Published online on the journal's website: https://jes-tm.org/index.php/jestm/index

Journal of Engineering Science and Technology Management

| ISSN (Online) 2828 -7886 |

Article

DICOM Image Analysis for Lung Cancer Detection Using Convolutional Neural Network (CNN)

M. Arib Insanul Kamil[1a]✉, Romi Mulyadi[2,b]

[1.2] Electro-medical Engineering Technology, Faculty of Technology, Al Insyira Institute of Health and Technology, Pekanbaru, Indonesia

DOI: 10.31004/jestm.v5i2.290

E-mail: aribkamil5@gmail.com (Corresponding author)

ARTICLE INFORMATION

Volume 5 Issue 2
Received: 20 July 2025
Accepted: 02 September 2025
Publish Online: 05 September 2025
On line: at https:JESTM.org/

Keywords

Lung cancer,
CT Scan,
DICOM,
Deep Learning,
Convolutional Neural Network (CNN),
Early detection

ABSTRACT

Lung cancer remains the leading cause of cancer-related deaths worldwide, with the highest burden in Asia, including Indonesia. Early detection is critical, yet access to radiology services is often limited by infrastructure, cost, and a shortage of trained specialists. Recent advances in artificial intelligence, particularly Convolutional Neural Networks (CNNs), offer promising solutions for automated image-based diagnosis. This study aims to analyze the effectiveness of CNN in detecting lung cancer from CT scan images in DICOM format. A dataset consisting of lung CT images from Kaggle and local hospitals was preprocessed through Gaussian blur filtering, segmentation, and pixel normalization before model training. Images were classified into two categories: cancer and non-cancer. The CNN architecture was trained and validated with an 80:20 split ratio, and model performance was assessed using accuracy, precision, recall, and F1-score. The experimental results show that the proposed CNN model achieved an accuracy of 88.27%, precision of 88.96%, recall of 97.43%, and an F1-score of 92.98%. The high recall value indicates the model’s strong ability to minimize false negatives, which is essential for clinical application. Performance graphs demonstrated stable accuracy and loss across training and validation sets, suggesting minimal overfitting.In conclusion, the developed CNN model demonstrates strong potential as a supportive diagnostic tool for early lung cancer detection, particularly in resource-limited healthcare settings. Its integration into radiology workflows may accelerate screening processes and improve clinical decision-making

1. Introduction

Lung cancer remains the leading cause of cancer death worldwide. Data from the Global Cancer Observatory (GCO) shows that in 2022, there were approximately 2.48 million new cases, representing approximately 12.4% (≈ 1 in 8) of all malignancies, and 1.8 million deaths related to this disease. These figures confirm that lung cancer remains a top priority for global cancer control programs.(Sung et al., 2021).

The disease burden is heaviest in Asia: >60% of global lung cancer cases and deaths occur on this continent, influenced by its large population, high smoking prevalence among men, and worsening air pollution. The clinical and economic consequences make early detection in Asia, including Indonesia, a strategic challenge.(Ardila et al., 2018).

In Indonesia, the WHO cancer profile places lung cancer as the highest incidence in men and the fifth most common in women. A study by the Ministry of Health's Research and Development Agency, cited by Febriani et al. (2018), reported 34.2% of new cases with a mortality rate of 30%. The majority of patients present at an advanced stage, when curative therapy is difficult to achieve and life expectancy declines sharply.(Sasikala et al., 2018).

Imaging technology, particularly computed tomography (CT) scanning with DICOM file format, is the gold standard for assessing pulmonary nodules. However, this examination requires expensive infrastructure, long interpretation times, and the expertise of trained radiologists, making its availability limited, especially in areas with limited radiology resources.(Hernández-rodríguez, 2022).

The development of deep learning-based artificial intelligence (AI) opens new opportunities. Convolutional Neural Network (CNN) architecture is capable of extracting spatial-temporal patterns in stacked 3-dimensional CT slices and automatically classifying nodules. Various studies, including 3D DL-CNN on CT-DICOM datasets (accuracy ≥ 93%), conventional 3D-CNN (accuracy ≈ 95%), and the application of Mask R-CNN for nodule segmentation, have demonstrated high performance and the potential for saving radiologists' reading time. Therefore, the integration of CNN algorithms into DICOM workflows is expected to increase screening sensitivity, accelerate diagnosis, and reduce the cost of lung cancer services in Indonesia.(Mastouri, 2020)This study aims to analyze the effectiveness of the Convolutional Neural Networks (CNN) model in detecting lung cancer in CT scan images and can assist medical personnel in more effective diagnosis.

2. Literature Review

2.1 Lung cancer

Lung cancer, broadly defined, encompasses all malignant disorders of the lungs, whether originating in the lung tissue itself (primary) or metastasizing from other organs. In clinical practice, primary lung cancer is defined as a malignant neoplasm that develops from bronchial epithelial cells (bronchial carcinoma). Although its etiology is not entirely clear, long-term exposure to carcinogenic agents through inhalation is considered a major factor, alongside other factors such as immune status, genetic factors, and so on. Various studies also show that smoking habits are closely linked to the development of lung cancer.(Meaden et al., 2019).

2.2 Image processing

"Image processing refers to a set of techniques used to manipulate or analyze images to obtain information, improve visual quality, or prepare data for further analysis. It involves a series of operations such as image enhancement, image restoration, and feature extraction performed on digital images. These operations aim to improve image quality or extract meaningful data from images, which can then be used in various applications such as medical imaging, remote sensing, industrial inspection, and computer vision. The fundamental goal of image processing is to transform an input image into an enhanced or transformed output image, which can be used for a specific task. The process typically involves manipulating pixel values, applying filters, or using mathematical transformations that can bring out interesting features or reduce unwanted noise" (Gonzales et al., 2008).

2.3 Deep learning

Deep learning can be viewed as a subcategory of machine learning, focusing on the design of algorithms based on deep neural networks. This approach allows computer systems to independently identify and learn complex patterns from large, unstructured data sets, such as images, audio, text, and video.(Alfarizi et al., 2023).

*"Deep Learning*have achieved remarkable success in a wide range of applications, from computer vision and natural language processing to autonomous systems and medical diagnostics. These models, inspired by the structure and functioning of the human brain, build multi-level representations of data, enabling them to handle highly complex tasks with remarkable precision. Leveraging large amounts of labeled data and significant computing power, they have become one of the most important areas of artificial intelligence research" (Goodfellow et al., 2016).

2.4 Visual Studio Code

Visual Studio Code is a code editor application developed by Microsoft and can run on various operating systems such as Windows, Linux, and macOS. This editor offers various features such as debugging capabilities, direct integration with code management services like Git and GitHub, and facilities for syntax highlighting and code auto-completion. Furthermore, users can customize the appearance and functionality of this editor according to their needs through theme settings, keyboard shortcuts, and additional extensions to increase programming productivity (Tensorflow Team, 2021).

2.5 Python

Python has become the language of choice for software developers, data scientists, and researchers thanks to its ease of use, concise and clear syntax, and its ability to support a variety of paradigms, from procedural to object-oriented to functional. Python's main advantages are its highly active community and the abundance of libraries available for a wide range of applications, from data analysis and machine learning to web development.

Python also has the ability to integrate with other languages and efficiently perform various computationally demanding tasks. Furthermore, Python is widely used in academia and industry, thanks to its ability to process large amounts of data and build prototypes quickly. With its wide range of applications and support for various platforms, Python has become a top choice for application development across a wide range of disciplines (Van Rossum et al., 2020).

"A Python library is a collection of modules and code libraries designed to simplify and accelerate application development with Python. With over thousands of libraries available, Python enables developers to tackle a wide range of complex tasks, from data analysis and machine learning to web development and system automation. Some of the most well-known libraries include NumPy for numerical computation, Pandas for data manipulation, Matplotlib for data visualization, and TensorFlow for machine learning. The main advantage of using Python libraries is that they allow developers to leverage existing solutions, allowing them to focus on solving core problems without having to write code from scratch. Using these libraries also speeds up development time, increases efficiency, and reduces the likelihood of errors in applications." (G. Van Rossum., 2019)

2.6 Grayscale image

A grayscale image is a type of image that consists solely of variations of gray without any other colors. Each pixel in this image has an intensity value indicating its brightness, ranging from black as the lowest value to white as the highest value. Generally, grayscale images present various shades of gray indicating varying levels of brightness, without the color information found in color images.(Nasha Hikmatia AE & Zul, 2021).

In image processing, the initial step often taken is to convert a color image to grayscale. The goal is to simplify the image representation and make processing more efficient. Color images are composed of three matrix layers: red (R), green (G), and blue (B). In the analysis process, these three layers must be considered simultaneously, which means each operation must be performed on each layer.

To reduce this complexity, these three layers are usually combined into a single matrix in the form of a grayscale image, thus speeding up and simplifying the subsequent processing process" (Rina Candra nur et al, 2011)

2.7 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN), or ConvNet, is an advanced extension of the Multilayer Perceptron (MLP) architecture specifically designed for the analysis of two-dimensional data. Within the framework of lung cancer detection, CNNs operate through a sequence of computational layers: convolutional layers extract salient features from CT scan images, pooling layers reduce the dimensionality of these features while preserving critical information, and fully connected layers perform the final classification based on the extracted representations (Ilham & Rochmawati, 2020)

3. Research Methodology

The techniques through which the work was carried out are as presented below.

3.1 Types of research

This study used an experimental approach with image analysis using a machine learning algorithm. The experiment was conducted to test a Convolutional Neural Network (CNN) model for automatically detecting lung cancer from CT scan images. The experimental process included data collection, data processing, model training, and analysis of the results.

3.2 Flow chart

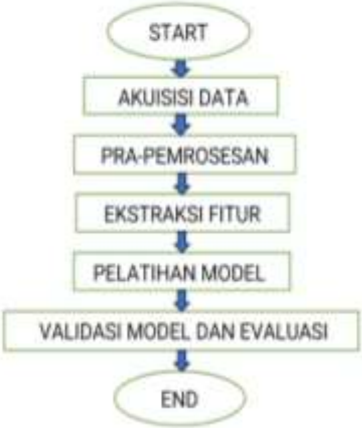

Figure 1. Research flowchart

3.3 Data analysis

This study used a dataset of lung medical images obtained from Kaggle and hospitals, converted from DICOM in standard format (.jpg/.png). The data were divided into two classes: images with cancer indications ('yes') and images without cancer ('no').

Before model training, preprocessing was performed, including noise reduction with Gaussian Blur, image segmentation to separate lung areas from the background, and pixel normalization for consistency. Image features were then automatically extracted using a Convolutional Neural Network (CNN), which captures textural and spatial characteristics, including the shape, size, and pattern of lesions.

The model was tested using individual CT scan images to generate predictions in the form of labels ('cancer' or 'not cancer') and their probabilities. The evaluation was conducted by splitting the data into 80% training and 20% testing data, using accuracy, precision, recall, and F1-score metrics to assess the model's overall performance.

4. Results and Discussion

This research aims to create and evaluate a Convolutional Neural Network model for lung cancer identification from CT scan images in DICOM format. The model is built through several key stages: data acquisition, preprocessing, feature extraction, model training and validation, and performance evaluation. The details are as follows.

a) Data Acquisition
The image data was taken from two sources: the Kaggle website, which provides a dataset of CT scan images labeled with cancer and non-cancer labels, and from Arifin Achmad Regional Hospital in Pekanbaru for system testing on real-world data. The images were collected in DICOM format, a common standard in medical imaging.

b) Preprocessing
Preprocessing steps were performed to improve image quality, reduce noise using Gaussian Blur, segment lung areas, and normalize pixel intensity scales. This aims to ensure that the input to the CNN model is consistent and high-quality.

c) Feature Extraction
The CNN model automatically extracts spatial and textural features. Features such as contrast, contour, and lesion size are processed from grayscale images.
The CNN structure consists of several convolutional, pooling, and fully connected layers that work sequentially to extract important information from the image.

d) Model Training and Validation.
The model was trained on a preprocessed dataset, using a ratio of 80% training and 20% testing. During training, parameters were adjusted to accurately recognize cancer patterns. Loss and accuracy visualizations were displayed using Matplotlib.

e) Performance Evaluation
The model was evaluated using accuracy, precision, recall, and F1-score metrics. Initial results indicate the model has a high ability to accurately detect lung cancer. The balance between precision and recall values demonstrates that the model not only provides accurate predictions but is also responsive in identifying actual cancer cases.

It is hoped that this model will be able to function as a support system in the decision-making process for medical personnel, especially in areas with limited radiology experts.

4.1 Data sheet image collection

This section is the dataset collection. The images used in this study come from two main sources: the Kaggle website and Prof. Dr. Tabrani Pekanbaru Hospital. Kaggle provides a dataset of lung CT scan images in jpg/png format that have been labeled between images indicating cancer and images of healthy lungs. This dataset is very useful for the initial model training process because it has a large and varied data volume.

Meanwhile, data from Prof. Dr. Tabrani Hospital in Pekanbaru was collected for real-world model testing. The hospital data consisted of patient CT scan images converted from DICOM to .jpg or .png to facilitate processing using Python and related libraries such as TensorFlow and Keras. This data provides added value to the study because it reflects actual conditions in the field and helps assess the model's generalizability to real clinical data.

Before being used in model training and testing, all data were classified into two classes: lung images with cancer (positive) and lung images without cancer (negative). The labeling process was performed manually based on medical information from DICOM metadata and confirmation from medical personnel. With this approach, the dataset used in this study is of high quality and highly relevant to the research objective, namely to develop a lung cancer detection model based on CT scan images using the Convolutional Neural Network (CNN) method.

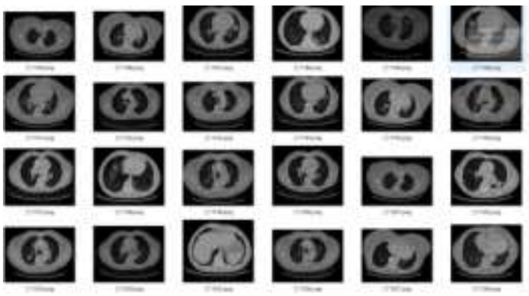

Figure 2. Cancer image data sheet

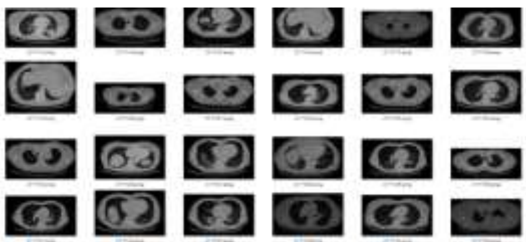

Figure 3. Normal image data sheet

4.2 Pre-Processing

Image pre-processing is a critical stage in the medical image analysis pipeline because the quality of the input data greatly influences the model's performance in detecting lung cancer.
In this study, CT Scan image data obtained from Kaggle and Prof. Dr. Tabrani Pekanbaru Hospital first went through a pre-processing process so that the image was uniform, free from visual interference (noise), and optimal

for training the Convolutional Neural Network (CNN) model.

1. Loading Cancer Image Datasheet. This code is used to load images from a folder containing positive data (containing cancer). Using a for loop, each image file in the path_yes folder is retrieved, then read and converted to grayscale mode and resized to 128x128 pixels. Next, the image is converted to an array and normalized to a scale of 0–1. This image array is stored in the data list, while the label 1 (indicating cancer) is inserted into the labels list as a class marker.

```
for file in os.listdir(path_yes):
    img_path = os.path.join(path_yes, file)
    img = load_img(img_path, color_mode='grayscale', target_size=(128, 128))
    img_array = img_to_array(img) / 255.0
    data.append(img_array)
    labels.append(1)
```

Figure 4. Cancer Datasheet Loading Code

2. Loading Normal Image Datasheet. This code is used to load images from a folder containing negative data (not containing cancer). Using a for loop, each image file in the path_No folder is retrieved, then read and converted to grayscale mode and resized to 128x128 pixels. Next, the image is converted to an array and normalized to a scale of 0–1. This image array is stored in the data list, while the label 1 (indicating cancer) is inserted into the labels list as a class marker.

```
for file in os.listdir(path_no):
    img_path = os.path.join(path_no, file)
    img = load_img(img_path, color_mode='grayscale', target_size=(128, 128))
    img_array = img_to_array(img) / 255.0
    data.append(img_array)
    labels.append(0)
```

Figure 5. Normal Datasheet Loading Code

3. Numpy Array Conversion After all images and labels are collected in a list, they are converted to NumPy arrays using np.array(). This conversion aims to simplify the computational process and ensure compatibility with machine learning libraries like TensorFlow, as the NumPy array format is more efficient and can be directly used as input for model training.

```
data = np.array(data)
labels = np.array(labels)
```

Figure 6. Numpy Array Conversion

4. Loading image data generator.This code creates an ImageDataGenerator object from Keras, which is used for preprocessing and augmenting image data. The rescale=1./255 parameter normalizes pixel values from 0–255 to a 0–1 scale, which is essential for model stability during training. Meanwhile, validation_split=0.2 means that 20% of the data will be used as validation data, and the remaining 80% for training. This helps evaluate model performance during training without the need for manual dataset splitting.

```
datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)
```

Figure 7. Image Generator Code

5. Reads images in a folder for training. Flow_from_directory will read all images from the data_path folder, resize each image to 128×128 pixels, and convert them to grayscale mode. Pixel values are automatically normalized (rescale=1./255) according to the ImageDataGenerator settings. With batch size=32, data will be fetched in batches of 32 images at a time for efficient training. The 'binary' class mode indicates that this is a two-class classification problem (cancer vs. non-cancer). This generator takes 80% of the training data as specified.

```
train_data = datagen.flow_from_directory(
    data_path,
    target_size=(128, 128),
    color_mode='grayscale',
    batch_size=32,
    class_mode='binary',
    subset='training'
)
```

Figure 8. Code Reading Training Folder Image

6. Reading the Code Validation folder. The Code Validation folder is used to generate validation data with imagedatagenerator. Data is taken from the data_path, the images are converted to 128x128 pixels, converted to grayscale, and then normalized to the 0–1 range. With batch_size=32, each batch contains 32 images. The class_mode='binary' parameter indicates a two-class classification, while subset='validation' takes 20% of the data as validation data.

Figure 9. Reading the Validation folder

4.2 Feature extraction

Feature extraction is the process of extracting important information from medical images so they can be recognized by a machine learning model. In this study, this process is automated by a CNN architecture through convolution and pooling layers. The CNN model is used to detect lung cancer from 128x128 pixel grayscale CT scan images. The model has three convolution layers to capture important features, then the data is flattened and processed in a dense layer for classification. Dropout is used to prevent overfitting, and the output uses sigmoid activation to predict the presence or absence of cancer in a binary manner.

4.3 Model Training

Convolutional Neural Network (CNN) training uses preprocessed image data. The goal is for the model to recognize lung patterns and distinguish between normal and cancerous conditions.

1. CNN Model Compilation.
   The model was compiled with a binary_crossentropy loss due to the binary nature of the classification. Optimization used a learning rate of 0.0001 for stability, and the accuracy metric was used to assess the accuracy of the predictions.
2. Training Process.
   Training was performed using the model. Fit function using 10 epochs of training and validation data. The steps_per_epoch and validation_steps parameters were adjusted to optimize the number of batches. During training, the model weights were updated to reduce loss and improve accuracy in detecting lung cancer.

4.4 Model validation and evaluation

After training is complete, the model is tested on validation data to assess its performance. Evaluation using the model.evaluate() function produces two metrics: validation loss and validation accuracy. Loss indicates the level of prediction error, while accuracy indicates the accuracy of image classification. These results are used to assess how well the model recognizes lung cancer patterns in new data.

Figure 10. Model validation result code

4.5 Calculation of confusion matrix

The analysis was conducted by examining accuracy, loss, and confusion matrix values to assess the model's ability to distinguish cancerous and non-cancerous images. This evaluation aimed to determine the model's effectiveness and accuracy in classifying the test data.

Confusion Matrix

CNN model training results: Based on evaluation using a confusion matrix, the model successfully classified 266 tumor images correctly (true positives) and 35 non-tumor images correctly (true negatives). However, there were 33 false positives (non-tumors predicted as tumors) and 7 false negatives (tumors not detected). These results indicate that the model has high sensitivity for detection.

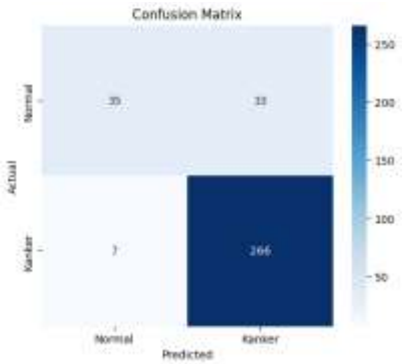

Figure 10. Confusion matrix results

Manual calculation of confusion matrix

Based on the evaluation results, the CNN model demonstrated quite good performance with an accuracy of 89.74%, precision of 90.48%, recall of 97.43%, and F1-score of 93.80%. The high accuracy value indicates that most of the model's predictions match the actual labels, while precision confirms that most of the images predicted as cancer are indeed positive. The recall of 97.43% indicates the model's excellent ability to recognize almost all tumor images, so the risk of cancer cases going undetected is very small. The high F1-score value also shows a balance between precision and recall, which means the model not only focuses on detecting cancer, but also attempts to reduce misclassification of non-cancerous images. With these results, the CNN model can be considered quite reliable and has great potential for application in early lung cancer detection systems as an accurate diagnostic tool.

Table 1. Manual calculation of confusion matrix

No	Matrix	Formula	Mark (%)
1	Accuracy	$akurasi = \frac{Tp + Tn}{Tp + Tn + Fp + Fn}$	88.27
2	Precision	$Presisi = \frac{Tp}{Tp + Fp}$	88.96
3	Recall	$Recall = \frac{Tp}{Tp + Fn}$	97.43
4	F1-Score	$F1 - Score = 2 \times \frac{Precision + Recall}{Precision + Recall}$	92.98

Accuracy model

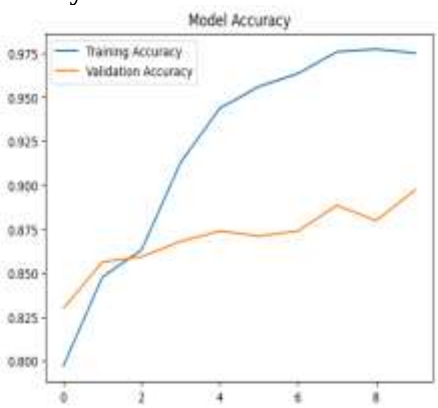

Figure 11. Accuracy graph

The graph in the figure shows the progress in CNN model accuracy during the 10 epochs of training. Accuracy on the training data increased from 80% to nearly 98%, while validation accuracy also increased from around 83% to over 90% at the end of the epoch. This consistent increase indicates the model's ability to adequately understand data patterns. Because the training and validation accuracies differ only slightly, the model does not appear to be significantly overfit and can be applied to new data with satisfactory results.

Loss model

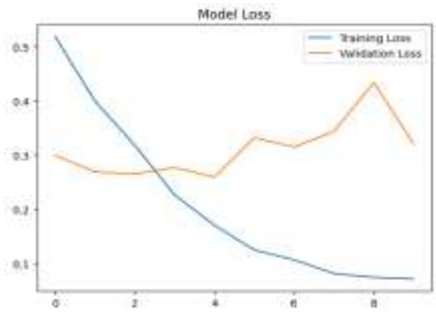

Figure 12. Loss mode graph

The Model Loss graph shows the changes in the training and validation loss values of the CNN model. The training loss consistently decreased from around 0.52 to less than 0.07, indicating that the model successfully learned from the training data. Meanwhile, the validation loss showed slight fluctuations, especially after the fifth epoch, which could indicate the onset of overfitting. Nevertheless, the validation loss remained relatively low, indicating that the model is still able to generalize well to previously unseen data.

4.6 Hospital Image Detection

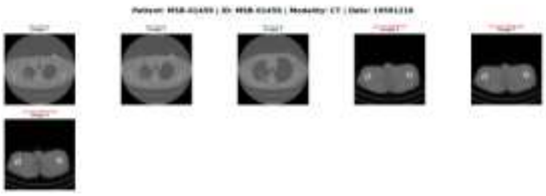

Figure 13. Lung cancer detection results

Based on the CT scan image processing results of patient ID MSB-01459, six cross-sectional images of the patient's body were classified. Each image was analyzed using a deep learning-based cancer detection model. These CT scan images were taken from Tabrani Hospital as part of the test data for the automated classification system. The detection results for each image are as follows:

1. Image 1 - 3: No cancer detected. The image shows lung tissue structure with normal density and air distribution patterns, with no indication of suspicious masses or lesions.
2. Image 4 - 6: Cancer Detected. The image shows an abnormal area in the soft tissue surrounding the lower extremity (thigh), characterized by differences in tissue density and contrast to surrounding structures, indicating the possible presence of a tumor mass.

Thus, of the total six CT images analyzed, three images (Images 4, 5, and 6) showed indications of cancer, while the other three images (Images 1, 2, and 3) were declared free of indications of cancer.

5. Conclusion

Based on the entire series of studies, the developed Convolutional Neural Network (CNN) model proved effective in distinguishing between lung CT scan images with and without cancer. The model's performance showed an overall accuracy of 88.27%, supported by a precision value of 88.96% and a recall of 97.43%, indicating a high lung cancer detection capability and minimal false negative errors. Furthermore, the stable training and validation curve patterns until the final epoch indicated no significant overfitting, so the model has good generalization to the test data. Therefore, this classification system is worthy of consideration as an aid in the early diagnosis of lung cancer based on CT scans.

Acknowledgement

The author expresses his deepest gratitude to his supervisors who provided direction, guidance, and motivation throughout the preparation of this research. He also expresses his gratitude to the university for providing support, facilities, and a conducive academic environment, enabling the smooth running of this research.

References

Adnan, Suhartini, & Kusbiantoro, B. (2013). Identification of Rice Varieties Based on Color and Surface Texture Using Digital Image Processing and Artificial Neural Networks. Food Crop Agricultural Research, 32(2), 91–97.

Alfarizi, MRS, Al-farish, MZ, Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Using Python as a Programming Language for Machine Learning and Deep Learning. Scientific Work of Tauhid Students (Karimah Tauhid), 2(1), 1–6.

Alfath Daryl Alhajir, Yisti Vita Via, & Wahyu Syaifullah Jauharis Saputra. (2021). Rice Object and Foreign Object Detection System Based on Keras and Google Colab. Journal of Informatics and Information Systems, 2(3), 580–586. https://doi.org/10.33005/jifosi.v2i3.369

Ardila, D., Kiraly, A.P., Bharadwaj, S., Choi, B., Reicher, J.J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., & Naidich, D.P. (2018). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine. https://doi.org/10.1038/s41591-019-0447-x

Brawijaya, U., Rachmatika, IS, Muflikhah, L., & Setiawan, BD (2017). Faculty of Computer Science: Detecting Mutations in Lung Cancer Through CT-Scan Images: Application of Convolutional Neural Networks (CNN) Algorithm Model and Adam Optimizer. 1(1), 2548–2964. http://j-ptiik.ub.ac.id

Dartiko, F., Pradana, RJ, Sari, RE, Syahputra, W., & Oktoeberza, WK (2024). CNN-Based Skin Cancer Classification with Hybrid Preprocessing Method. Medika Teknika: Indonesian Journal of Electromedical Engineering, 5(2), 124–132.https://doi.org/10.18196/mt.v5i2.22675

Fadli, C., & Desmulyati, D. (2021). Implementation of Face Detection Calculation Using the Haar Cascade Classifier Method. National Journal of Computing and Information Technology (JNKTI), 4(6), 535–542. https://doi.org/10.32672/jnkti.v4i6.3721

Febriani, A., & Furqon, A. (2020). Lung Cancer Metastasis. Journal of Respiration, 4(3), 94. https://doi.org/10.20473/jr.v4-i.3.2018.94-101

Hernández-rodríguez, J. (2022). Convolutional Neural Networks for Multi-scale Lung Nodule Classification in CT : Influence of Hyperparameter Tuning on Performance. 11(1), 297–306. https://doi.org/10.18421/TEM111

I Komang Setia Buana. (2020). Implementation of a Speech to Text Application to Make it Easier for Journalists to Record Interviews with Python. Journal of Systems and Informatics (JSI), 14(2), 135–142. https://doi.org/10.30864/jsi.v14i2.293

Ilham, F., & Rochmawati, N. (2020). Transliteration of Handwritten Javanese Script into Latin Script Using CNN. Journal of Informatics and Computer Science (JINACS), 1(04), 200–208. https://doi.org/10.26740/jinacs.v1n04.p200-208

Mastouri, R. (2020). A bilinear convolutional neural network for lung nodules classification on CT images. Dl.

Meaden, C.W., Kashani, J.S., & Vetrano, S. (2019). Pulmonary Edema Occurring after Nitric Acid Exposure. Case Reports in Emergency Medicine, 2019(1), 1–4. https://doi.org/10.1155/2019/9303170

Mulyawan, H., Samsono, MZH, & Setiawardhana. (2011). Image-Based Object Identification and Tracking. Real-Time Image Processing-Based Object Identification and Tracking, 1–5. http://repo.pens.ac.id/1324/1/Paper_TA_MBAH.pdf

Munandar, A. (2020). Programming language. THEMATICS | Technology Management and Informatics Research Journals TEMATICS | Technology Management and Informatics Research Journals, 4(2), 12.

Nasha Hikmatia AE, & Zul, MI (2021). An Android-based Indonesian Sign Language to Speech Translator Application using Tensorflow. Applied Computer Journal, 7(1), 74–83. https://doi.org/10.35143/jkt.v7i1.4629

Phiadelvira, BY (2021). Classification of cervical cancer based on colposcopy images using the Convolutional Neural Network (CNN) Alexnet Model. In JIKO (Jurnal Informatika dan Komputer) (Vol. 4, Issue 1).

Adnan, Suhartini, & Kusbiantoro, B. (2013). Identification of Rice Varieties Based on Color and Surface Texture Using Digital Image Processing and Artificial Neural Networks. Food Crop Agricultural Research, 32(2), 91–97.

Alfarizi, MRS, Al-farish, MZ, Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Using Python as a Programming Language for Machine Learning and Deep Learning. Scientific Work of Tauhid Students (Karimah Tauhid), 2(1), 1–6.

Alfath Daryl Alhajir, Yisti Vita Via, & Wahyu Syaifullah Jauharis Saputra. (2021). Rice Object and Foreign Object Detection System Based on Keras and Google Colab. Journal of Informatics and Information Systems, 2(3), 580–586. https://doi.org/10.33005/jifosi.v2i3.369

Ardila, D., Kiraly, A.P., Bharadwaj, S., Choi, B., Reicher, J.J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., & Naidich, D.P. (2018). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine. https://doi.org/10.1038/s41591-019-0447-x

Brawijaya, U., Rachmatika, IS, Muflikhah, L., & Setiawan, BD (2017). Faculty of Computer Science: Detecting Mutations in Lung Cancer Through CT-Scan Images: Application of Convolutional Neural Networks (CNN) Algorithm Model and Adam Optimizer. 1(1), 2548–2964. http://j-ptiik.ub.ac.id

Dartiko, F., Pradana, RJ, Sari, RE, Syahputra, W., & Oktoeberza, WK (2024). CNN-Based Skin Cancer Classification with Hybrid Preprocessing Method. Medika Teknika: Indonesian Journal of Electromedical Engineering, 5(2), 124–132. https://doi.org/10.18196/mt.v5i2.22675

Fadli, C., & Desmulyati, D. (2021). Implementation of Face Detection Calculation Using the Haar Cascade Classifier Method. National Journal of Computing and Information Technology (JNKTI), 4(6), 535–542. https://doi.org/10.32672/jnkti.v4i6.3721

Febriani, A., & Furqon, A. (2020). Lung Cancer Metastasis. Journal of Respiration, 4(3), 94. https://doi.org/10.20473/jr.v4-i.3.2018.94-101

Hernández-rodríguez, J. (2022). Convolutional Neural Networks for Multi-scale Lung Nodule Classification in CT : Influence of Hyperparameter Tuning on Performance. 11(1), 297–306. https://doi.org/10.18421/TEM111

I Komang Setia Buana. (2020). Implementation of a Speech to Text Application to Make it Easier for Journalists to Record Interviews with Python. Journal of Systems and Informatics (JSI), 14(2), 135–142. https://doi.org/10.30864/jsi.v14i2.293

Ilham, F., & Rochmawati, N. (2020). Transliteration of Handwritten Javanese Script into Latin Script Using CNN. Journal of Informatics and Computer Science (JINACS), 1(04), 200–208. https://doi.org/10.26740/jinacs.v1n04.p200-208

Mastouri, R. (2020). A bilinear convolutional neural network for lung nodules classification on CT images. Dl.

Meaden, C.W., Kashani, J.S., & Vetrano, S. (2019). Pulmonary Edema Occurring after Nitric Acid Exposure. Case Reports in Emergency Medicine, 2019(1), 1–4. https://doi.org/10.1155/2019/9303170

Mulyawan, H., Samsono, MZH, & Setiawardhana. (2011). Image-Based Object Identification and Tracking. Real-Time Image Processing-Based Object Identification and Tracking, 1–5. http://repo.pens.ac.id/1324/1/Paper_TA_MBAH.pdf

Munandar, A. (2020). Programming language. THEMATICS | Technology Management and Informatics Research Journals TEMATICS | Technology Management and Informatics Research Journals, 4(2), 12.

Nasha Hikmatia AE, & Zul, MI (2021). An Android-based Indonesian Sign Language to Speech Translator Application using Tensorflow. Applied Computer Journal, 7(1), 74–83. https://doi.org/10.35143/jkt.v7i1.4629

Phiadelvira, BY (2021). Classification of cervical cancer based on colposcopy images using the Convolutional Neural Network (CNN) Alexnet Model. In JIKO (Jurnal Informatika dan Komputer) (Vol. 4, Issue 1).

Sasikala, S., Bharathi, M., & Sowmiya, B. R. (2018). Lung Cancer Detection and Classification Using Deep CNN. 2, 259–262.

Setiawan, H., Utami, E., & Al Fatta, H. (2020). Application of Arima and Artificial Neural Networks to Predict Dengue Fever Patients in Sragen Regency. Jogja Maritime Scientific Magazine, 18(2), 64–78. https://doi.org/10.33489/mibj.v18i2.220

Sung, H., Ferlay, J., Siegel, R.L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020 : GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. 71(3), 209–249. https://doi.org/10.3322/caac.21660

Wirawan, S., Wayan,) I, Wicaksana, S.,

Guritno, S., & Harjoko, A. (2005). Embedded XML Data in Images with SVG Format for Medical Image Representation. National Seminar on Information Technology. http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd
Yuliani, S., Lubis, LE, Nurlely, N., & Soejoko, DS (2018). Quantitation and analysis of computed radiography images in paranasal sinus examinations of pediatric patients using the line profile method. Journal of Medical Physics and Biophysics, 5(1), 139–154. http://jmpb.org/index.php/jmpb/article/view/316