LUNG CANCER DETECTION USING A MODIFIED CONVOLUTIONAL NEURAL NETWORK (CNN)

Cari Cari, Mohtar Yunianto*, Aisyah Ajibah Rahmah
Physics Department, Universitas Sebelas Maret Surakarta, Indonesia
* mohtaryunianto@staff.uns.ac.id

Received 22-07-2023, Revised 20-10-2023, Accepted 31-10-2023,
Available Online 01-04-2024, Published Regularly April 2024

ABSTRACT

Image processing is used to classify lung images with malignant or normal nodules. The Convolutional Neural Network (CNN) method is often used to classify images. This study uses a modified CNN architecture with various layers, filters, batch size, dropout, and epoch values. Variations were made to determine the best accuracy value and reduce the overfitting value of the proposed CNN architecture. This study implements the Keras library method with the Python programming language. The data is in the form of CT-Scan images of lung cancer and normal lungs. The results of several experiments from the proposed model produce an accuracy value of 95% using three layers, 128 filters on the first layer, 256 on the second layer, and 512 filters on the third layer, then with 32 batch sizes, 0.5 dropout.

Keywords: lung cancer; batch size; number of filters; number of layers; convolutional neural network

Cite this as: Cari., Yunianto, M., & Rahmah, A. A. 2024. Lung Cancer Detection Using a Modified Convolutional Neural Nerwork. *IJAP: Indonesian Journal of Applied Physics, 14*(1), 52-62. doi: https://doi.org/10.13057/ijap.v14i1.77032

INTRODUCTION

Cancer is a genetic disease caused by the formation of tumors by damaged tissue that develops into cancer. Cancer cells can interfere with other tissues by growing directly in adjacent tissues or moving to distant locations [1]. The exact cause of lung cancer is unknown, but prolonged exposure to carcinogenic substances is one of the main causes of lung cancer [2]. Long-term exposure to air pollution from factories, cars, food fumes, and formaldehyde from indoor decorations can increase the risk of lung cancer [3]. Based on data from the 2018 Global Cancer Observatory (GCO), of all people living with cancer worldwide, around 11.6% have lung cancer. It is recorded that lung cancer causes death in 18.4% of the total data of cancer patients who died. Based on data from the Cancer Country Profile (CCP), in 2018, there was 8.6% lung cancer in Indonesia, with a mortality rate of up to 12.6%.

Image results from medical equipment such as X-rays, ultrasound (ultrasound), CT-scan (Computer Tomography scanner), and MRI (Magnetic resonance imaging) can be used to diagnose diseases such as cancer [4]. A computer-aided diagnosis (CAD) system has been developed to help radiologists make a final decision by considering the system's results along with the clinical diagnosis [5]. The CAD learning method creates a training model for medical images to process data in a computer-assisted structure [6]. From an analytical point of view, layered images in MRI and CT display more detail and produce images with higher ratio levels, making them easier to process [7].

In establishing the diagnosis of lung cancer, the identification of lung conditions is important. This identification can be done by utilizing the design of artificial neural network software. The design of artificial neural networks will help classify medical images in identifying normal lungs, effusions, and lung cancer [8]. Convolutional Neural Network (CNN) is a Deep Learning method for classification, segmentation, and image analysis. The CNN architecture has several layers: a convolution layer with stride size and zero padding, a max pooling layer, and a fully connected layer [9]. CNNs are suitable for analysing X-ray images because they easily train networks. In addition, CNN does not require any manual feature engineering. This network can train raw images and classification labels [7].

MATERIALS AND METHODS

Convolutional Neural Networks

A convolutional neural network (CNN) is an advanced neural network. This network is usually a combination of several layers consisting of several filters. The pooling layer is used to perform linear sampling operations. CNN has a very good performance in image recognition [7]. The training process has three stages: the convolutional layer, the pooling layer, and the connection layer.

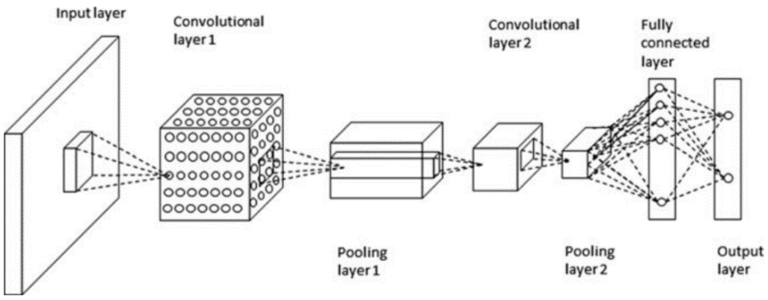


Figure 1. Basic Structure of Convolutional Neural Network [10]

Convolution Layers

The convolutional layer will receive image input and process it to produce output as a vector to be processed in the next layer. This layer obtains a pixel derived from the preferred and neighbouring pixel values. This layer applies the output function as a feature map of the input image [11]. The output value of the convolution layer is the product of the multiplication between the image input matrix and the matrix in the kernel. The equation is

$$H = I\,.K \tag{1}$$

Where H is the output feature map of the convolutional layer, I is the input image, and K is the kernel matrix [12]. The dimensions of the output matrix of the feature map are based on the size of the input image matrix, kernel matrix, stride, and padding. The shift size of a kernel is also known as a stride. If the stride value is 1, the convolution process shifts 1 pixel. If the stride value is 2, the convolution process shifts 2 pixels [9]. *Padding* is the value added to each side of the input matrix. The convolution process also includes input layers, filters, and convolution layers. Input volume dimensions, filter dimensions, and output volume dimensions are stated in [9] :

$$V_{input} = h \times w \times d \tag{2}$$

In the image matrix, V_{input} is the input volume dimension, h is the height, w is the width, and d is the dimension

$$f = f_h \times f_d \qquad (3)$$

In the filter matrix section, f is a filter, f_h is a height filter, and f_d is a dimension filter

$$V_{output} = (h - f_h + 1)x\ (f - f_w + 1) \times 1 \qquad (4)$$

In the convolution layer, V_{output} is the output volume dimension.

Rectified Linear Unit (ReLU)

Rectified Liner Unit (ReLU) is an activation function used in CNN. ReLU will change the feature map input value to 0 and infinity. The ReLU activation function is stated as follows[12]:

$$F_{relu} = max\ (0, x) \qquad (5)$$

When the activation value is 0, ReLU cannot be processed via the gradient march method because all gradients will be zero. Therefore, the leaky ReLU will be calculated using[6]:

$$F'(z) = \{1\ if\ z > 0\ 0\ if\ z\ \leq 0\,\}$$
(6)

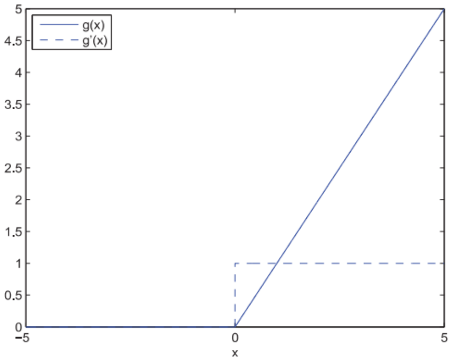
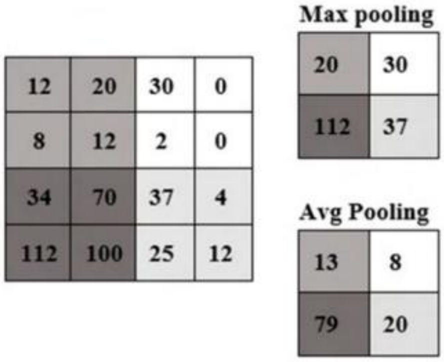


Figure 2. ReLU Function and Derivatives [13]

Pooling Layer

The pooling layer is a layer that uses a feature map as input and processes it statistically based on the closest pixel value. This layer can reduce the size of the output volume on the feature map, thereby reducing the number of light calculations, and can control overfitting [11]. The commonly used pooling layers are max pooling and average pooling. The results of max pooling are obtained by taking the highest value in each grid. In average pooling, it will take the average value [6]:



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

Max pooling

20	30
112	37

Avg Pooling

13	8
79	20

Figure 3. Pooling Layer Results [6]

Figure 4 shows that average pooling is obtained by calculating the average of each grid, while max pooling is obtained by taking the largest value from each grid in the image pixel. The size of the kernel dimensions used in the pooling layer is determined by the size of the stride (shift) and the size of the kernel dimensions as shown in equation [12]:

$$O = \frac{i-k}{s} + 1 \qquad (7)$$

Where O is the dimension of the pooling layer, i is the dimension of the input image matrix, k is the size of the kernel, and s is the number of strides. The average pooling value can be determined using equation [12]:

$$f_A = \frac{\Sigma \quad C_y}{|C_y|}$$

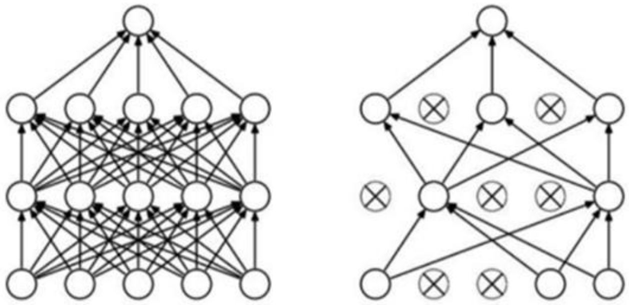
(8)

The max-pooling value can be determined using equation [12]:

$$f_M = max\ (C_y) \qquad (9)$$

Dropout Layers

The dropout layer is useful for randomly removing neurons with a probability of p in each training data iteration process, and p can be determined through trials. The dropout layer causes the network to become thinner and reduces overfitting to the model, and speeds up the training process [14]:



(a) (b)

Figure 4. (a) Artificial Neural Network (b) Artificial Neural Network After Dropout Technique [14]

Fully Connected Layer

The fully connected layer is used to make the final predictions. This layer is used to classify the output generated from the convolution and pooling layers. Every neuron in this layer is connected to every neuron in the previous layer. Usually, this layer is followed by a dropout layer which increases the model's generalizability by preventing overfitting, which usually occurs at the training stage [15]. The purpose of using this layer is to use existing features to classify the input image into various classes based on the training dataset. The number of output probabilities from the fully-connected layer is 1 [16].

Sigmoid

The activation function is used to formulate the output of each neuron. The sigmoid activation function is the most common. This activation function will interpret the input range of –∞ and +∞ values into 0 and 1. The sigmoid activation function is defined as follows[17]:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{10}$$

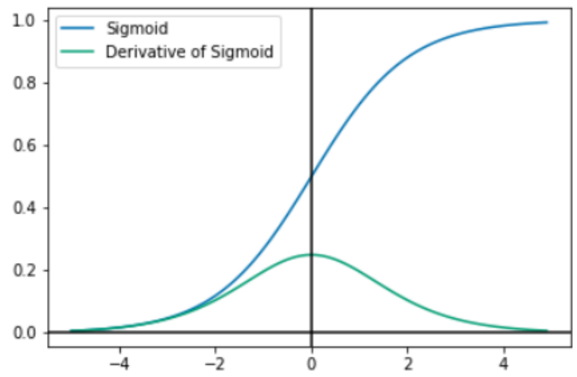
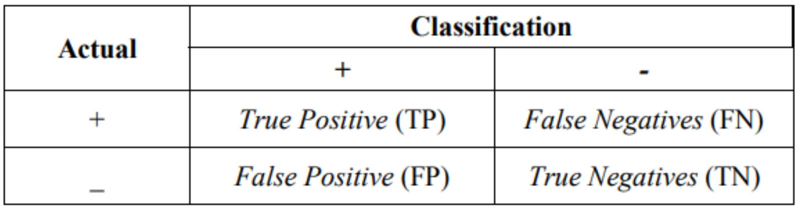


Figure 4. The sigmoid function and its derivatives [17]

Confusion Matrix

Confusion matrix is a method used to measure classification performance. This method can determine the accuracy, sensitivity, and specificity level by providing a detailed explanation of the correct and incorrect classification in the form of a matrix. To get the accuracy value on the confusion matrix, it is necessary to look for the True Negative (TN), False Negative (FN), True Positive (TP), and False Positive (FP) values. The confusion matrix table is shown in the following table.

Table 1. Confusion Matrix [18]



Actual	**Classification**	
	+	-
+	*True Positive* (TP)	*False Negatives* (FN)
_	*False Positive* (FP)	*True Negatives* (TN)

After getting the TP, TN, FP, and FN values, the accuracy values can be found using the following equation [18]:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{11}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{12}$$

$$Specifity = \frac{TN}{TN+FP} \tag{13}$$

Where TN is the true class of negative data that is correctly grouped as negative, TP is the actual class of positive data that is classed as positive, FP is the actual class of negative data that is misclassified as positive, and FN is the actual class of positive data that is incorrectly grouped as positive. to be negative [19].

Pre-processing

The data used in this study is in the form of CT-Scan image data in .jpg format downloaded from https://nbia.cancerimagingarchive.net/nbia-search/. The data taken is in the form of normal lung images with images of malignant lung nodules with specifications of more than 700 pixels or having a diameter of malignant lung nodules of more than 0.2 cm. The training data consists of 160 training data, including 80 image data sets of patients with normal lungs and 80 data sets of patients with malignant lung nodules. After the data is collected, the next process to be carried out is the initial data processing (pre-processing).

Resizing

Pre-processing is an initial stage that is carried out to eliminate some of the problems that can interfere with data processing. Digital images usually come in different sizes because the data collected may come from different sources. So that changing the size of the image or resizing is needed to uniformize the size of digital images so that processing techniques and feature extraction techniques do not produce anomalies [20]. In the resizing feature, the size of the image can increase or decrease in size. Uniforming the size of the image will get the same number of features from all the image data taken. In addition, resizing can reduce processing time so that it can improve system performance [21]. This research changes the image size to 225×225 pixels as the input size set in the CNN method. This is done so that the image data set has a uniform size. Smaller image sizes will help speed up the data management process.

Scaling

The data in this study will be grouped into two classes, namely normal lungs, and lungs with malignant nodules. As part of data labelling, non-numeric categories are converted to numeric values. Abnormal and normal values will be changed to be in the range of 0 and 1 using the selection of the min-max normalization feature, which is shown using the following equation [22]:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{13}$$

Where x is the original value of a feature, x_{min} is the minimum value of x, x_{max} is the maximum value of x, and x′ is the normalized value.

Training Datasets

The training process is where the CNN is trained to obtain high accuracy from the classification. This stage is carried out using the backpropagation and feedforward algorithms.

Dataset Testing

In the testing process, test data and previous training results are used. There is only a feedforward process and no backpropagation process in this process. The result of this process is the level of accuracy of the learning outcomes that have been carried out.

RESULTS AND DISCUSSION

Training Models

The system design stage was created to complete training and testing of lung cancer classification. This system design stage is a combination of software and hardware components of a system. This lung cancer detection system will use Python software. The CNN model is built on Google Colab, allowing anyone to write and run Python code. Google Colab supports many popular machine-learning libraries and is easy to run on your notebook.

Test Results for Number of Layers

The test results in this study were to compare the accuracy values for each layer in the proposed CNN architecture. The global parameter used in this test is to use the same number of epochs for each layer test, namely 30. The test phase is carried out by distributing the number of sequential filters from the first layer to the last layer, namely 16, 32, 64, 128, 256, 512, and 1024.

Table 2. Accuracy Testing Results at Each Layer of the Proposed CNN

Layer		Size and feature maps	Epoch	Accuracy	
				Train	Validation
Input	Image	225x225x3	-	-	-
1	Convolution	225x225x16	50	Error	-
	Max Pooling	112x112x16			
2	Convolution	112x112x32	50	Error	-
	Max Pooling	56x56x32			
3	**Convolution**	**56x56x64**	**50**	**0,9564**	**0,9019**
	Max Pooling	**28x28x64**			
4	Convolution	28x28x128	50	0,9469	0,8589
	Max Pooling	14x14x128			
5	Convolution	14x14x256	50	0,9392	0,8419
	Max Pooling	7x7x256			
6	Convolution	7x7x512	50	0,9278	0,8290
	Max Pooling	3x3x512			
7	Convolution	3x3x1024	50	0,9524	0,7969
	Max Pooling	1x1x1024			

Table 2 is the result of testing using four, five, six, and seven layers in this study. The table shows the best accuracy value when using three layers. This is because the more layers used, the more complex the neural network model will be, and can result in overfitting. The model is said to be overfitting when the validation results are worse than the training results. So the CNN model with three layers will be used for the next test.

Test Results for Variation in the Number of Filters

This stage is tested to determine the filter size in each convolution layer. This test aims to compare the accuracy value when using three different numbers of filters on each proposed CNN layer. Following are the details of the distribution of the number of filters in each layer:

1. The first model are 16, 32, and 64 pieces.
2. The second model is 32, 64, and 128 pieces.
3. The third model is 64, 128, and 256 pieces.
4. The fourth model is 128, 256, and 512 pieces.
5. The fifth model is 256, 512, and 1024 pieces.

Table 3. Test Results for Variation in the Number of Filters at 3 Layers of the Proposed CNN

No	Conv1	Conv2	Conv3	Accuracy	
				Train	**Validation**
1	3x3x16	3x3x32	3x3x64	0,8921	0,8489
2	3x3x32	3x3x64	3x3x128	0,9297	0,8569
3	3x3x64	3x3x128	3x3x256	0,9601	0,8839
4	**3x3x128**	**3x3x256**	**3x3x512**	**0,9612**	**0,8849**
5	3x3x256	3x3x512	3x3x1024	0,9542	0,8699

From the trials that have been carried out using five variations in the number of filters, the results are shown in Table 3. In the first experiment using 16, 32, and 64 filters, an accuracy value of 89.21 was obtained. The second experiment using 32, 64, and 128 filters obtained an accuracy value of 92.97%. The third experiment using 64, 128, and 256 filters obtained an accuracy value of 96.01%. Then in the fourth experiment using 18, 256, and 512 filters, the accuracy value was 96.12%. Therefore, increasing the number of filters can increase the accuracy of results because the more filters used, the more features extracted by the model. So filter extraction with 128, 256, and 512 values will be used in further research.

Batch Size Variation Test Results

In the third test, the parameter being evaluated is the number of batch sizes used at the model training stage. Batch size is the amount of training data used in one iteration. The batch size variations used in this test are 8, 16, 32, and 64.

Table 4. Results of the Effect of Batch Size Amount on the Proposed CNN Model

Batch Size	Accuracy		Overfitting
	Train	**Validation**	
8	0,9596	0,8949	0,0647
16	0,9645	0,8939	0,0706
32	**0,9656**	**0,9069**	**0,0587**
64	0,9389	0,8709	0,068
128	0,7596	0,8145	-0,0549

The use of batch size values from small to large is done so that the GPU and RAM of the laptop used are not too full. Table 4 shows the test results of the effect of batch size values on the proposed CNN model. Based on the results of the test, it was found that the training and validation accuracy values for batch size 8 were 95.96% and 89.49%. The training and validation accuracy values for batch size 16 were 96.45% and 89.39%. The best value was obtained for batch size 32 with training and validation accuracy of 96.56% and 90.69%, respectively. While the accuracy of training and validation when batch size 64 decreased to 92.89% and 87.09%. Then the accuracy when the batch size is 128 is 75.96% in training accuracy and 81.49% in validation accuracy. This shows that the model's ability to generalize data appears to decrease at large batch size values.

Dropout Variation Trial Results

The parameter evaluated in the fourth experiment was the dropout variation used in the model training stage. Dropouts are a type of modification used at the last layer in architecture to reduce overfitting by increasing validation accuracy. Dropout = 0.1 means that 0.1 neurons are randomly removed. The dropout variation used in this test is 0.1; 0.3; 0.5; 0.7; and 0.9.

Table 5. Results of Variation of Dropout Values Against the Proposed CNN Model

Dropout	Accuracy		Overfitting
	Train	Validation	
0,1	0,9616	0,9009	0,0607
0,3	0,9622	0,8979	0,0643
0,5	**0,9625**	**0,9069**	0,0556
0,7	0,9494	0,8999	0,0495
0,9	0,8735	0,8599	0,0136

Table 5 shows the test results of the variation in the dropout values in the proposed CNN model. Based on the test results, it was found that the greater the dropout value, the smaller the overfitting value that occurs in the neural network model. However, when viewed from how it works, it is obtained that the accuracy will decrease if the dropout value is too large. This happens because the greater the dropout value, the more neurons will be ignored, so the model cannot recognize the data properly.

Epoch Variation Trial Results

In the third test, the parameter evaluated is the number of epochs used in the model training stage. The epoch will affect the number of iterations required for the learning rate to transition. The epoch variations used in this test are 20 epoch, 60 epoch ¸ 80 epoch, and 100 epoch.

Table 6. Results of the Influence of the Number of Epochs on the Proposed CNN Model

Epoch	Accuration		Overfitting
	Training	Validation	
20	0,9037	0,8299	0,0738
60	**0,9672**	**0,9074**	**0,0598**
80	0,9716	0,9043	0,0673
100	0,9782	0,8854	0,0928

Table 6. shows the test results of the effect of the number of epochs on the proposed CNN model. Based on the test results, it was found that the greater the number of epochs, the higher the accuracy value. This is because the more the number of epochs, the more often the model will repeat the learning rate on data to make the input data more recognizable. Neural networks on a deep learning basis require thousands of data, whereas this study only used 200 data, so the more epochs are carried out, the more overfitting the data will be.

Testing Models

Based on several variations that have been made, the most optimal results are obtained for the proposed CNN model. These results are then tested using separate data not trained in the system. Tests were carried out to obtain the model's accuracy, sensitivity, and specificity parameters using a confusion matrix. Then the output of the test model is shown in table 7, where the test data that the model has processed will be given predictions according to the labels stored from the training results. The following are the results of detection by the system based on the variations made.

Table 7. Proposed CNN Confusion Matrix After Modification

	Suspected Lung Cancer	Normal Lung
Suspected Lung Cancer	10	0
Normal Lung	1	9

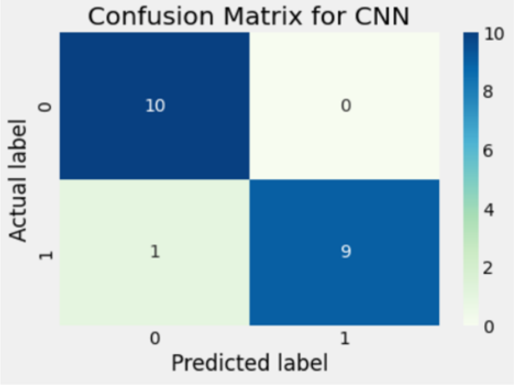


Figure 6. Proposed CNN Confusion Matrix after modification

Results obtained Accuracy = 95%, Sensitivity = 90% and Specificity = 100%. The results of this study are to compare the results of the best accuracy for the convolutional neural network before and after repairs. The activation functions used are ReLU and Sigmoid. In the neural network, the model could not recognize the test data well before being modified, so an accuracy value of only 70% was obtained. The modified convolution neural network model is better regarding test data, namely 95%.

CONCLUSION

This study applied the CNN method for lung cancer image classification using the hardware library with the Python programming language. The effect of improvements on the convolutional neural network algorithm makes the accuracy results increase to 95% with the CNN architecture of the proposed model using three layers, 128 filters on the first layer, 256 on the second layer, and 512 filters on the third layer, then with 32 batch sizes, 0.5 dropouts. The modified convolutional neural network algorithm has been proven capable of optimizing accuracy results in lung cancer detection applications using lung inputs with malignant nodules and normal lungs. The results showed that the Convolutional Neural Network (CNN) method can be used to identify lung cancer with an accuracy of 95%, a sensitivity of 90%, and a specificity of 100%.

REFERENCE

1 Rachmaniar N, S., & Fuad, Y. 2017. Stabilitas Sistem Dinamik Pertumbuhan Sel Kanker dengan Terapi Radiasi. *Jurnal Ilmiah Matematika Volume*, *3*(6).

2 Joseph, J., & Rotty, L. W. 2020. Kanker Paru: Laporan Kasus. *Medical Scope Journal*, *2*(1).

3 Mao, Y., Yang, D., He, J., & Krasna, M. J. 2016. Epidemiology of lung cancer. *Surgical Oncology Clinics*, *25*(3), 439-445.

4 Lakshmanaprabu, S. K., Mohanty, S. N., Shankar, K., Arunkumar, N., & Ramirez, G. 2019. Optimal deep learning model for classification of lung cancer on CT images. *Future Generation Computer Systems*, *92*, 374-382.

5 Thabsheera, A. A., Thasleema, T. M., & Rajesh, R. (2019). Lung cancer detection using CT scan images: A review on various image processing techniques. In *Data Analytics and Learning*, 413-419. Springer, Singapore.

6 Polat, H., & Danaei Mehr, H. 2019. Classification of pulmonary CT images by using hybrid 3D-deep convolutional neural network architecture. *Applied Sciences*, *9*(5), 940.

7 Dong, Y., Pan, Y., Zhang, J., & Xu, W. 2017. Learning to read chest X-Ray images from 16000+ example using CNN. In *2017 IEEEA/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 51-57. IEEE.

8 Listyalina, L., Utari, E. L., & Puspaningtyas, D. E. 2020. Penentuan Penyakit Paru Dengan Menggunakan Jaringan Saraf Tiruan. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, *11*(1), 233-240.

9 Setiawan, W., Utoyo, M. I., & Rulaningtyas, R. 2019. Classification of neovascularization using convolutional neural network model. *TELKOMNIKA*, *17*(1), 463-472.

10 Noman, N. (2020). A shallow introduction to deep neural networks. *Deep Neural Evolution: Deep Learning with Evolutionary Computation*, 35-63.

11 Ilahiyah, S., & Nilogiri, A. 2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, *3*(2), 49-56.

12 Ker, J., Wang, L., Rao, J., & Lim, T. 2017. Deep learning applications in medical image analysis. *Ieee Access*, *6*, 9375-9389.

13 Ding, B., Qian, H., & Zhou, J. 2018. Activation functions and their characteristics in deep neural networks. In *2018 Chinese control and decision conference (CCDC)*, 836-1841. IEEE.

14 Peryanto, A., Yudhana, A., & Umar, R. 2020. Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format J. Ilm. Tek. Inform*, *8*(2), 138.

15 Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. 2018. Malware classification with deep convolutional neural networks. In *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*, 1-5. IEEE.

16 Coşkun, M., Uçar, A., Yildirim, Ö., & Demir, Y. 2017. Face recognition based on convolutional neural network. In *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, 376-379. IEEE.

17 Rasamoelina, A. D., Adjailia, F., & Sinčák, P. 2020. A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 281-286. IEEE.

18 Foeady, A. Z., Novitasari, D. C. R., Asyhar, A. H., & Firmansjah, M. 2018. *Automated Diagnosis System of Diabetic Retinopathy Using GLCM Method and SVM Classifier. 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*.

19 Cheng, T. Y. D., Cramb, S. M., Baade, P. D., Youlden, D. R., Nwogu, C., & Reid, M. E. 2016. The international epidemiology of lung cancer: latest trends, disparities, and tumor characteristics. *Journal of Thoracic Oncology, 11*(10), 1653-1671.

20 Alenezi, N. S. A. 2019. A method of skin disease detection using image processing and machine learning. *Procedia Computer Science*, *163*, 85-92.

21 Sharma, A., Shinde, S., Shaikh, I. I., Vyas, M., & Rani, S. 2021. Machine learning approach for detection of diabetic retinopathy with improved pre-processing. In *2021 International conference on computing, communication, and intelligent systems (ICCCIS)*, 517-522. IEEE.

22 Raihan-Al-Masud, M., & Mondal, M. R. H. 2020. Data-driven diagnosis of spinal abnormalities using feature selection and machine learning algorithms. *Plos one*, *15*(2), e0228422.