Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2025, 2025:10
https://doi.org/10.28919/cmbn/9021
ISSN: 2052-2541

PERFORMANCE CONVOLUTIONAL NEURAL NETWORK (CNN) AND SUPPORT VECTOR MACHINE (SVM) ON TUBERCULOSIS DISEASE CLASSIFICATION BASED ON X-RAY IMAGE

ARIEF RACHMAN HAKIM*, BUDI WARSITO, MUHAMMAD RIONANDO DIVANGGA, DIAH SAFITRI, ARDIANA ALIFATUS SA’ADAH

Department of Statistics, Diponegoro University, Semarang 50275, Indonesia

Abstract: Tuberculosis (TB) problem continues to be a significant health issue worldwide, with a specific emphasis on developing countries. Early diagnosis of TB can aid effective treatment and prevent disease transmission. Image-based X-ray classification of tuberculosis has become an interesting research topic in the development of automatic diagnostic systems that can assist doctors in making better decisions. In this research, the performance of the Convolutional Neural Network and Support Vector Machine in the X-ray image-based classification of tuberculosis was compared. The data used consisted of 1400 X-ray images from tuberculosis and normal patients, comprising 700 tuberculosis images and 700 non- tuberculosis images. The results showed that CNN outperformed SVM in the classification of tuberculosis based on X-ray of thorax. CNN achieved an accuracy of 97.86%, while SVM only reached 96.07%. Additionally, CNN also had higher recall and precision values than SVM, indicating that CNN is more suitable for use in X-ray image-based TB classification. In conclusion, this study demonstrated that CNN is superior to SVM in X-ray image-based TB classification. However, further research is needed to enhance the

---

*Corresponding author
E-mail address: arief.rachman@live.undip.ac.id

Received November 18, 2024

performance of CNN and SVM algorithms in X-ray image-based TB classification using more sophisticated techniques.

Keywords: tuberculosis; image classification; CNN; SVM.

2020 AMS Subject Classification: 62M45, 68T07, 92B20.

1. INTRODUCTION

Tuberculosis (TB) is an infectious disease that remains a major global health challenge today, especially in developing countries. Early diagnosis of TB is essential to avoid the spread of the disease and increase the patient's chance of recovery. However, TB diagnosis still faces challenges, especially in developing countries where medical personnel are scarce. One way to maximize TB diagnosis is by using x-ray image-based classification technology [1].

X-ray image-based TB disease classification has become an interesting research topic in the development of automated diagnosis systems [2]. In this research, Convolutional Neural Network and Support Vector Machine algorithms are used to compare their performance in x-ray image-based TB disease classification. The data used are 1400 x-ray images of TB and non-TB patients. This study aims to identify the best algorithm for TB disease classification based on X-ray images. Artificial intelligence advances are gradually changing medical practice and will be increasingly used in the more complex and rapidly evolving field of health care [3]. It is expected that the results of this study can help to improve the accuracy and efficiency of TB diagnosis and provide support for doctors in making better decisions.

2. PRELIMINARIES

Tuberculosis, commonly referred to as TB, is a disease condition caused by the bacteria mycobacterium tuberculosis. The common way for these bacteria to infect the human body is through the process of inhaling air which then enters the lungs. After entering the lungs, the bacteria may spread to other parts of the body. The thorax X-ray is the main tool with high sensitivity in diagnosing pulmonary TB, which shows infiltrating patches or nodules in the patient

[4]. Fig 1 are some example the thorax X-ray image for TB.

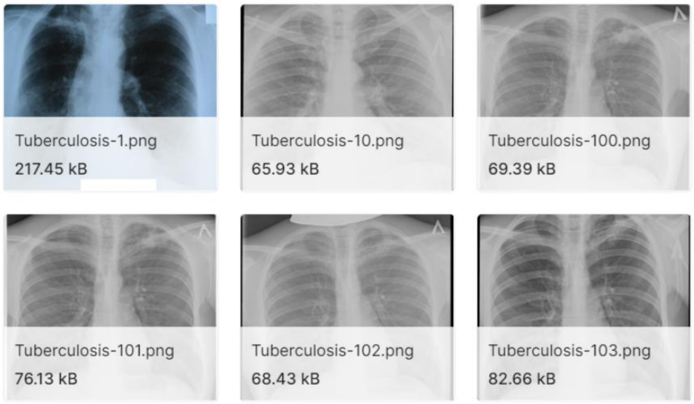
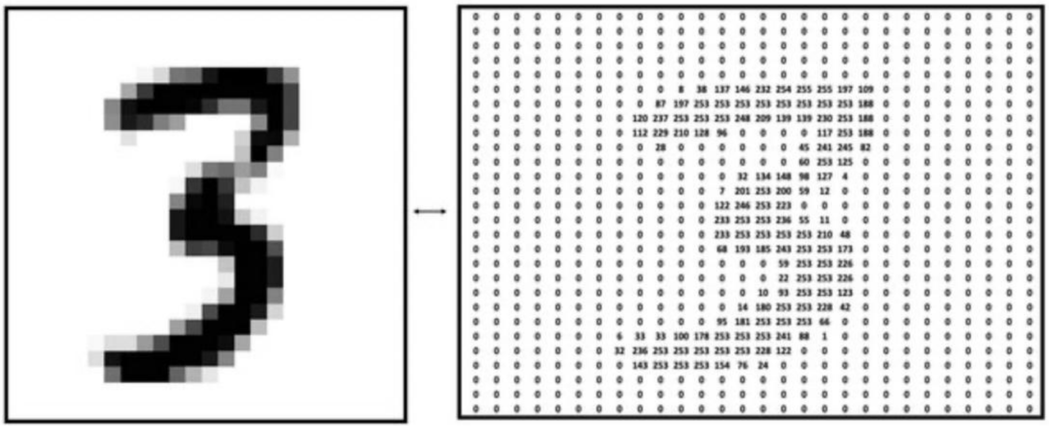


Figure 1. Example the thorax X-ray image for TB

2.1. Digital images

Digital images are obtained as a result of image processing by machines using several different methods based on sampling and quantization techniques, resulting in an accurate representation of the image. The sampling is represented in the form of squares arranged in rows and columns. In other words, image sampling refers to the size of pixels in the image, and quantization refers to the brightness level value represented by the grayscale level determined by the number of binary bits used by the machine [5]



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	8	38	137	146	232	254	255	255	197	109	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	87	197	253	253	253	253	253	253	253	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	120	237	253	253	253	248	209	139	139	230	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	112	229	210	128	96	0	0	0	0	117	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	45	241	245	82	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	253	125	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	32	134	148	98	127	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	7	201	253	200	59	12	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	122	246	253	223	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	233	253	253	236	55	11	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	233	253	253	253	253	210	48	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	68	193	185	243	253	253	173	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	253	253	226	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	253	253	226	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	93	253	253	123	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	14	180	253	253	228	42	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	181	253	253	253	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	6	33	33	100	178	253	253	253	241	88	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	32	236	253	253	253	253	253	228	122	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	143	253	253	253	154	76	24	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2. *Image Representation*

There are 3 types of digital images, binary image is one type of digital image that only consists of two possible colors, black and white, which are represented by binary values, which are 0 or 1. Then grayscale image is a digital image whose image type consists of gray, ranging from black to the part with the lowest intensity and the white part with the highest intensity. RGB image is a type of digital image that consists of three color channels, which are red, green, and blue [5]. The steps in the process of converting image data into numeric is that each pixel in the image is converted into a numeric number that represents the color intensity of that pixel. In grayscale images, each pixel has an intensity value that ranges from 0 to 255, while in color images (RGB), each pixel consists of three intensity values of red, green, and blue which also range from 0 to 255. [5]. Then, normalization is carried out on the data which initially has a range of 0-255 to 0-1 with the following equation 1.

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \tag{1}$$

2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a neural network algorithm that is capable of processing input images and determining the weights and biases of various features or objects present in the image, thus enabling the recognition of differences between objects in the image.

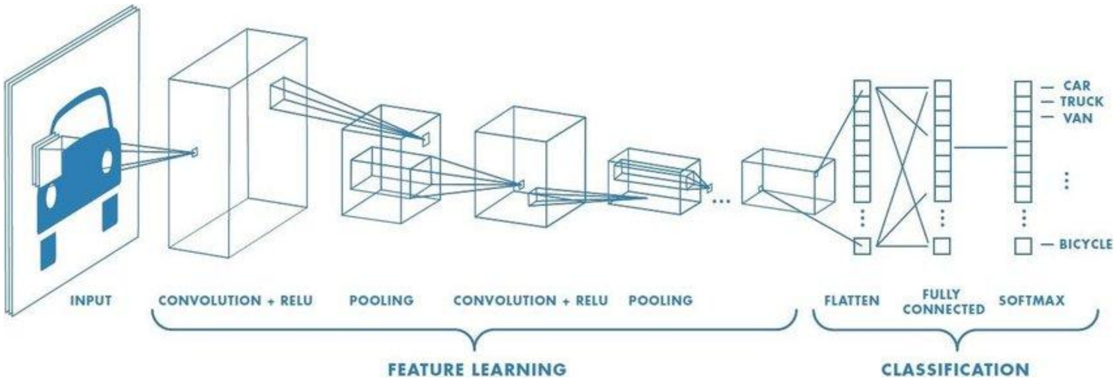


Figure 3. CNN Architecture

The architecture of a CNN is based on the structure of the visual cortex of the human brain. CNN architecture consists of several layers that aim to perform feature learning and classification on images [6]. The following are the layers in CNN:

a. Convolutional Layer

Convolutional Layer performs a linear function that involves the multiplication of a set of weights with the input data, similar to conventional neural networks. When dealing with two-dimensional inputs, this operation involves the multiplication of an input array with a two-dimensional filter or kernel containing weighted values [7].

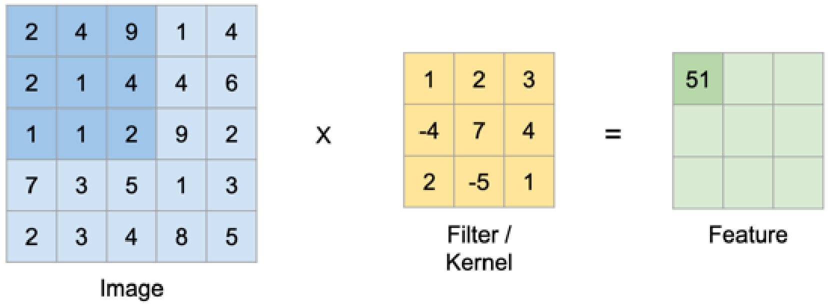


Figure 4. How Convolutional Layer Works

The convolution operation on the convolutional layer can be written as follows:

$$\boldsymbol{S_q(g,h) = \left(\sum_{u=1}^{U}\sum_{v=1}^{U} I_p\,(g+u,h+v).K_{pq}(u,v)\right) + b_q} \tag{2}$$

$\boldsymbol{S_q(g,h)}$ is the q feature map element, in the $\boldsymbol{g}$ row of the $\boldsymbol{h}$ column. $\boldsymbol{I_p(g+u,h+v)}$ is the $\boldsymbol{p}$ input channel matrix element, in the $\boldsymbol{m+u}$ row, $\boldsymbol{h+v}$ column**.** $\boldsymbol{K_{pq}(u,v))}$ is the pq kernel matrix element, in the $\boldsymbol{u}$ row, $\boldsymbol{v}$ column. bq is the bias of the q filter. The number of parameters in the convolutional layer can be written as follows:

$$\boldsymbol{Number\ of\ Parameter = ((U \times U \times P) + 1) \times Q} \tag{3}$$

U is the number of rows/columns in the kernel matrix. P is the number of kernel matrices in a filter and Q is the number of filters in a layer.

b. Pooling Layer

The Pooling Layer is responsible for reducing the size of the feature map, with the objective of decreasing the dimensionality of the output feature map from the previous layer. There are two types of pooling layers commonly used, first is max pooling that extracts the maximum value of a region with a specified size from the feature map, and average pooling, that computes the average value of all the values within a certain region of the feature map with a specified size [7]. The output shape of the pooling layer can be calculated as follows:

$$\boldsymbol{Output\ shape} = \frac{\boldsymbol{input}}{\boldsymbol{U}} \tag{4}$$

U refers to the number of rows or columns in the filter matrix. If the dimension of the feature maps is a multiple of U, then the input value is equal to the number of rows/columns in the feature maps. If the dimension of the feature maps is not a multiple of U, then the input value is equal to the dimension of the feature maps minus the part that cannot be reached by the filter.

c. Flattening Layer

Flattening in CNN is a step performed after the convolution and pooling stages in the first layer of CNN. Flattening is done with the aim of changing the output of the preceding layer into a one-dimensional vector, which can then be processed by the fully connected layer in CNN. The following is a calculation to find the output shape formula of the flatten layer.

$$\boldsymbol{Output\ shape} = (\boldsymbol{S} \times \boldsymbol{S}) \times \boldsymbol{P} \tag{5}$$

The flatten layer output shape is a column vector with multiple rows, where S represents the number of rows/columns in the input matrix and P denotes the input matrix.

d. Fully Connected Layer

The output of each neuron in the fully connected layer is transmitted to every neuron in the subsequent layer, while receiving input from all neurons in the previous layer. The output of each neuron in the fully connected layer is determined by computing the product of each input weight and adding a bias term [8]. The calculation is as follows:

$$\boldsymbol{z}_{\mathbf{r}} = \sum_{\boldsymbol{c}=\mathbf{1}}^{\boldsymbol{J}} \boldsymbol{x}_{\boldsymbol{c}} \boldsymbol{w}_{\boldsymbol{cr}} + \boldsymbol{b}_{\boldsymbol{r}} \tag{6}$$

J is the number of input neurons, z_r is the output of the *r* neuron, x_c is the input of the *c* neuron, w_{cr} is the weight of the *c* input neuron to the *r* output neuron and b_r is the bias of the *r* output neuron [9]. The number of parameters in the fully connected layer can be calculated as follows:

$$\boldsymbol{Number\ of\ Parameter} = (\boldsymbol{J} + \mathbf{1}) \times \boldsymbol{R} \tag{6}$$

The variable J represents the number of inputs from the previous layer to the current layer, while R denotes the number of neurons in the current layer.

2.3. Support Vector Machine

Support Vector Machine (SVM) is a popular supervised learning technique utilized for classification purposes. The SVM algorithm is used to identify the optimal hyperplane by maximizing the margin between the classes. [10]. The SVM method aims to find the optimal hyperplane to separate the two classes by maximizing the margin, which is the distance between the hyperplane and the nearest data points from each class. These data points are referred to as support vectors [11].

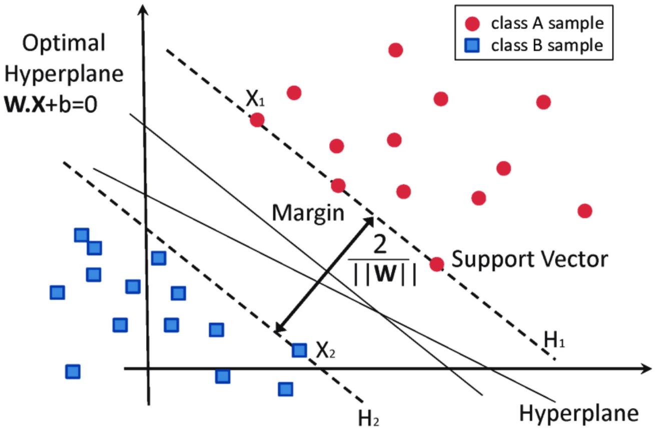


Figure 5. How SVM Works

Linearly separable data is a type of data that can be linearly separated. For example, training data is denoted as and class label as for , where is the sum of the data. This method aims to find the best separating function (hyperplane) from an infinite number of functions to separate two types of objects. The general linear model used in SVM to find the hyperplane is as follows equation 8 [10]:

$$y_i = \mathbf{W}^T\mathbf{x}_i + b, \qquad i = 1, 2, .., n \tag{8}$$

W : weight parameter vector

x_i : vector of independent variables

B : bias

$y_i \in \{-1, +1\}$: target value of the x data set

n : amount of data

It is assumed that the positive and negative classes can be ideally separated by a hyperplane

$\mathbf{W}^T\mathbf{x} + b = 0$[10]. The margin between the positive class and the negative class can be calculated by finding the distance between the supporting hyperplane $\mathbf{W}^T\mathbf{x}_1 + b = +1$ that supports the positive class and the supporting hyperplane $\mathbf{W}^T\mathbf{x}_2 + b = -1$ that supports the negative class. The way to find the best hyperplane is through the use of Quadratic Programming (QP) problem. Quadratic Programming (QP) is one of the optimization equations commonly used to solve this problem. In the following way:

$$\boldsymbol{min}\frac{\mathbf{1}}{\mathbf{2}}||\mathbf{W}||^{\mathbf{2}} \tag{9}$$

Subject to $\boldsymbol{y_i(\mathbf{W}^T\mathbf{x}_i + b) \geq 1, i = 1, 2, .., n}$. Optimization can be solved with the Lagrange Multiplier technique, which gives one multiplier for each constraint, resulting in a Lagrange function in equation 10:

$$\boldsymbol{L_p(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{W}||^2 - \sum_{i=1}^{n} a_i\,(y_i(\mathbf{W}^T\mathbf{x}_i + b) - 1)} \tag{10}$$

In the equation, the optimal value can be achieved by minimizing L with respect to the values of w and b, and maximizing L with respect to αi. Optimization can be solved with the lagrangian multiplier function, which gives one A multiplier for each constraint, then produces the Lagrange function:

$$\boldsymbol{max_a L_D = max\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left(x_i^T x_j\right)} \tag{11}$$

Subject to $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i$. Generally, the two classes in the input space cannot be ideally separated, causing the constraints to not be met and resulting in optimization not being possible. However, to overcome this problem, SVM can use the soft margin technique, where equation (8) is modified by introducing slack variables.

$$min\,(\tau, \xi) = min\left(\frac{1}{2}||\boldsymbol{w}||^2 + c\sum_{i=1}^{n}\xi_i\right) \tag{12}$$

Subject to $\sum_{i=1}^{n} a_i y_i = 0, a_i \geq 0(\, i = 1,2, .., n)$

The role of the C parameter is to set the optimization level between the margin and the classification error ξ. Therefore, the higher the value of C used, the more significant the effect on the classification error. Real-life data problems usually do not have linear properties and are more often non-linear. SVM uses a kernel function to handle such non-linear problems. Kernel functions allow the use of models in feature spaces that have larger dimensions [11]. The following equation is Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i, x_j \right\|^2\right) \tag{13}$$

The following equation is used to obtain the classification result

$$f(\phi(x)) = sign\left(\sum_{i=1}^{ns} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b\right) \tag{14}$$

$$f(\phi(x)) = \begin{cases} 1,\ jika\ \sum_{i=1}^{ns} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b \geq 0 \\ -1,\ jika\ \sum_{i=1}^{ns} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b < 0 \end{cases} \tag{15}$$

$f(\phi(x))$	: classification result of x data
y_i	: data class
α_i	: lagrange coefficient
$K(\boldsymbol{x}_i, \boldsymbol{x}_j)$	: kernel function of test data and training data
b	: bias
ns	: number of support vectors

2.4. Data and Evaluation System

The data in this study were 1400 X-ray images of the lungs, the dataset used consisted of 700 positive cases of tuberculosis and 700 images of normal lungs. Data was collected from 3 different sources, first being the National Library of Medicine in the US [14]. The second is the RSNA pneumonia detection challenge dataset [15]. Third is Belarus Set [16] by the National Institute of Allergy and Infectious Diseases, Ministry of Health Republic of Belarus. The following are the stages in the data analysis process:

a. Input and load data
b. Perform Data Pre-processing
c. Classify x-ray image with CNN algorithm

d. Classifying x-ray images with SVM algorithm

e. Comparing the Classification Performance of CNN and SVM methods

A confusion matrix is a tabular representation used to evaluate the performance of a machine learning model on a given dataset. Confusion matrix describes how well the model can classify the data into the correct categories. The confusion matrix displays the count of accurate and inaccurate forecasts for each classification category. Using these values, the effectiveness of a classification model can be assessed [12].

Table 1 *Confusion Matrix table*

Confusion Matrix		**Prediction Value**	
		Positive	**Negative**
Actual Value	**Positive**	TP	FN
	Negative	FP	TN

There are several metrics used to evaluate classification models, such as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

3. MAIN RESULTS

The pre-processing process in tuberculosis classification using Support Vector Machine (SVM) involves several steps as follows:

a. **Data loading:** The first step is to load the chest radiograph images that have been collected and stored in the dataset. This dataset consists of chest radiograph images that are infected and not infected with tuberculosis.

b. **Convert to grayscale:** This is done by taking the average of the red, green, and blue color channels for each pixel. Grayscale images only have one color channel and are easier to work with in many machine learning applications.

c. **Image resizing:** Since the images have different sizes, the next step is to resize the images to the same size 224x224.

d. **Convert the image to a numerical array:** Most machine learning models require input images to have a fixed size, so this step is important. This is done by assigning a numeric value to each pixel in the image. For grayscale images, each pixel is assigned a single value between 0 and 255 that represents the brightness of the pixel.

e. **Data normalization:** After convert image dataset into array, the next step is to normalize the data by changing the pixel values from a scale of 0-255 to 0-1. This is done to speed up the training time and improve the model's accuracy.

f. **Data splitting:** After the radiograph images are processed, the next step is to divide the dataset into two parts: training data and testing data with the ratio is 80:20.

After the pre-processing process is completed, the CNN and SVM model can be tested using the processed and normalized testing data. Proper pre-processing can improve the accuracy and performance of the model in classifying infected and not infected chest radiograph images with tuberculosis.

Table 2 *Result of Number of Convolutions Trial*

Total Convolution	*Accuracy Validation*
1	96.07%
2	94.64%
3	97.50%
4	97.86%
5	97.14%

Table 3 *Pooling Layer Type Trial Results*

Pooling Layer Type	*Accuracy Validation*
Max Pooling	97.86%
Average Pooling	94.29%

Table 4 *Kernel Size Trial Result*

Kernel Size	*Accuracy Validation*
3x3	97.86%
5x5	96.43%
7x7	96.07%

Table 5 *Learning Rate Trial Result*

Learning Rate	*Accuracy Validation*
0.1	50.00%
0.01	50.00%
0.001	97.86%
0.0001	95.71%

Based on the results of the tests that have been carried out, it is written in tables 2, 3, 4 and 5. The model to be used has a number of convolutions of 4 sized 3x3 kernels using max pooling with a learning rate of 0.001. The following table 6 is a summary of the architecture used and the calculation of its parameters.

Table 6 *CNN Best Model Architecture*

No	*Layer*	*Size*	*Parameter*
1	*Input layer*	(224, 224, 1)	0
2	*Convolution layer 1*	(222, 222, 32)	320
3	*Pooling layer 1*	(111, 111, 32)	0
4	*Dropout 0.25*	(111, 111, 32)	0
5	*Convolution layer 2*	(109, 109, 64)	18496
6	*Pooling layer 2*	(54, 54, 64)	0
7	*Dropout 0.25*	(54, 54, 64)	0
8	*Convolution layer 3*	(52, 52, 128)	73856
9	*Pooling layer 3*	(26, 26, 128)	0
10	*Dropout 0.25*	(26, 26, 128)	0
11	*Convolution layer 4*	(24, 24, 256)	295168
12	*Pooling layer 4*	(12,12,256)	0
13	*Dropout 0.25*	(12,12,256)	0
14	*Flatten Layer*	(43264)	0
15	*Hidden Layer*	(128)	4718720
16	*Dropout 0.5*	(128)	0
17	*Output*	(2)	258
	Total		5106818

From the modeling results that have been done based on the architecture in table 7. Here are the results of the CNN confusion matrix.

Table 7 CNN *Confusion Matrix*

Prediction Value	**Actual Value**	
	Positive	**Negative**
Positive	136	4
Negative	2	138

With result $Accuracy = 0.9786,\ Recall = 0.9714,\ Precision = 0.9855.$

Each epoch experienced an increase in accuracy and by using the number of epochs 100. To see whether the model made overfitting or underfitting can be seen from the following graph.

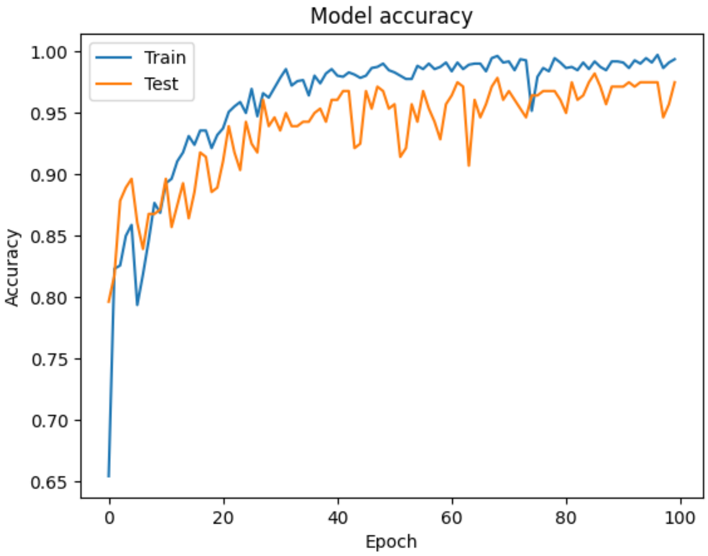


Figure 6. CNN Model Accuracy Graph

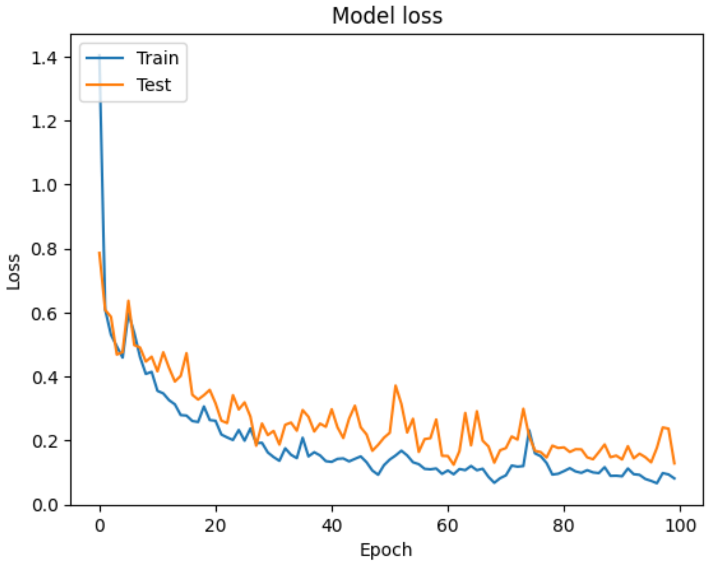


Figure 7. CNN Model Loss Graph

Figure 6 and 7 shows the results of the accuracy of validation data and training data as well as the loss of validation data and training data with the number of epochs 100. In the accuracy model graph, the graph shows a continuous increase in the accuracy of validation and training at each epoch, and at epoch 100, the accuracy reaches 97.86%. While on the loss model graph, the graph shows a continuous decrease in validation and training loss at each epoch, and at epoch 100, the resulting loss value is 0.1287, indicating good performance of the model used.

In this research using Support Vector Machine algorithm utilizing the RBF kernel function. Determination of classification with SVM kernel RBF method depends on the gamma and C parameters. Gamma is a parameter in the RBF kernel that determines how far the influence of training data samples in classification. C is the SVM parameter chosen to control the error. The value of the C parameter is found in the optimization of the Lagrange function. Optimization of the C parameter is done by conducting trials. The following Table 8 are the results of the trial to find the best C parameter.

Table 8 *Resul of The Cost Trial*

Evaluation Metrics	**Cost (C)**				
	0,01	**0,1**	**1**	**10**	**100**
Accuracy	0.8178	0.8964	0,9571	0.9571	0.9607

$$\gamma = \frac{1}{ncol} + \frac{1}{50176} = -0.0000199298469$$

With ncol being the number of data dimensions. If in research, ncol is the number of columns in the image data that has been converted to numeric data according to the image resolution, which is 224 × 224.

In SVM, the determination of support vector points depends on the result of dot multiplication between data that has been transformed into a higher dimensional space. To perform this dot multiplication, the RBF kernel function is used on each data. In the kernel method, data is not represented separately, but rather through a comparison between a pair of data. From the modeling results that have been done, here are the results of the SVM confusion matrix.

Table 9 SVM *Confusion Matrix*

Prediction Value	Actual Value	
	Positive	**Negative**
Positive	134	6
Negative	5	135

With result $Accuracy = 0.9607$, $Recall = 0.9571$, $Precision = 0.9640$.

To find the best algorithm model for tuberculosis disease classification, it is necessary to compare the model evaluation between the CNN algorithm and the SVM algorithm, the following Table 10 are comparison of the evaluation matrix.

Table 10 SVM *Confusion Matrix*

Model Evaluation	Method	
	CNN	**SVM**
Accuracy	0.9786	0.9607
Recall	0.9714	0.9571
Precision	0.9855	0.9640

From the two algorithms that have been tested, the convolutional neural network algorithm has the best evaluation matrix value, namely with an accuracy of 97.86%. In addition, CNN also has higher recall and Precision values compared to SVM. The performance of the Convolutional Neural Network algorithm outperforms the support vector machine in classifying tuberculosis disease. The results show that CNN has better evaluation metrics results and more suitable than SVM in the classification of TB disease based on x-ray.

ACKNOWLEDGMENT

Special thanks to Faculty of Science and Mathematics, Diponegoro University, Indonesia for support and funded this Research through the Main Research Scheme 2023.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] WHO, Tuberculosis, World Health Organization, 27 Oct. 2022. https://www.who.int/news-room/fact-sheets/detail/tuberculosis.

[2] WHO, Systematic Screening for Active Tuberculosis: Principles and Recommendations, World Health Organization, 20 Sept. 2016. https://iris.who.int/bitstream/handle/10665/84971/9789241548601_eng.pdf

[3] K.-H. Yu, A.L. Beam, I.S. Kohane, Artificial Intelligence in Healthcare, Nat. Biomed. Eng. 2 (2018), 719–731. https://doi.org/10.1038/s41551-018-0305-z.

[4] M.X. Rangaka, S.C. Cavalcante, B.J. Marais, et al. Controlling the Seedbeds of Tuberculosis: Diagnosis and Treatment of Tuberculosis Infection, The Lancet 386 (2015), 2344–2353. https://doi.org/10.1016/S0140-6736(15)00323-2.

[5] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Pearson, New York, 2018.

[6] K. O'Shea, R. Nash, An Introduction to Convolutional Neural Networks, arXiv:1511.08458 [cs.NE] (2015). https://doi.org/10.48550/arXiv.1511.08458.

[7] S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a Convolutional Neural Network, in: 2017 International Conference on Engineering and Technology (ICET), IEEE, Antalya, 2017: pp. 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186.

[8] M.A. Nielsen, Neural Networks and Deep Learning, Determination Press, (2015).

[9] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, IEEE Trans. Neural Netw. Learn. Syst. 33 (2022), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827.

[10] L. Wang, Support Vector Machines: Theory and Applications, Springer, New York, (2005).

[11] W.S. Noble, What Is a Support Vector Machine?, Nat. Biotechnol. 24 (2006), 1565–1567. https://doi.org/10.1038/nbt1206-1565.

[12] J. Liang, Confusion Matrix, POGIL Act. Clear. 3 (2022), 4.

[13] T. Rahman, A. Khandakar, M.A. Kadir, et al. Reliable Tuberculosis Detection Using Chest X-Ray with Deep Learning, Segmentation and Visualization, IEEE Access 8 (2020), 191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384.

[14] Pandey. Chest Xray Masks and Labels. Accessed: Nov. 10, 2023. https://www.kaggle.com/datasets/nikhilpandey360/chest-xray-masks-and-labels.

[15] B.P. Health, Belarus Tuberculosis Portal, Accessed: Nov. 10, 2023. (2020).

[16] kaggle, RSNA Pneumonia Detection Challenge. Accessed: Nov. 10, 2023. https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge.