Convolutional neural network modeling for classification of pulmonary tuberculosis disease

Lingga Aji Andika, Hasih Pratiwi, Sri Sulistijowati Handajani
Department of Statistics, Universitas Sebelas Maret, Indonesia

Corresponding author: linggaajiandika@gmail.com

Abstract. Tuberculosis (TB) is caused by bacteria (Mycobacterium tuberculosis) that most often affect the lungs so called Pulmonary Tuberculosis or PTB. Diagnosis using a chest radiograph image manually by doctor requires a long time, even difficult to detect PTB. Convolutional neural network (CNN) is a deep learning method that adopts the performance of human brain neurons called neural network and convolution functions to classify images. CNN can also help classify PTB based on chest radiograph images. This study uses data from the National Library of Medicine, Maryland, USA in collaboration with Shenzhen No.3 People’s Hospital, Guangdong Medical College, Shenzhen, China including 663 images entered into two classes, normal and PTB. This study uses adaptive momentum optimization (Adam) which serves to improve the accuracy of the model. The classification results of the models built were 99.19% for training data and 80.60% for validation data with 75 epochs, and accuracy in the test data was 84% which means that the model was able to qualify 84% of the test data into normal classes and PTB appropriately. 25 correctly classified as normal lungs, 5 incorrectly and 26 correctly classified as PTB and 5 incorrect.

1. Introduction
Tuberculosis (TB) is caused by bacteria (Mycobacterium tuberculosis) that most often affect the lungs. TB is curable and preventable [1]. TB is spread from person to person through the air. When people with lung TB cough, sneeze or spit, they propel the TB germs into the air. About one-quarter of the world's population has latent TB, which means people have been infected by TB bacteria but cannot transmit the disease. People infected with TB bacteria have a 5–15% lifetime risk of falling ill with TB. TB mostly affects adults in their most productive years. However, all age groups are at risk. Over 95% of cases and deaths are in developing countries. People who are infected with HIV are 20 to 30 times more likely to develop active TB. The risk of active TB is also greater in persons suffering from other conditions that impair the immune system. One million children (0–14 years of age) fell ill with TB, and 230,000 children (including children with HIV associated TB) died from the disease in 2017. Tobacco use greatly increases the risk of TB disease and death. 7.9% of TB cases worldwide are attributable to smoking. TB often occurs in the lungs so it is called pulmonary tuberculosis or PTB [1].

Early diagnosis of PTB has a big impact on a patient's life. The diagnosis of PTB is generally done clinically (physical symptoms by a doctor). In addition, PTB can also be diagnosed through chest radiographs. Chest radiograph examination is one of the most commonly used medical imaging examinations because it is more affordable. Chest radiograph photo reading has a disadvantage, which is difficult to detect the disease, so it takes a long time before medical personnel or doctors diagnose the disease suffered by the patient. One method to overcome these problems is to classify chest radiograph...
images into specific classes using machine learning. The method used to process data is multilayer perceptron (MLP) [2]. MLP has drawbacks for some types of data, especially for images, MLP is not well adapted so that it loses the spatial information contained in the image. Deep learning is one branch of science from machine learning or what is often referred to as machine learning. Deep learning is a disruptive technique that changes the old paradigm in machine learning. Deep learning has a much higher level of accuracy compared to shallow learning, namely machine learning methods that existed previously such as MLP.

Data in the form of chest radiograph images are two-dimensional image data that can be processed using deep learning with the Convolutional Neural Network (CNN) method [3]. CNN is a neural network that uses convolution instead of general matrix multiplication, where there is at least one convolution in each layer. CNN is able to classify images with a very high degree of accuracy because it can reduce the number of free parameters and can handle input image deformations such as translation, rotation, and scale.

Previous related work was automatic tuberculosis screening using chest radiographs [4]. The research classifies PTB by comparing several architectures using lung segmentation and object detection. The weakness in the research is using shallow learning in machine learning methods, one of which is the support vector machine (SVM). To overcome these deficiencies, in this study using the deep learning method, CNN, besides that it compares the best accuracy in training data by using several iterations or epochs.

2. Convolutional Neural Network

The convolutional network, also known as the Convolutional Neural Network (CNN), is a type of neural network specifically for processing data that has similar grid topology [5]. The name CNN indicates that the network uses a mathematical operation called convolution. CNN is a neural network that uses convolution instead of general matrix application where there is at least one convolution at each layer. In mathematics, convolution is a mathematical operation on two functions to produce a third function that expresses how one form is modified by another. The term convolution refers to the result function and its calculation process. In convolutional network terminology, the function of x to the network is often called input, the function w is called the kernel, and the output is sometimes produced as a feature map or activation map, which is a mapping where certain features are found in an image. Since it is assumed t is calculated per second so that t is a round number, then the discrete convolution equation is:

\[ s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a) w(t - a) \]

2.1. Convolutional layers

The convolution layer is the core part of CNN where some computation is done at this layer. The output convolutional layers is obtained from the kernel convolution of parts of the image. Then, the kernel moves by the number s pixels, s called stride.

When the steps are small, redundant information is obtained. Sometimes the input is added to zero padding, which is a margin containing zero values around the image to control the size of the output. It is assumed that the C0 kernel (also called a filter) is applied, each measuring k × k in the figure, where k is the spatial level. If the input image size is \( W_i \times H_i \times C_i \) (\( W_i \) shows width, \( H_i \) height, and \( C_i \) number of channels or depth, usually \( C_i = 3 \) (Red, Green, Blue)), the output volume is \( W_0 \times H_0 \times C_0 \) where \( C_0 \) corresponds to the number of K kernel, so we get a summary of the convolution layer:

\[
W_0 = \frac{W_i - k + 2p}{s} + 1
\]

\[
H_0 = \frac{H_i - k + 2p}{s} + 1
\]
$C_0 = K$

Figure 1 shows how the convolution layer works [6].

2.2. Pooling layers
The pooling layer functions progressively reduce the size of the spatial representation which has the effect of reducing the number of parameters and calculations in the network, also called subsampling, and also controlling overfitting. Pooling layers are used periodically between convolution layers [6]. Max pooling is the pooling layer that takes the maximum value of each kernel operation with input. Max pooling is used more than other pooling such as average pooling [7]. Figure 2 [8] explains the application of max pooling.

2.3. Fully-connected layers
The last layer in the convolutional neural network process is a fully connected layer that functions to change the multidimensional array into a vector. Each neuron in the convolution layer needs to be transformed into one-dimensional data before it can be put into a fully-connected layer. Because it causes data to lose spatial information and is not reversible, while the fully-connected layer can only be implemented at the end of the network. The activation of this layer can be calculated by multiplying the matrix followed by bias. Figure 3 [9] explains the application of fully connected layer.
Figure 3. Fully connected layer

2.4. Confusion Matrix
Confusion Matrix is a matrix that represents the results of classification in a dataset [10]. Confusion matrix table can determine the performance of the classification, accuracy, precision, recall. Accuracy is the number of proportions of correct predictions. The accuracy calculation formula can be seen from the equation below.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}
\]

Precision is the ratio of TP / (TP + FP). Intuitive precision is the ability of classifiers to not label positive samples as negative. Recall is the ratio of TP / (TP + FN). recall intuitively is the ability of the classifier to find all positive samples.

| Class     | Positive                      | Negative                      |
|-----------|-------------------------------|-------------------------------|
| Positive  | True Positive (TP)            | False Negative (FN)           |
| Negative  | False Positive (FP)           | True Negative (TN)            |

3. Data and Methods
Data is obtained from the National Library of Medicine, Maryland, USA in collaboration with Shenzhen No.3 People’s Hospital, Guangdong Medical College, Shenzhen, China [4] [11]. The Chest X-rays are from out-patient clinics and were captured as part of the daily routine using Philips DR Digital Diagnose systems.

The CNN method used in this study goes through four stages to get the desired output. First, before data analysis needs to be done preprocessing data, which aims to see the characteristics of the data. Data characteristics are a general description of how a computer reads an image into an array which is used for further processing such as the spatial effect in the image data. Preprocessing also divides data into three data sets, training data, data validation, and test data aimed at modeling in the next step.

Second, data modeling, where the core part of CNN in data modeling will be repeated several times between the convolution layer and the pooling layer before ending with the fully connected layer. Third, After the training data modeling has been built, the next step is to determine the number of epochs. Adam optimization is used to improve the accuracy of the models that have been made. Adam is another level of adaptive learning optimization algorithm.

Algorithm Adam

Require: Step size \( \varepsilon \), initial parameters \( \theta \), exponential decay for moment estimates \( \rho_1 \) and \( \rho_2 \), small constant \( \delta \), used for numerical stabilization

Initialize 1st and 2nd moment variables \( s = 0, \quad r = 0 \)

Initialize time step \( t = 0 \)

While stopping criterion not met do

Sample a minibatch of \( m \) example from the training set \( \{ x^{(1)}, ..., x^{(m)} \} \) with corresponding targets \( y^{(1)} \)
Compute gradient: \[ \hat{g} \leftarrow \frac{1}{m} \nabla_y \sum_i L(f(x^{(i)}; \theta), y^{(i)}) \]

\[ t \leftarrow t + 1 \]

Update biased first moment estimate: \[ s \leftarrow \rho_1 s + (1 - \rho_1) g \]

Update biased second moment estimate: \[ r \leftarrow \rho_2 r + (1 - \rho_2) g \odot g \]

Correct bias in first moment: \[ \hat{s} \leftarrow \frac{s}{1 - \rho_1^t} \]

Correct bias in second moment: \[ \hat{r} \leftarrow \frac{r}{1 - \rho_2^t} \]

Compute update: \[ \epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta} \]

Apply update: \[ \theta \leftarrow \theta + \Delta \theta \]

End while

The name "Adam" comes from the phrase "adaptive moments" [12]. Adam is a combination of RMSProp and momentum with some important differences. First, in Adam, momentum is combined directly as an estimate of first-order moments (with exponential weights) of gradients. Second, Adam incorporated bias corrections into the estimation of first-order moments (momentum) and second-order moments (not centralized) to explain the origin initialization. The output of the selection of the number of epochs optimization is increased accuracy and decreased loss function in the training data even though using the same CNN modeling. The final step is the classification of test data seen from the confusion matrix to see how well the model classifies data.

4. Results

4.1. Preprocessing

The first step in preprocessing data is data sharing, from the raw data obtained and then sampled using a probability sample that is a simple random sample. The data sharing stage itself includes the separation of data into training data, validation data, and test data with each amount of data shown in Table 2.

| Part         | Amount of data | Condition                  | Amount |
|--------------|----------------|----------------------------|--------|
| Training data| 534            | Normal                     | 263    |
|              |                | Pulmonary tuberculosis     | 271    |
| Validation data| 67            | Normal                     | 33     |
|              |                | Pulmonary tuberculosis     | 34     |
| Test data    | 61             | Normal                     | 30     |
|              |                | Pulmonary tuberculosis     | 31     |
|              |                | Total                      | 663    |
The next preprocessing stage is the conversion of images into arrays, this is because the computer can only read an image as a number arrangement. Image conversion will also divide each image into three parts of color or depth, namely red, green, blue (RGB). Figure 4 is a sample of the conversion of an image into an array, where the array shown is 5x5 the first arrangement of numbers for each color section. This arrayed data will be processed on CNN, when the array is plotted back it will become a separate image based on color sections.

![Figure 4. Image conversion sample](image)

### 4.2. CNN modeling

The construction of the CNN model consists of several stages, which are repetitions between the convolution layer and the pooling layer, which ends with the fully-connected layer.

| Layer       | Output Shape | Param |
|-------------|--------------|-------|
| Convolution layer | (222,222,64) | 1792  |
| Max pooling | (111,111,64) | 0     |
| Convolution layer | (109,109,64) | 36928 |
| Max pooling | (54,54,64)   | 0     |
| Convolution layer | (52,52,128)  | 73856 |
| Max pooling | (26,26,128)  | 0     |
| Convolution layer | (24,24,128)  | 147584|
| Max pooling | (12,12,128)  | 0     |
| Flatten     | (18432)      | 0     |
| Dropout     | (18432)      | 0     |
| Dense       | 512          | 9437696|
| Total params |               | 9,698,369|
| Trainable params |           | 9,698,369|
| Non-trainable params |     | 0     |
Table 3 explains that the model was built from four iterations of the convolution and pooling layers before ending with the fully connected layer. Based on the above output, total parameters 9,698,269 are obtained, parameters trained are 9,698,369 and there are no untrained parameters.

4.3. Compare the best model

After getting the model the next stage is the application of the model formed into the training data and validation data by selecting several iterations or epochs. Each epoch goes through 16 steps.

| Information          | Max epochs |
|----------------------|------------|
|                      | 25  | 50  | 75  | 100 |
| Training accuracy    | 94.34% | 97.21% | 99.19% | 99.40% |
| Training loss function | 0.2517 | 0.1977 | 0.1043 | 0.1013 |
| Validation accuracy  | 80.60% | 79.10% | 80.60% | 74.63% |
| Validation loss function | 0.6261 | 0.7041 | 0.8342 | 0.9140 |

Application of the model in the training data and validation data is optimized using adaptive momentum or Adam. Adam was chosen because it is a refinement of optimization in pre-existing deep learning. Optimization function is to improve the learning rate which has an impact on increasing accuracy.

Table 4 displays the output of each epoch selection. The highest accuracy is 99.40% and the lowest loss function is 0.1013 in the training data and the highest accuracy is 80.60% and the lowest loss function is 0.6261 in the validation data, then the best model chosen uses 75 epochs because the addition of epochs to 100 only adds 0.21% to the training data and actually reduces the accuracy of the validation data, while for the highest validation at 25 epochs and 75 epochs have different levels of accuracy with a difference of 4.85% which will affect against test data.
Figure 5 explains that the accuracy of the model tends to increase and the loss function tends to decrease for training data, while for validation data, the accuracy of the model tends to decrease and the loss function tends to increase. Based on the validation data plot there are fluctuations in each increase epochs.

4.4. Classification
The model that has been obtained then implemented into the test data

![Confusion Matrix](image)

*Figure 6. Confusion Matrix*

Figure 6 explains that the model that has been built can correctly classify normal lungs as much as 25 data and there are 5 errors, whereas for lungs with pulmonary tuberculosis conditions, the model is able to classify 26 data correctly and 5 data are incorrect. Classification of the test data can be seen more clearly from the report on the classification results shown in Table 5.

**Table 5. Classification reports**

| Condition                  | Precision | Recall |
|----------------------------|-----------|--------|
| Normal                     | 0.83      | 0.83   |
| Pulmonary tuberculosis     | 0.84      | 0.84   |

Accuracy **0.84**

Table 5 explains the precision of lung pulmonary tuberculosis by 0.84 which means that the ability of the classifier to not label pulmonary tuberculosis as a normal sample is 84%, while for normal is 83%. Recall of pulmonary in the lungs is 0.84 which means that the ability of the classifier to find all pneumonia samples is 84%, and for normal lungs 83%, the overall accuracy rate is 84%.

5. Conclusion
The best model to classify pulmonary tuberculosis disease using convolutional neural network is built using 75 epochs which accuracy results in the training and validation data is 99.19% and 80.60% respectively. The model then implemented into test data with an accuracy rate of 84%. Normal lungs being correctly classified as 25 data and 5 data are incorrect, for lungs with pulmonary tuberculosis, the model was able to classify 26 data correctly and 5 data are incorrect.

6. Acknowledgement
Authors would like to thank Univeristas Sebelas Maret Surakarta for supporting this research through Grant of Improvement of Research Laboratory Capacity 2019.
References

[1] WHO, “Tuberculosis,” World Health Organization, 2018.
[2] A. Rizal, R. Hidayat and H. Nugroho, "Entropy measurement as features extraction in automatic lung sound classification," in International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2017.
[3] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri and R. M. Summers, "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017.
[4] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Z. Xue, K. Palaniappan, R. Singh, S. Antani, G. Thoma, Y. Wang, P. Lu and C. McDonald, "Automatic tuberculosis screening using chest radiographs," IEEE Trans Med Imaging, vol. 33, no. 2, pp. 233-45, 2014.
[5] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Massachusetts: MIT Press, 2016.
[6] F.-F. Li, J. Johnson dan S. Yeung, Class Lecture, Topic: “CS231n Convolutional Neural Networks for Visual Recognition,” College of Computer Science, Stanford University, Stanford, 2019.
[7] D. Scherer, A. Muller and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in Proc. 20th International Conference on Artificial, Berlin, 2010.
[8] A. Saravia, N. M. Fonseca, L. Sousa, N. J. Costa, J. V. Sousa, D. B. Santos, S. Soares and A. Valente, "Classification of Images of Childhood Pneumonia using Convolutional Neural Networks," in 12th International Joint Conference on Biomedical Engineering Systems and Technologies, Prague, 2019.
[9] A. Deshpande, “A Beginner's Guide To Understanding Convolutional Neural Networks,” 2019.
[10] F. Pedregosa, A. Gramfort, V. Michel, B. Thirion and O. Grisel, "Scikit-learn: Machine learning in Python," Journal of machine learning research, p. 2825–2830, 2011.
[11] S. Candemir, S. Jaeger, K. Palaniappan, J. Musco, R. Singh, Z. Xue, A. Karargyris, S. Antani, G. Thoma and C. McDonald, "Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration," IEEE Trans Med Imaging, vol. 33, no. 2, pp. 577-90, 2014.
[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization.," CoRR, 2014.