learning and neural*

networks. Heaton Research, Incorporated.

20. Hijazi, S., Kumar, R., & Rowen, C. (2015). Image Recognition Using Convolutional Neural Networks. *Cadence Whitepaper,* 1–12.
21. Kayalibay, B., Jensen, G., & van der Smagt, P. (2017). *CNN-based segmentation of medical imaging data*.
22. Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. *Synthesis Lectures on Computer Vision*, *8*(1), 1–207.
23. Konsil Kedokteran Indonesia. (2012). *Standar kompetensi dokter Indonesia*. Http://Www.Kki.Go.Id/Assets/Data/Arsip/SKDI_Perkonsil,_11_maret_13.Pdf.
24. Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. *Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, 844–848.
25. Lundervold, A. S., & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift Für Medizinische Physik*, *29*(2), 102–127.
26. Mabrouk, A., Redondo, R. P. D., Dahou, A., Elaziz, M. A., & Kayed, M. (2022). Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. *Applied Sciences*, *12*(6448), 2–15.
27. Nagi, J., Ducatelle, F., di Caro, G. A., Ciresan, D., Meier, U., & Giusti, A. (2011). Max-Pooling Convolutional Neural Networks forVision-based Hand Gesture Recognition. *IEEE International Conference on Signal and Image Processing Applications*.
28. Peryanto, A., Yudhana, A., & Umar, R. (2020). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format : Jurnal Ilmiah Teknik Informatika*, *8*(2).
29. Rahman, T., Khandakar, A., Kadir, M. A., & Islam, K. R. (2020). Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning, Segmentation and Visualization. *IEEE Access*, *8*, 191586–191601.
30. Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, *29*(9), 2352–2449.
31. Silva, P., Luz, E., Silva, G., Moreira, G., Silva, R., Lucio, D., & Menotti, D. (2020). COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis. *Informatics in Medicine Unlocked*, *20*(100427).
32. Soffer, S., Ben-Cohen, A., Shimon, O., Amitai, M. M., Greenspan, H., & Klang, E. (2019). Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide. *Radiology*, *290*(3), 590–606.
33. Sorić, M., Pongrac, D., & Inza, I. (2020). Using Convolutional Neural Network for Chest X-ray Image classification. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*.
34. Sulaiman, S. C., Handayani, L., Suwandi, M. Y. S., & Soedarsono, S. (2018). Gambaran Radiografi Tuberkulosis Paru Multidrug-Resistant: Studi Retrospektif di Rumah Sakit Umum Dr. Soetomo Surabaya. *Jurnal Respirasi*, *4*(3).
35. Szarvas, M., Yoshizawa, A., Yamamato, M., & Ogata, J. (2005). Pedestrian Detection with Convolutional Neural Networks. *Intelligent Vehicles Symposium IEEE*, 224–229.

36. Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep Neural Networks for Object Detection. *Advances in Neural Information Processing Systems*, 2553–2561.
37. Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. *Applied Sciences*, *10*(1897).
38. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging Volume*, *9*, 611–629.
39. Yusoff, M., Saaidi, M. S. I., Afendi, A. S. M., & Hassan, A. M. (2021). Tuberculosis X-Ray Images Classification based Dynamic Update Particle Swarm Optimization with CNN. *Journal of Hunan University Natural Sciences*, *48*(9), 504–516.
40. Zebin, T., Scully, P. J., Peek, N., Casson, A. J., & Ozanyan, K. B. (2019). Design and Implementation of a Convolutional Neural Network on an Edge Computing Smartphone for Human Activity Recognition. *IEEE Xplore*, *7*, 133509–133520.

Notes on contributors

Firda Rahmatul Ummah is a student of the Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia, Indonesia. She graduated in 2022, and her research interest is in data science.

Dina Tri Utari is lecturer at the Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia, Indonesia. Her main teaching and research interests include Big Data Analytics, Optimization, Computational Statistics, and Data Science. She has published several research articles in international journals.