

Regionprops Segmentation in Convolutional Neural Network for Identification of Lung Cancer Disease and Position

Zahra Ghina Syafira¹⁾, Christy Atika Sari^{*1)}, Ibnu Utomo Wahyu Mulyono¹⁾, Feri Agustina¹⁾, Suprayogi¹⁾, Mohamed Doheir²⁾

¹⁾Informatics Engineering, Universitas Dian Nuswantoro, Semarang, Indonesia

²⁾Faculty of Technology Management & Technopreneurship, Universiti Teknikal Malaysia Melacca, Melacca, Malaysia

* Corresponding author: christy.atika.sari@dsn.dinus.ac.id

Abstract

Lung cancer is one of the leading causes of death in the world, so early detection is very important to increase the chances of patient recovery. This study aims to develop a method for identifying lung cancer types using Convolutional Neural Network (CNN) combined with Regionprops segmentation technique to determine the position of cancer in CT scan images. The dataset used consists of 1,294 CT scan images classified into three classes, namely Benign, Malignant, and Normal, with variations in the ratio of training and testing data: 80:20, 70:30, 60:40, 50:50, and 40:60. The CNN method is used to perform classification, while the Regionprops segmentation technique is applied to determine the position of the cancer. The results showed that the model with a data ratio of 80:20 achieved the highest accuracy of 99.54%, indicating a very good generalization ability of the model. The Regionprops segmentation technique successfully separated the nodule area in the CT scan image clearly, thus providing more detailed information regarding the position of the cancer. The conclusion of this study shows that the combination of CNN and Regionprops segmentation methods is effective in detecting and analyzing lung cancer and has the potential to be used as a diagnostic tool in the medical field. This study recommends further testing with a larger dataset and optimization of model parameters to improve classification and segmentation performance.

Keywords : Lung Cancer, Convolutional Neural Network (CNN), CT Scan, Regionprops Segmentation, Cancer Identification, Deep Learning.

1 Introduction

One of the important organs in the human respiratory system is the lungs which function as a place to exchange oxygen from the air with carbon dioxide from the blood [1]. In every human body there are two lungs located in the chest cavity which are protected by the ribs [2], [3]. The right lung consists of three lobes, namely the superior lobe, the middle lobe, and the inferior lobe. While the left lung has two lobes, namely the superior lobe and the inferior lobe [4]. The left lung only has two lobes because there is a heart organ located in the groove in the lower lobe [4], [5]. Both lungs are covered by a protective layer called pleural tissue [6]. The function of the pleural tissue itself is as a lubricant in the lung cavity so as not to irritate the lungs when breathing. This tissue has two thin layers that have their respective benefits. The first layer is the visceral pleura which is attached directly to the surface of the lungs and the second layer is the parietal pleura which lines the chest cavity [1], [6]. One of the most important and valuable things in the world of health is maintaining lung health, because if lung damage occurs it can cause death with various symptoms that arise, such as coughing up phlegm,

shortness of breath, coughing up blood, and can even cause weight loss. Lung diseases include asthma, pneumonia, bronchitis, tuberculosis, lung cancer, pulmonary edema, sarcoidosis, and so on.

Lung cancer is one of the diseases that causes the most deaths in the world. In 2019 the estimated number of deaths due to lung cancer in the United States reached 142,670 which is 23.5% of all cancer cases [7], [8]. While in 2020 in Indonesia, lung cancer was recorded as ranking third with 34,783 cases that have been observed by the Global Burden of Cancer Study from the World Health Organization [6]. Lung cancer or what is called bronchogenic carcinoma is a type of malignant tumor that originates from the respiratory tract. Not only does it occur in the lungs but this cancer is also found in the thoracic cavity or what is called the mediastinum, namely the cavity between the right and left lungs. The main cause of lung cancer is smoking [9]. Cigarette smoke contains more than 7,000 chemicals, with hundreds of particles in it being toxic and 70 other substances containing carcinogens that can cause cancer [10]. However, smoking is not the only factor that causes lung cancer, namely air pollution, lifestyle, radiation exposure, asbestos exposure, and genetic factors. Men have a higher risk of developing lung cancer compared to women because men have a higher tendency to smoke. Lung cancer has two types, namely small cell lung cancer (SCLC) and non-small cell lung cancer (NSCLC) [11]. SCLC is a very aggressive type of cancer because it spreads very quickly. This type of lung cancer is usually caused by heavy smoking habits which are characterized by coughing and shortness of breath. Meanwhile, NSCLC is the most common type that grows slower than other types. The symptoms caused by this type are coughing up blood, shortness of breath, and weight loss. The growth of these nodules can be detected using images in the form of CT scans or X-ray films which take about 10–30 minutes. With a long enough time based on this experience, many researchers have conducted research using image detection methods using deep learning, namely Convolutional Neural Network (CNN) [12], [13], [14]. CNN is a part of machine learning that uses deep learning techniques in pattern recognition in image detection by extracting complex features automatically and efficiently, so this method is suitable for use in large-scale calculations. In this study, researchers used images in the form of CT scans as image detection that would be accurate using CNN. CT scans are standard radiology tools used to detect lung cancer nodules [15].

2 Literature Review

In a study conducted by [16] based on a convolutional neural network (CNN) model based on the AlexNet architecture, it showed significant potential in early detection and classification of lung cancer nodules using CT scans. Using a dataset consisting of 110 CT scans categorized as normal, benign, and malignant, this model achieved an overall accuracy of 93.548%, sensitivity of 95.714%, specificity of 95%, precision of 97.1015%, and an F1 score of 96.403%. This study emphasizes the importance of using artificial intelligence in improving the process of lung cancer diagnosis, which can assist healthcare professionals in early detection and classification of the disease.

The second study was conducted by [17] This study showed that the CNN-SVM hybrid method for classifying lung CT images achieved a classification accuracy of 97.91% on the test set, with a sensitivity of 97.90%, a specificity of 99.32%, and a precision of 97.96%. The number of datasets used in this study was 5103 images, which were obtained through the color transformation technique with intensity adjustment. These results emphasize the effectiveness of deep learning algorithms in medical imaging and show the potential for further improvement using larger datasets and various imaging modalities.

The next study conducted by [18] showed that the detection and segmentation system for cancer nodule areas in CT Scan images using the Mask R-CNN method has high performance in identifying benign nodules, with an average precision of 85.2% and a sensitivity of 95.8%. The number of datasets used in this study was 500 images, consisting of 100 tumor object images and 400 non-tumor object images. However, this system still has difficulty in detecting malignant nodules, especially those with low contrast and unclear boundaries, so further development is needed to improve the detection of malignant nodules. Meanwhile, the last study conducted by [19] showed very good results, with a training accuracy of 98.60% and a validation accuracy of 99.20% taken from 300 datasets. This model also showed high effectiveness in distinguishing between cancer types, with performance metrics such as precision, recall, and F1-score showing satisfactory results. This study emphasizes the potential of CNN in assisting fast and accurate lung cancer diagnosis, and provides hope for the development of further detection systems using advanced algorithms.

The last study conducted by [20] said that for lung cancer detection through CT scan images taken in the LUNA 16 dataset consisting of n axial scans measuring 512×512 with 200 images per scan. Of these, only 1,351 were positive nodules based on annotation, while the rest were negative nodules. Showing significant results, with the overall accuracy of the ensemble model reaching 95%. The individual models in the ensemble, namely CNN1, CNN2, and CNN3, achieved accuracies of 94.07%, 94.44%, and 94.23%, respectively. This study emphasizes the importance of early detection of lung cancer and demonstrates the potential of deep learning methods in improving the efficiency and accuracy of lung nodule detection. Based on several studies above, it can be concluded that these studies highlight the advantages of the CNN method in detecting lung cancer CT Scans with varying levels of accuracy. Meanwhile, the advantage of the research conducted by the researcher is that after the data has been identified, the type and accuracy results, the data will be segmented using regionprops to determine the position of the existing cancer. This is what makes the difference between researchers and the five studies above. This study provides a strong foundation for compiling a final assignment with a focus on the application and evaluation of methods that have been proven effective.

3 Research Methods

3.1 Convolutional Neural Network (CNN)

Machine learning is used to teach how to handle data more efficiently [21], [22]. In machine learning, algorithms are used to recognize patterns in data to make predictions or decisions based on existing patterns [23], [24]. The process of making these decisions uses training data as model training to map input to output for the desired results. Machine learning and deep learning are closely related, where deep learning is a more specific subfield of machine learning [25], [26]. One of the most well-known architectures in deep learning is the Convolutional Neural Network (CNN), which is designed to process and recognize patterns in grid-shaped data, such as images [27], [28]. In the context of machine learning, CNN functions as a deep learning model that can automate the process of extracting features from visual data without the need for human intervention [29]. CNN learns from data through convolutional layers that capture important features such as edges, textures, or objects in the image, then the pooling layer reduces the dimensionality of the data, and the fully connected layer performs classification. Unlike traditional machine learning which often requires manual feature selection, CNN automatically learns and refines relevant features to produce more accurate models, especially in tasks such as image recognition, object detection, and medical image analysis [30]. Deep learning with

CNNs enables machine learning to achieve much higher levels of performance in complex, big data-driven problems.

3.2 Proposed Stages

Identification and segmentation process flow using CT Scan images of lung cancer using CNN method, and regionprops segmentation technique as follow in Figure 1.

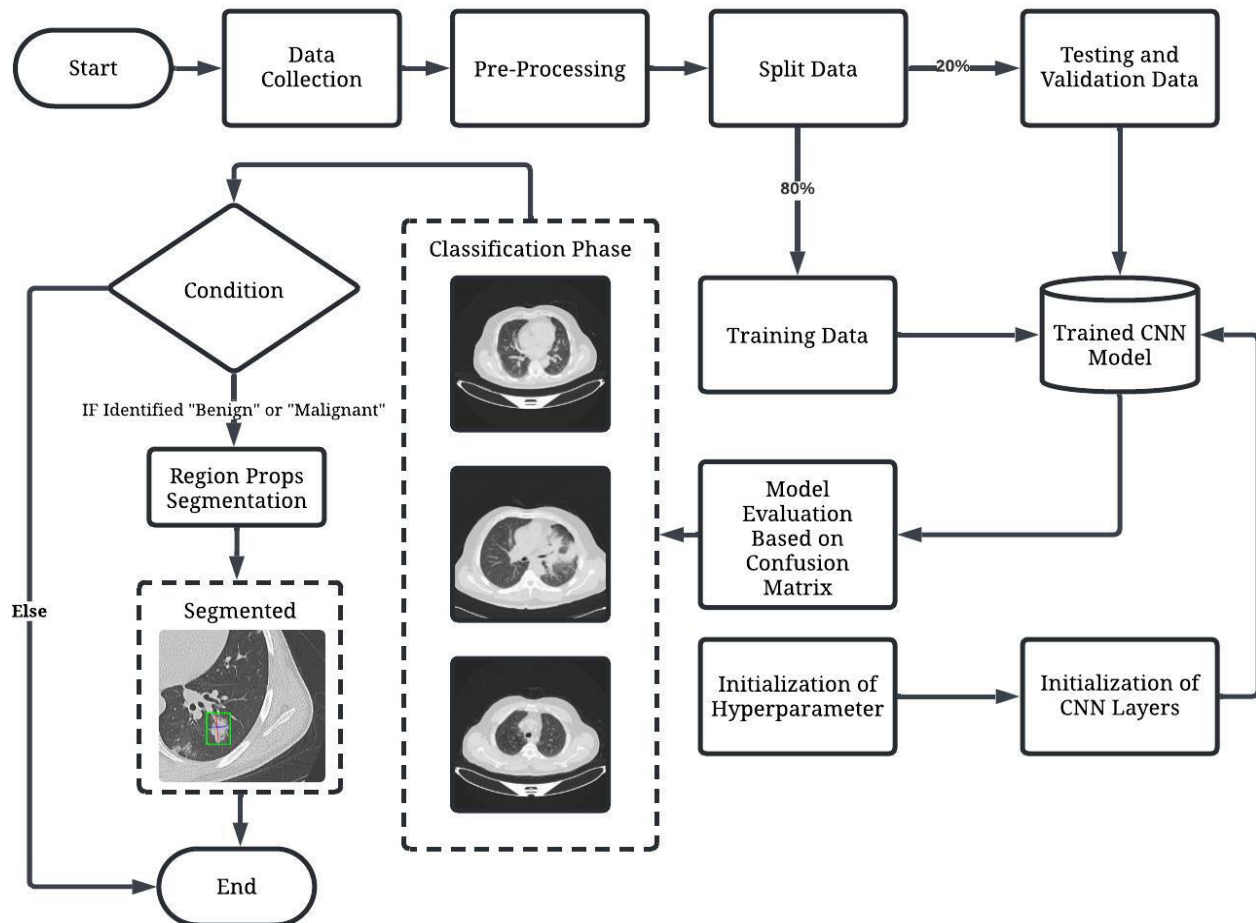


Figure 1 Proposed stages for segmentation technique

1. Data Collection. This study uses secondary data techniques that are not directly generated by researchers, but obtained from other sources, namely Kaggle. In addition, researchers also conducted literature studies through journals, articles, and other sources related to lung cancer, image processing, and CNN as the basis for research. The data is divided into three groups, namely benign, malignant, and normal lung cancer from <https://www.kaggle.com/datasets/adityahimkar/iqothnccd-lung-cancer-dataset> as visualized in Figure 2. The dataset used by this researcher is public, which means that the dataset can be used by anyone who needs it. The dataset contains data in the form of images of 1,294 CT-Scan of human lungs. All data in this dataset has files with jpg and png types with different dataset sizes.



Figure 2 Sample datasets of lung cancer

2. **Splitting Data.** The testing data used is 20% which functions to predict data on CT scan images, while the training data is 80% which functions to make predictions or decisions from information obtained from the testing data [31]. In addition, this study uses validation data which functions to optimize parameters so that the model does not experience overfitting. From a total dataset of 1,294 lung cancer CT scans, a training data split was carried out, including: Benign cases 96 data, Malignant cases 449 data, and Normal cases 333 data. While for the testing data split, including: Benign cases 24 data, Malignant cases 112 data, Normal cases 83 data.
3. **Regionprops Segmentation.** Segmentation is a technique in image processing used to analyze and extract images that have been segmented in a certain area. In addition, regionprops are also used to calculate the geometric characteristics of objects in the image in each region [15], [32]. This technique is often used by medical professionals to identify the size and position of lung cancer on CT scans. Regionprops work with CNN to improve image analysis to be more detailed. By using both methods, areas that indicate lesions or nodules can be isolated for further analysis. The segmentation process begins with data preprocessing, after which the thresholding technique is applied to distinguish normal and abnormal tissue. Furthermore, regionprops are then used to analyze the identified region of interest (ROI), calculating properties such as area, perimeter, orientation, eccentricity, and other relevant aspects. Regionprops also allows precise identification of cancer positions by providing coordinates of segmented ROIs. This information is very useful for doctors in determining the specific location of cancer in lung CT scan images.

With the help of this technique, the detection process becomes faster and more accurate, and supports better clinical decision making.

4. CNN Training Hyperparameter. Training Hyperparameter is the process of setting parameters that are not learned by the model during training [33]. These hyperparameters are determined before the training process begins, this can significantly affect model performance [34]. Choosing the right hyperparameters can make the model work more optimally and provide more accurate prediction results as in Table I and Table II. The hyperparameter steps are divided into maxepoch, ADAM optimizer, learning rate, mini batch size, and validation sequence. Furthermore, after the data has been processed through these steps, the data will be continued using a confusion matrix to evaluate and compare the classification results with the actual values. MaxEpoch is a hyperparameter used to determine the maximum number of epochs in the deep learning model training process. The number of one epoch indicates that the model has seen the entire dataset once, so MaxEpoch can set how many times the model will repeat the process during training. The more epochs that train, the longer the model trains, so there are more opportunities to learn from the data that causes overfitting. However, if the number of epochs is too small, the model may not learn well, which can result in underfitting. The determination of the MaxEpoch value is usually done based on experiments and depends on the characteristics of the dataset used. A large dataset will require a larger number of epochs for the model to learn patterns well. However, the optimal number of epochs also depends on the training method and evaluation technique used. MaxEpoch plays an important role in controlling the balance between time, training, and model quality. By choosing the right values and combining them with techniques such as early stopping, model training can become more efficient and produce models that are able to perform optimally on new data. Table 1 shown the proposed hyperparameter.

Table 1 CNN training hyperparameter option

Hyperparameter	Value
Optimizer	Adam
Execution Environment	GPU
Max Epochs	16
Mini Batch Size	32
Validation Data	20%
Validation Frequency	30
Verbose	False

5. Adam Optimizer. Optimize Adaptive Moment Estimation (ADAM) is one of the hyperparameters used to efficiently update model weights during the backpropagation process [35]. In ADAM there are two algorithms, namely AdaGrad which adjusts the learning rate for each parameter and RMSProp which stabilizes learning through the average of the squared gradient [36]. The advantages of ADAM are that it can handle large data, has good performance in solving non-stationary problems, and has a stable ability in convergence. ADAM also requires less adjustment of the learning rate compared to other optimization algorithms, making it very practical in many cases, especially when time is limited for experimentation and model training. Technically, ADAM calculates two moments, the first calculates the average gradient which helps the model move in a more stable direction and reduces unnecessary fluctuations in the training process [37]. Second, the average of the squared gradient is used to adjust the learning

rate adaptively for each parameter. Using both of these pieces of information, ADAM can update the model weights in a more efficient way and speed up convergence without requiring complicated manual tuning of the learning rate [38]. However, while ADAM often produces excellent results, there are times when it can suffer from problems such as overfitting if not tuned properly. In another side, choosing the right mini-batch size is critical for training performance. If the mini-batch size is small, it can help reduce the variance of gradient estimates and can speed up convergence during training. However, if the mini-batch size is too small, it will result in less accurate gradient estimates. Commonly used mini-batch sizes range from 32 to 256, but the optimal value depends on the dataset and the available computational power. This is very important to achieve a balance between training speed and model accuracy.

Table 2 Proposed CNN layer

Name	Type	Activations	Learnables	
Imageinput	Image Input	256x256x3	-	
conv_1	Convolution	256x256x8	Weights	3x3x3x8
			Bias	1x1x8
batchnorm_1	Batch Normalization	256x256x8	Offset	1x1x8
			Scale	1x1x8
relu_1	ReLu	256x256x8	-	
maxpool_1	Max Pooling	128x128x8	-	
conv_2	Convolution	128x128x16	Weights	3x3x3x16
			Bias	1x1x16
batchnorm_2	Batch Normalization	128x128x16	Offset	1x1x16
			Scale	1x1x16
relu_2	ReLu	128x128x16	-	
maxpool_2	Max Pooling	64x64x16	-	
conv_3	Convolution	64x64x32	Weights	3x3x3x32
			Bias	1x1x32
batchnorm_3	Batch Normalization	64x64x32	Offset	1x1x8
			Scale	1x1x8
relu_3	ReLu	64x64x32	-	
Fc	Fully Connected	1x1x3	Weights	3x131072
			Bias	3x1
Softmax	Softmax	1x1x3	-	
Classoutput	Classification Output	-	-	

3.3 Confusion Matrix

Confusion Matrix is a method used to evaluate and compare classification results with actual values. In showing the number of correct and incorrect predictions, the confusion matrix uses a table generated by the model in each class [39], [40]. There are four terms to present the classification results, namely:

1. True Positive (TP): the model predicts the positive and correct class.
2. True Negative (TN): the model predicts the negative and correct class.
3. False Positive (FP): the model predicts the positive class but is wrong.
4. False Negative (FN): the model predicts the negative class but is wrong.

From these four terms, a matrix can be found that can evaluate model performance, namely accuracy, recall, precision, and F1-Score. Accuracy is a value that presents how much data can be classified according to actual or correct data as in (1). Precision is used to measure how many of the positive predictions are actually positive, as in (2). Recall is used to measure how much data from all

positive cases is successfully predicted correctly, as in (3). F1 Score is a combination of precision and recall which is used to give equal weight to both. This is useful when there is a class imbalance in the dataset, as in (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

4 Results and Discussion

Here, we implement the data split technique to perform data testing and training (Nguyen et al. 2021). In dividing data for model analysis and training, researchers use several ratios to ensure that the model created can be tested objectively on data that has never been seen during the training process (Ranftl et al. 2022). The data divisions include 80:20, 70:30, 60:40, 50:50, and 40:60. Larger ratios for training such as 80:20 and 70:30 will provide more data for the model to learn and are suitable for large datasets. While balanced ratios such as 60:40 and 50:50 will provide a larger amount of test data and are useful for deeper analysis and evaluation when using smaller datasets. And finally, this 40:60 ratio is actually rarely used, but is useful in certain cases where the focus is on validation or testing the model as a whole.

The following is an explanation of the graph in Figure 3 using a training process comparison of 80:20. At the beginning of the iteration in the first epoch, the model appears to have a low accuracy of around 40%, but after passing the first 50 iterations and entering the second epoch, there is an increase in accuracy reaching 80%. Every 30 iterations, the model performs validation to check its performance against previously unseen data. This pattern is consistent throughout the training process where the black dots on the graph indicate the process. What is interesting about the total of 432 iterations that occurred during the 16 epochs is that the model's performance continues to increase gradually in each epoch. In the middle of training, in the 8th to 12th epoch, the model has achieved very good performance but continues to show small improvements. Entering the 13th to 16th epoch, the model managed to maintain accuracy consistently with the training and validation accuracy values running in balance. This process ended right at the 432nd iteration and the 16th epoch with a perfect accuracy result of 99.54%.

The following is an explanation of the graph in Figure 4 using a 70:30 training process comparison. In this training process, the model underwent 16 epochs with 24 iterations in each epoch. At the beginning of training, the model started with an accuracy of around 40% and there was a very rapid increase in the first 50 iterations where the model managed to achieve an accuracy of around 90%. Every 30 iterations, the model performs validation which is marked by black dots on the graph. Of the total 384 iterations, we can see that the largest performance increase occurred at the beginning of training. After passing the 3rd epoch, the accuracy increase gradually becomes significant. In the middle of training, from the 8th to the 12th epoch, the model has achieved stable performance with accuracy above 95%. At the end of the epoch, the model maintains its high performance very well. The training process ends at the 384th iteration which is the last iteration of the 16th epoch with the final accuracy reaching 98.78%. The total time required for the entire training process is 1 minute 24 seconds, using a constant learning rate of 0.001 on a single GPU.

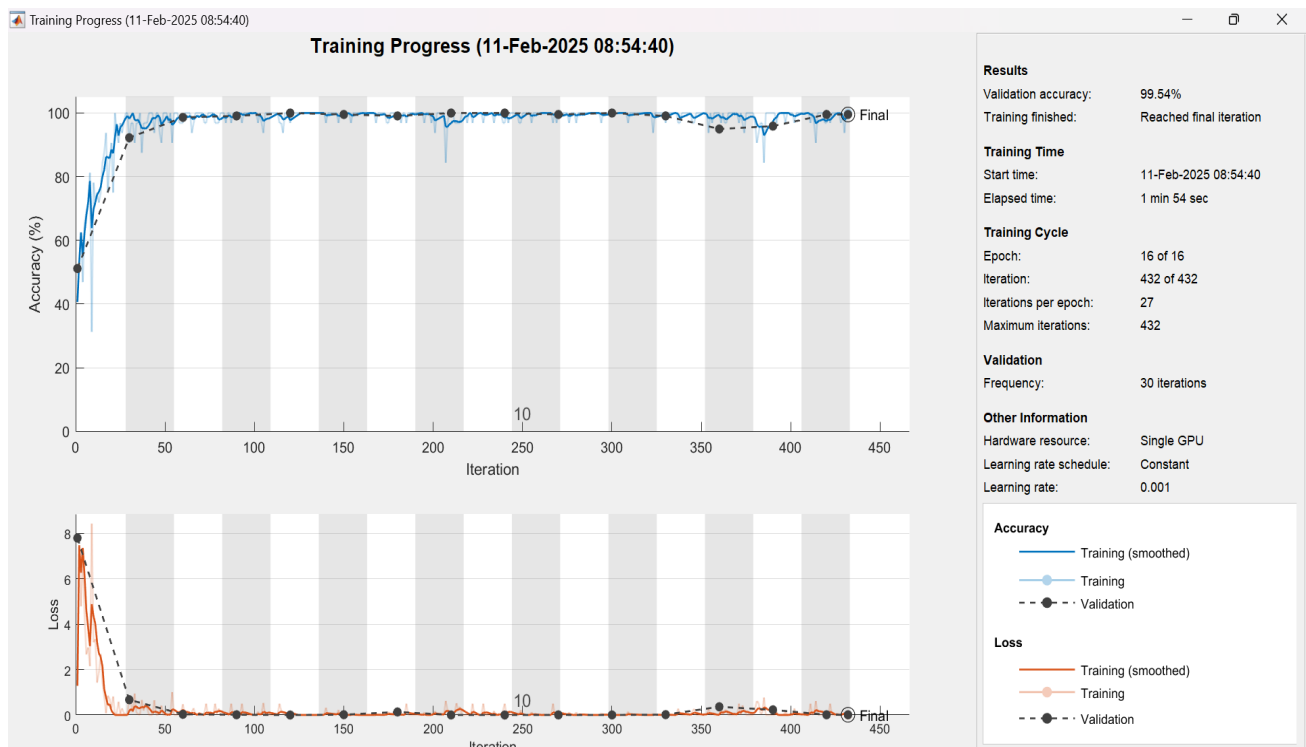


Figure 3 Training Process Training 80:20

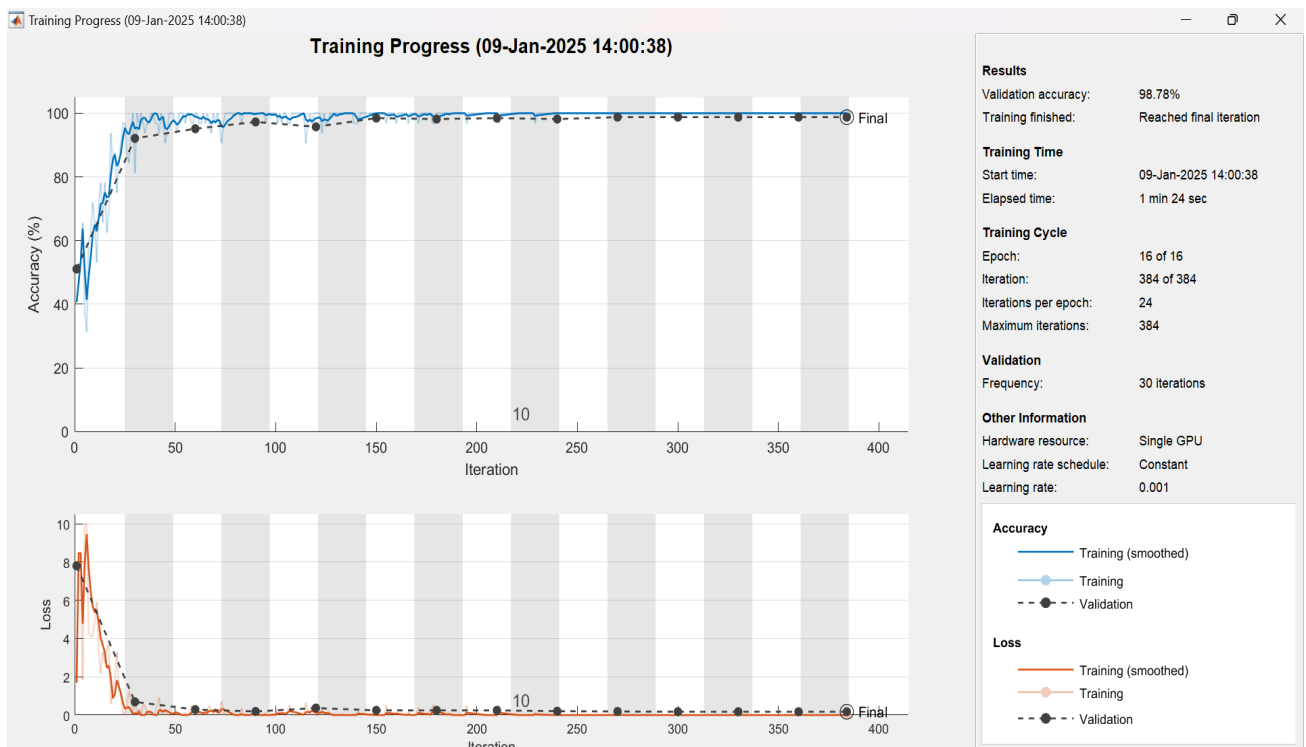


Figure 4 Training Process Training 70: 30

Figure 5 uses a 60:40 training process comparison. At the beginning of this training, the model starts with a low accuracy of around 40%. However, there is a very rapid increase in the first 50 iterations where the model managed to increase its accuracy to 90%. The training process lasts for 1

minute 15 seconds by validating every 30 iterations as indicated by the black dots on the graph. Of the total 320 iterations that occurred during 16 epochs, there was a significant increase after passing the 3rd epoch. In the middle of training, from the 8th to the 12th epoch, the model showed stable performance with accuracy results above 95%. Entering the last epoch, the model managed to maintain its performance well. The training process ended in the 16th epoch with accuracy results reaching 95.66%. The learning rate used remains constant at 0.001 with training using a single GPU. The loss graph also shows a healthy decline, starting at around 7 and dropping steadily to near 0 at the end of training.

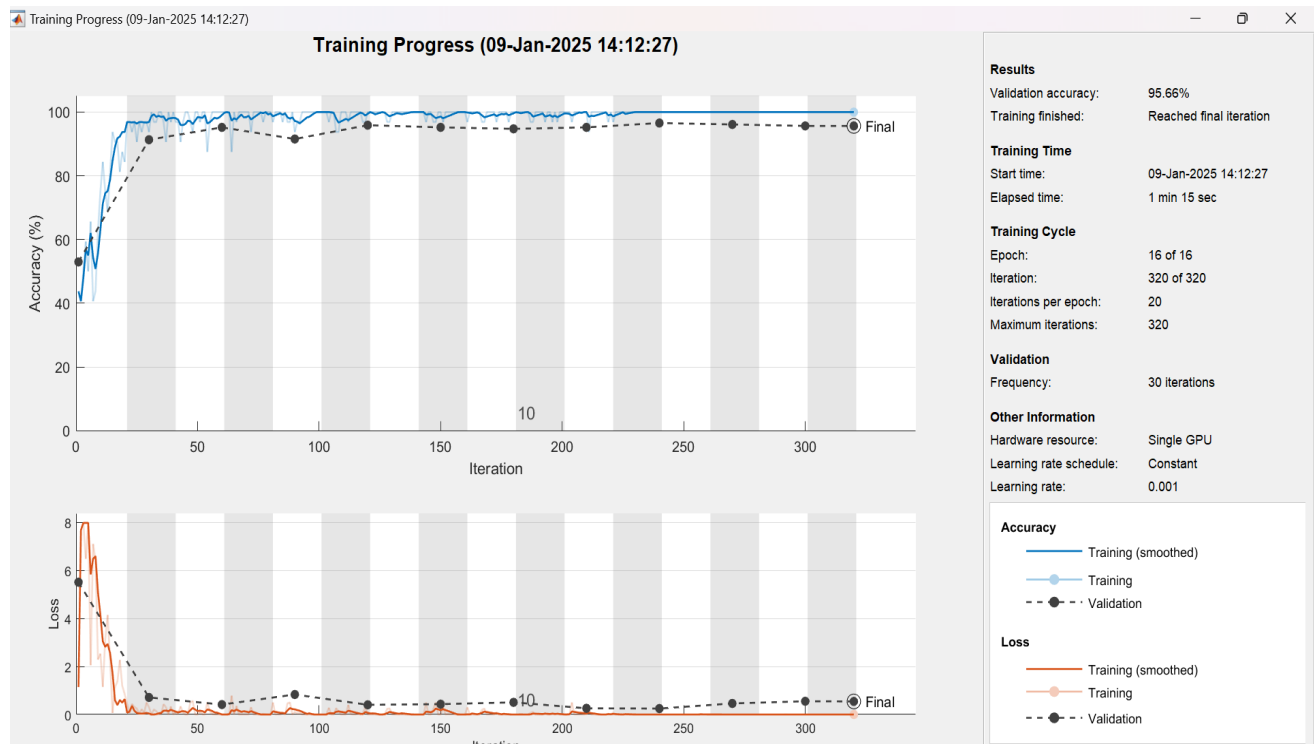


Figure 5 Training Process Training 60:40

The following is an explanation of the graph in Figure 6 using a 50:50 training process comparison. At the beginning of training in the first epoch, the model started with an accuracy of around 40%. However, there was a very rapid increase in the first 50 iterations where the model managed to increase its accuracy to 90%. The process lasted for 1 minute 2 seconds which was carried out every 30 iterations as marked by black dots on the graph. The journey of 272 iterations showed an interesting pattern, where performance showed a significant increase after passing the 3rd epoch which increased gradually. In the middle of training, from the 8th to the 12th epoch, the model had achieved stable performance with an accuracy above 95%. In the last epoch, the model managed to maintain its performance well. Training ended in the 16th epoch with an accuracy result of 98.72%. The learning rate remained constant at 0.001 with the use of a single GPU. The loss graph also shows a healthy decline, starting at around 8 and decreasing steadily to near 0 at the end of training.

The following is an explanation of the graph in Figure 7 using a 40:60 training process comparison. At the beginning of training in the first epoch, the model starts with an accuracy of around 40%. In the first 30 iterations, a very rapid increase is seen to reach an accuracy of around 85%. The training process is carried out every 30 iterations as indicated by the black dots on the graph. Of the

total 208 iterations, after passing the 3rd epoch there was a significant increase in accuracy and the model began to find a stable pattern. In the middle of training, from the 8th to the 12th epoch, the model had achieved stable performance with an accuracy above 95%. In the last epoch, the model managed to maintain its performance well. Training ended at the 16th epoch by achieving accuracy results. The learning rate was kept constant at 0.001 using a single GPU. The loss graph shows a healthy decline, starting at around 8 and dropping steadily to near 0 by the end.

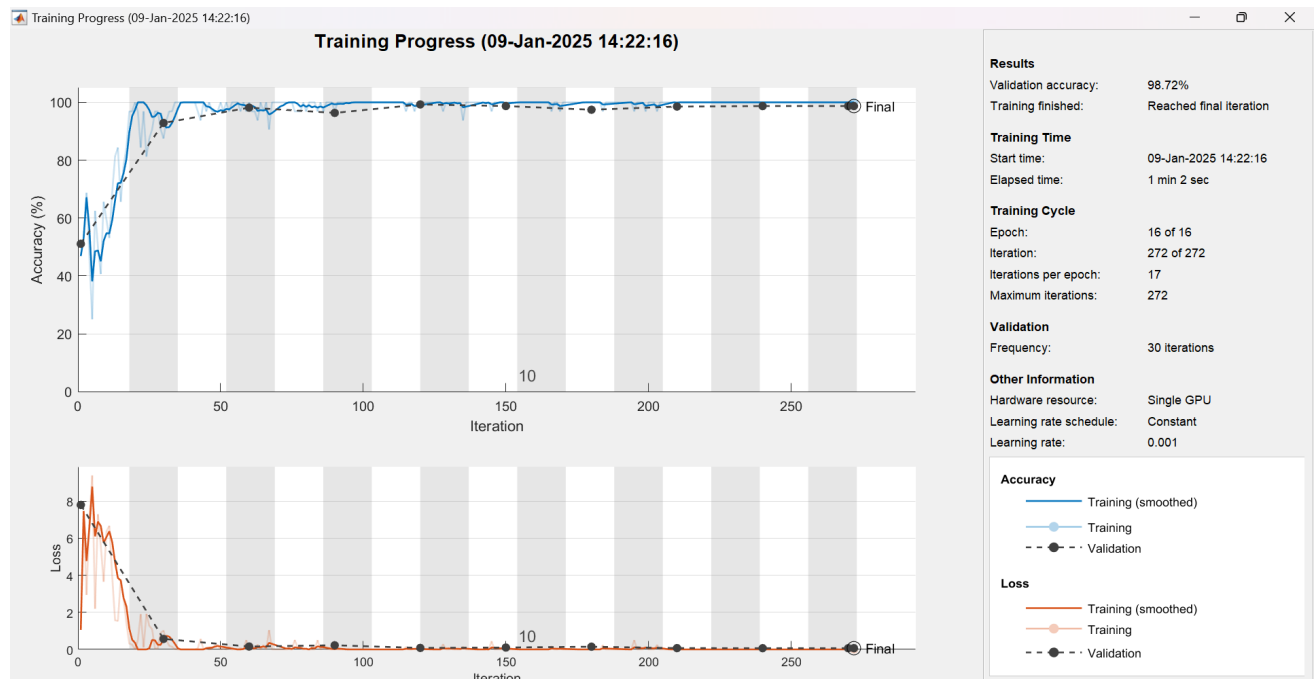


Figure 6 Training Process Training 50:50

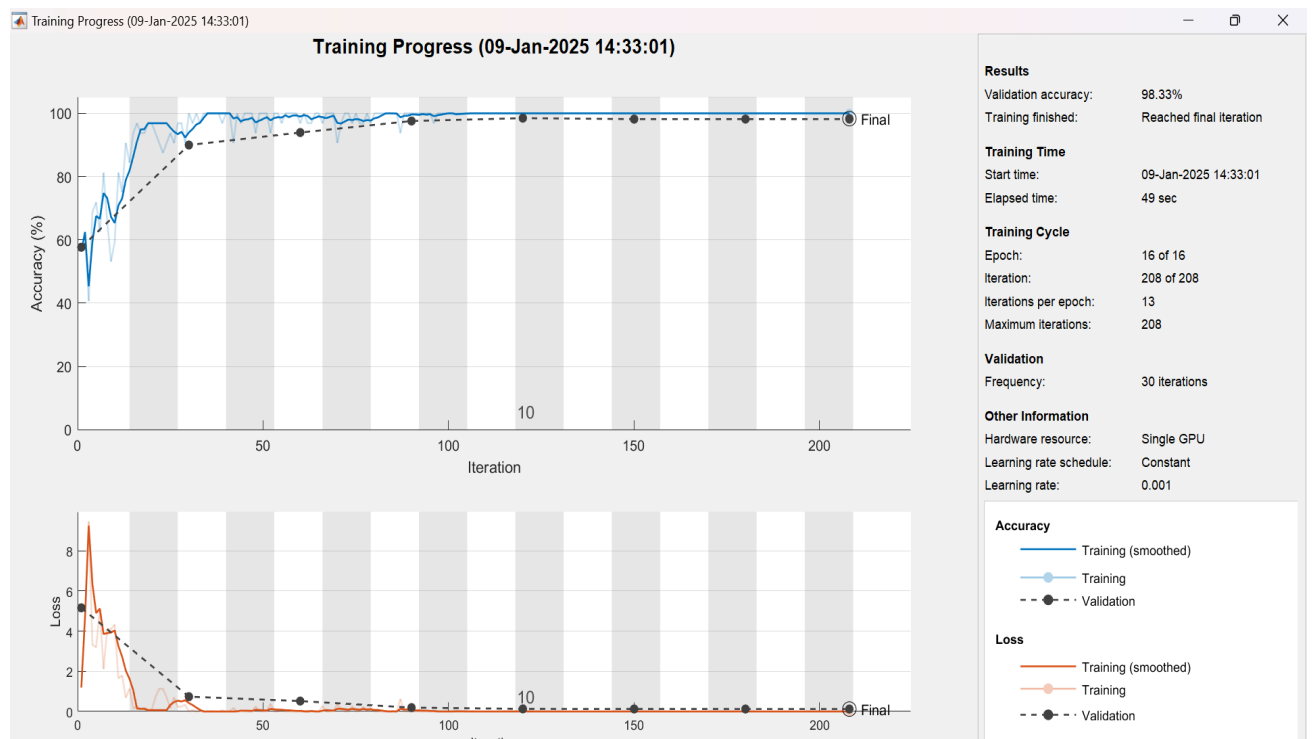


Figure 7 Training Process Training 40:60

Table 3 provides an in-depth analysis of the model's performance in predicting three classes, namely Benign, Malignant, and Normal based on different training and testing data divisions. In the 80:20 data division, the model achieved results with an accuracy of 99.54%. This shows that the model successfully predicted all test data correctly without any errors. This perfect performance indicates that the model has very good generalization when using a larger training data ratio than test data. In the 70:30 data division, the accuracy result was 98.78%, slightly lower than the previous ratio. The precision for the Benign class reached 94.74%, indicating that a small portion of the Benign class prediction data may be misclassified. Meanwhile, for the Malignant and Normal classes, the precision was very high, at 98.82% and 100% respectively. Recall for the Benign and Malignant classes remained perfect at 100%, but for the Normal class it decreased slightly to 96.80%. This indicates that the model is starting to experience a slight decline in performance, especially for the Normal class. The F1-score combining precision and recall remained high for all classes, although there was a slight decrease for the Normal class (98.37%). This shows that the model still performs very well despite having more testing data.

Table 3 Model's evaluation of proposed CNN layer

Split Data	Class	Accuracy	Precision	Recall	F1-Score
80:20	Benign	99.54%	100.00%	100.00%	100.00%
	Malignant		99.12%	99.07%	99.56%
	Normal		100.00%	100.00%	100.00%
70:30	Benign	98.78%	94.74%	100.00%	97.30%
	Malignant		98.82%	100.00%	99.41%
	Normal		100.00%	96.80%	98.37
60:40	Benign	95.66%	100.00%	64.58%	78.48%
	Malignant		96.97%	100.00%	98.46%
	Normal		93.18%	98.80%	95.91%
50:50	Benign	98.72%	100.00%	93.33%	96.55%
	Malignant		97.90%	100.00%	98.94%
	Normal		99.51%	98.56%	99.03%
40:60	Benign	98.33%	94.44%	94.44%	94.44%
	Malignant		99.70%	99.41%	99.55%
	Normal		97.61%	98.00%	97.80%


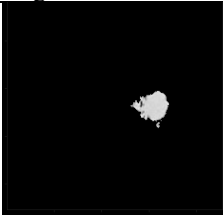

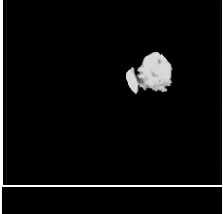
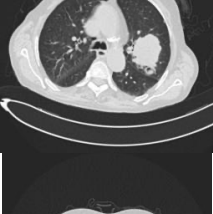
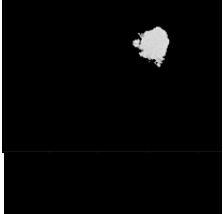
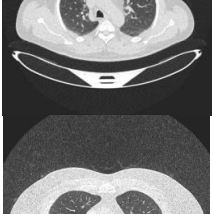
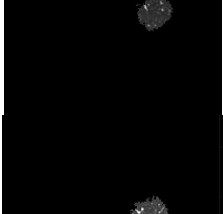
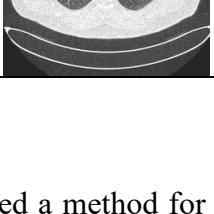
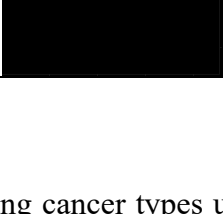
In the 60:40 data split, the accuracy dropped to 95.66%, indicating that the model's performance began to decline significantly. Precision for the Benign class remained perfect at 100%, while for Malignant and Normal it was 96.97% and 93.18%, respectively. However, recall for the Benign class dropped drastically to 64.58%, indicating that the model had difficulty recognizing most of the Benign class data. Recall for the Malignant and Normal classes remained high at 98.80% and 100%, respectively. The drastic decrease in recall for the Benign class caused the F1-score for that class to drop to 78.48%, although the F1-scores for the Malignant and Normal classes remained high. This indicates that the model began to lose its ability to recognize certain data patterns as the proportion of training data decreased. At 50:50 data split, accuracy increased again to 98.72%, almost approaching the 70:30 ratio.

Precision for the Benign, Malignant, and Normal classes were 100%, 97.90%, and 99.51%, respectively, indicating that most of the model's predictions were very accurate. Recall for the Benign and Malignant classes remained perfect at 100%, while for the Normal class it only decreased slightly to 98.56%. F1-scores for all classes were in the range of 96.55%-100%, indicating excellent performance across all metrics. The balanced data split between training and testing seems to provide

stability in the model's ability to recognize data patterns well. At 40:60 data split, accuracy was 98.33% with larger testing data compared to training data. Precision for the Benign, Malignant, and Normal classes were 94.44%, 99.70%, and 97.61%, respectively. Although the precision for the Benign class decreased slightly, the precision for the Malignant class remained very high, indicating that the model was very accurate in predicting the class. Recall for the Benign class also decreased to 94.44%, while for Malignant and Normal remained high at 99.41% and 97.61%. The F1-score for all classes remained high, although for the Benign class there was a slight decrease to 94.44%. This shows that the model is still able to maintain very good performance even though the amount of testing data is much larger, although there is a slight decrease in the ability to recognize patterns for certain classes. The conclusion from table 4.3 is that the performance of the model is greatly influenced by the ratio in dividing the training and testing data. At a ratio of 80:20, the model achieves perfect results in accuracy, precision, recall, and F1-score of 100% for all classes, indicating optimal generalization ability when the training data is larger than the testing data. However, as the proportion of testing data increases, such as at ratios of 70:30 and 60:40, there is a decrease in performance, especially for the Benign class. The decrease in recall for this class indicates that the model begins to have difficulty recognizing patterns in certain data when there was less training data. At a ratio of 50:50, the model stabilizes again with an accuracy of close to 99% and other metrics remain high. This indicates that the model is able to adapt well to a balanced data split. At a ratio of 40:60, despite the larger amount of testing data, the model still performs very well with an accuracy of over 98%. However, there is a slight decrease in precision, recall, and F1-score for the Benign class. Overall, the model performs well at all ratios, although its optimality decreases at ratios with less training data.

Table 4 shown the segmentation results on CT scan lung cancer dataset images that are grouped into two classes, namely Benign and Malignant. This table also shows two main columns, namely the original image table and the result table after segmentation. For the Malignant class, the original image shows nodules in the lungs that look denser and irregular in shape. In the segmentation results, the algorithm successfully extracts the nodule area from the background by providing binary image results, where the nodule area affected by lung cancer is displayed with a white object or only the part affected by cancer on a black background. In the Benign class, the original dataset image also shows nodules in the lungs, but with characteristics that tend to be more regular and homogeneous compared to the Malignant class. The segmentation results in this class also successfully separate the nodules from the background, with the nodule area still displayed in white or only displayed on the part affected by lung cancer on a black background. These results show that the segmentation method used is able to identify, extract, and determine the position of nodules from medical images well. This segmentation process is important for further analysis, such as measuring the size, shape, and texture of nodules which are indicators of lung cancer diagnosis. Overall, this table confirms the success of the segmentation method in separating nodule areas from medical images, thereby helping to improve the accuracy of radiological analysis and disease diagnosis. The difference in segmentation results between Malignant and Benign classes also reflects the potential to classify nodules based on their visual characteristics.

Table 4 Sample of Segmentation results

Class	Original Dataset	Segmentation result
Malignant		
Malignant		
Malignant		
Benign		
Benign		

5 Conclusion

Here, we had been developed a method for identifying lung cancer types using Convolutional Neural Network with Regionprops Segmentation technique. The dataset used in this study uses 1,294 CT scan images of lung cancer. In training and testing the model, several ratio divisions were carried out, namely 80:20, 70:30, 60:40, 50:50, and 40:60. Using the best ratio, namely 80:20, which produces an accuracy of 99.54%. For further research, it is recommended to use transfer learning techniques with more complex CNN architectures, such as ResNet or Efficientnet to improve performance on larger and more diverse datasets. In addition, expanding the types of medical data used, such as MRI or PET scans, can provide a more holistic approach to lung cancer detection. This research still has several shortcomings. To support method development and result optimization, future work suggests using hyperparameter optimization. To develop variational dataset patterns, it can be implemented using a Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) based on edge detection. Further research could use measurement patterns based on contours, gradients, or pixel grouping. Evaluating visual results based on thresholding values could also be a simple evaluation method.

Bibliography

- [1] E. R. Swenson, "The pathway for carbon dioxide: from tissues to lungs," in *On Oxygen*, Elsevier, 2025, pp. 211–228. doi: [10.1016/B978-0-443-21877-4.00008-5](https://doi.org/10.1016/B978-0-443-21877-4.00008-5).
- [2] S. Jana, P. Manjari, and I. Hyder, "Physiology of Respiration," in *Textbook of Veterinary Physiology*, Singapore: Springer Nature Singapore, 2023, pp. 171–192. doi: [10.1007/978-981-19-9410-4_7](https://doi.org/10.1007/978-981-19-9410-4_7).
- [3] R. Hasweh et al., "Radiological Differences in COVID-19 Related Lung Manifestations Between Smokers and Non-smokers: A Single-Center Retrospective Study in Jordan," *Cureus*, May 2023, doi: [10.7759/cureus.38437](https://doi.org/10.7759/cureus.38437).
- [4] R. Seleka, M. Petersen, and K. S. Mpolokeng, "Morphological variations of fissures, lobes, and hilar pattern of the lung in a select South African sample," *Surgical and Radiologic Anatomy*, vol. 46, no. 12, pp. 2005–2017, Oct. 2024, doi: [10.1007/s00276-024-03497-5](https://doi.org/10.1007/s00276-024-03497-5).
- [5] R. Maki et al., "Pulmonary vessels and bronchus anatomy of the left upper lobe," *Surg Today*, vol. 52, no. 4, pp. 550–558, Apr. 2022, doi: [10.1007/s00595-022-02471-1](https://doi.org/10.1007/s00595-022-02471-1).
- [6] I. P. Kamila, C. A. Sari, E. H. Rachmawanto, and N. R. D. Cahyo, "A Good Evaluation Based on Confusion Matrix for Lung Diseases Classification using Convolutional Neural Networks," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, p. 0240102, Dec. 2023, doi: [10.26877/asset.v6i1.17330](https://doi.org/10.26877/asset.v6i1.17330).
- [7] G. K. Bhatti, P. Pahwa, A. Gupta, U. Navik, and J. S. Bhatti, "Therapeutic Strategies Targeting Signaling Pathways in Lung Cancer," in *Targeting Cellular Signalling Pathways in Lung Diseases*, Singapore: Springer Singapore, 2021, pp. 217–239. doi: [10.1007/978-981-33-6827-9_9](https://doi.org/10.1007/978-981-33-6827-9_9).
- [8] A. Thurlapati et al., "Do the 2013 United States Preventive Services Task Force guidelines for lung cancer screening fail high-risk African American smokers? An institutional retrospective observational cohort study," *European Journal of Cancer Prevention*, vol. 30, no. 5, pp. 375–381, Sep. 2021, doi: [10.1097/CEJ.0000000000000652](https://doi.org/10.1097/CEJ.0000000000000652).
- [9] S. Punitha, T. Stephan, R. Kannan, M. Mahmud, M. S. Kaiser, and S. B. Belhaouari, "Detecting COVID-19 From Lung Computed Tomography Images: A Swarm Optimized Artificial Neural Network Approach," *IEEE Access*, vol. 11, pp. 12378–12393, 2023, doi: [10.1109/ACCESS.2023.3236812](https://doi.org/10.1109/ACCESS.2023.3236812).
- [10] A. Zotin, Y. Hamad, K. Simonov, and M. Kurako, "Lung boundary detection for chest X-ray images classification based on GLCM and probabilistic neural networks," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 1439–1448. doi: [10.1016/j.procs.2019.09.314](https://doi.org/10.1016/j.procs.2019.09.314).
- [11] M. K. Islam, M. M. Rahman, M. S. Ali, S. M. Mahim, and M. S. Miah, "Enhancing lung abnormalities detection and classification using a Deep Convolutional Neural Network and GRU with explainable AI: A promising approach for accurate diagnosis," *Machine Learning with Applications*, vol. 14, p. 100492, Dec. 2023, doi: [10.1016/j.mlwa.2023.100492](https://doi.org/10.1016/j.mlwa.2023.100492).
- [12] N. R. D. Cahyo and M. M. I. Al-Ghiffary, "An Image Processing Study: Image Enhancement, Image Segmentation, and Image Classification using Milkfish Freshness Images," *IJECAR International Journal of Engineering Computing Advanced Research*, vol. 1, no. 1, pp. 11–22, 2024.
- [13] F. Farhan, C. A. Sari, E. H. Rachmawanto, and N. R. D. Cahyo, "Mangrove Tree Species Classification Based on Leaf, Stem, and Seed Characteristics Using Convolutional Neural Networks with K-Folds Cross Validation Optimization," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 02303011, Oct. 2023, doi: [10.26877/asset.v5i3.17188](https://doi.org/10.26877/asset.v5i3.17188).
- [14] M. M. I. Al-Ghiffary, N. R. D. Cahyo, E. H. Rachmawanto, C. Irawan, and N. Hendriyanto, "Adaptive deep learning based on FaceNet convolutional neural network for facial expression

- recognition,” *Journal of Soft Computing*, vol. 05, no. 03, pp. 271–280, 2024, doi: <https://doi.org/10.52465/josce.v5i3.450>.
- [15] N. R. D. Cahyo, C. A. Sari, E. H. Rachmawanto, C. Jatmoko, R. R. A. Al-Jawry, and M. A. Alkhafaji, “A Comparison of Multi Class Support Vector Machine vs Deep Convolutional Neural Network for Brain Tumor Classification,” in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2023, pp. 358–363. doi: [10.1109/iSemantic59612.2023.10295336](https://doi.org/10.1109/iSemantic59612.2023.10295336).
- [16] H. F. Al-Yasriy, M. S. AL-Husieny, F. Y. Mohsen, E. A. Khalil, and Z. S. Hassan, “Diagnosis of Lung Cancer Based on CT Scans Using CNN,” *IOP Conf Ser Mater Sci Eng*, vol. 928, no. 2, p. 022035, Nov. 2020, doi: [10.1088/1757-899X/928/2/022035](https://doi.org/10.1088/1757-899X/928/2/022035).
- [17] A. Y. Saleh, C. K. Chin, V. Penshie, and H. R. H. Al-Absi, “Lung cancer medical images classification using hybrid CNN-SVM,” *International Journal of Advances in Intelligent Informatics*, vol. 7, no. 2, p. 151, Jul. 2021, doi: [10.26555/ijain.v7i2.317](https://doi.org/10.26555/ijain.v7i2.317).
- [18] F. A. Hermawati, “Sistem Deteksi Keganasan Kanker Paru-Paru pada CT Scan dengan Menggunakan Metode Mask Region-based Convolutional Neural Network (Mask R-CNN),” 2020. [Online]. Available: <https://www.researchgate.net/publication/355680874>
- [19] A. Al-Baset, R. Saabia, R. Majeed Azawi, and R. A. Kadhim, “Classification of lung cancer histology using CT images based on convolutional neural network-CNN,” *Int. J. Nonlinear Anal. Appl*, vol. 13, pp. 2008–6822, 2022, doi: [10.22075/ijnaa.2022.6847](https://doi.org/10.22075/ijnaa.2022.6847).
- [20] Z. UrRehman et al., “Effective lung nodule detection using deep CNN with dual attention mechanisms,” *Sci Rep*, vol. 14, no. 1, p. 3934, Feb. 2024, doi: [10.1038/s41598-024-51833-x](https://doi.org/10.1038/s41598-024-51833-x).
- [21] S. Bin Hulayyil, S. Li, and L. Xu, “Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security,” *Electronics (Basel)*, vol. 12, no. 18, p. 3927, Sep. 2023, doi: [10.3390/electronics12183927](https://doi.org/10.3390/electronics12183927).
- [22] A. Mozo, A. Karamchandani, L. de la Cal, S. Gómez-Canaval, A. Pastor, and L. Gifre, “A Machine-Learning-Based Cyberattack Detector for a Cloud-Based SDN Controller,” *Applied Sciences (Switzerland)*, vol. 13, no. 8, Apr. 2023, doi: [10.3390/app13084914](https://doi.org/10.3390/app13084914).
- [23] D. del-Pozo-Bueno, D. Kepaptsoglou, F. Peiró, and S. Estradé, “Comparative of machine learning classification strategies for electron energy loss spectroscopy: Support vector machines and artificial neural networks,” *Ultramicroscopy*, vol. 253, Nov. 2023, doi: [10.1016/j.ultramic.2023.113828](https://doi.org/10.1016/j.ultramic.2023.113828).
- [24] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Springer International Publishing, 2022. doi: [10.1007/978-3-030-89010-0](https://doi.org/10.1007/978-3-030-89010-0).
- [25] A. A. Khan, A. A. Laghari, and S. A. Awan, “Machine Learning in Computer Vision: A Review,” *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 32, pp. 1–11, 2021, doi: [10.4108/eai.21-4-2021.169418](https://doi.org/10.4108/eai.21-4-2021.169418).
- [26] Y. E. Almalki et al., “Isolated Convolutional-Neural-Network-Based Deep-Feature Extraction for Brain Tumor Classification Using Shallow Classifier,” *Diagnostics*, vol. 12, no. 8, Aug. 2022, doi: [10.3390/diagnostics12081793](https://doi.org/10.3390/diagnostics12081793).
- [27] A. P. Wibawa et al., “Decoding and preserving Indonesia’s iconic Keris via A CNN-based classification,” *Telematics and Informatics Reports*, vol. 13, Mar. 2024, doi: [10.1016/j.teler.2024.100120](https://doi.org/10.1016/j.teler.2024.100120).
- [28] N. E. W. Nugroho and A. Harjoko, “Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 221, Jul. 2021, doi: [10.22146/ijccs.66062](https://doi.org/10.22146/ijccs.66062).
- [29] M. Abdullah, M. Ahmad, and D. Han, “Facial Expression Recognition in Videos: An CNN-LSTM based Model for Video Classification,” in *2020 International Conference on Electronics, Information, and Communication, ICEIC 2020*, Institute of Electrical and Electronics Engineers Inc., Jan. 2020. doi: [10.1109/ICEIC49074.2020.9051332](https://doi.org/10.1109/ICEIC49074.2020.9051332).

- [30] M. N. Ab Wahab, A. Nazir, A. T. Z. Ren, M. H. M. Noor, M. F. Akbar, and A. S. A. Mohamed, "Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi," *IEEE Access*, vol. 9, pp. 134065–134080, 2021, doi: [10.1109/ACCESS.2021.3113337](https://doi.org/10.1109/ACCESS.2021.3113337).
- [31] M. M. I. Al-Ghiffary, C. A. Sari, E. H. Rachmawanto, N. M. Yacoob, N. R. D. Cahyo, and R. R. Ali, "Milkfish Freshness Classification Using Convolutional Neural Networks Based on Resnet50 Architecture," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 0230304, Oct. 2023, doi: [10.26877/asset.v5i3.17017](https://doi.org/10.26877/asset.v5i3.17017).
- [32] Q. A. Putra, C. A. Sari, E. H. Rachmawanto, N. R. D. Cahyo, E. Mulyanto, and M. A. Alkhafaji, "White Bread Mold Detection using K-Means Clustering Based on Grey Level Co-Occurrence Matrix and Region of Interest," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2023, pp. 376–381. doi: [10.1109/iSemantic59612.2023.10295369](https://doi.org/10.1109/iSemantic59612.2023.10295369).
- [33] V. Podgorelec, Š. Pečnik, and G. Vrbančič, "Classification of similar sports images using convolutional neural network with hyper-parameter optimization," *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–24, Dec. 2020, doi: [10.3390/app10238494](https://doi.org/10.3390/app10238494).
- [34] J. Zhang et al., "Deep Learning for Microfluidic-Assisted Caenorhabditis elegans Multi-Parameter Identification Using YOLOv7," *Micromachines (Basel)*, vol. 14, no. 7, p. 1339, Jun. 2023, doi: [10.3390/mi14071339](https://doi.org/10.3390/mi14071339).
- [35] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How do adam and training strategies help bnns optimization," in *International conference on machine learning*, PMLR, 2021, pp. 6936–6946.
- [36] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput Appl*, vol. 35, no. 23, pp. 17095–17112, Aug. 2023, doi: [10.1007/s00521-023-08568-z](https://doi.org/10.1007/s00521-023-08568-z).
- [37] A. Kumar, S. Sarkar, and C. Pradhan, "Malaria Disease Detection Using CNN Technique with SGD, RMSprop and ADAM Optimizers," in *Deep Learning Techniques for Biomedical and Health Informatics*, S. Dash, B. R. Acharya, M. Mittal, A. Abraham, and A. Kelemen, Eds., Cham: Springer International Publishing, 2020, pp. 211–230. doi: [10.1007/978-3-030-33966-1_11](https://doi.org/10.1007/978-3-030-33966-1_11).
- [38] J. Isański, S. Kupiński, M. Leszkowicz, W. Andrzejewski, and A. Bykowski, "Outdoor fitness activity under COVID-19 restrictions. Wearable devices, images, and posts as sources of information on users," *Człowiek i Społeczeństwo*, vol. 55, pp. 115–139, Jul. 2023, doi: [10.14746/cis.2023.55.7](https://doi.org/10.14746/cis.2023.55.7).
- [39] A. Theissler, M. Thomas, M. Burch, and F. Gerschner, "ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices," *Knowl Based Syst*, vol. 247, Jul. 2022, doi: [10.1016/j.knosys.2022.108651](https://doi.org/10.1016/j.knosys.2022.108651).
- [40] I. Markoulidakis and G. Markoulidakis, "Probabilistic Confusion Matrix: A Novel Method for Machine Learning Algorithm Generalized Performance Analysis," *Technologies (Basel)*, vol. 12, no. 7, p. 113, Jul. 2024, doi: [10.3390/technologies12070113](https://doi.org/10.3390/technologies12070113).