


Diagnosis of Lung Cancer from Computed Tomography Scans with Deep Learning Methods


Furkan Berk Seyrek

(Department of Information Systems Engineering, Institute of Science, Kocaeli University,
Kocaeli, Turkey

 <https://orcid.org/0000-0002-4780-7181>, fberkseayrek41@gmail.com)

Halil Yiğit

(Department of Information Systems Engineering, Technology Faculty, Kocaeli University,
Kocaeli, Turkey

 <https://orcid.org/0000-0003-0932-6966>, halilyigit@kocaeli.edu.tr)

Abstract: In recent years, rapid advancements in technology, particularly in the realm of artificial intelligence, have significantly transformed the landscape of lung cancer diagnosis. Early detection of lung cancer is pivotal in enhancing patient outcomes; however, traditional diagnostic methods are laborious and time-consuming. Leveraging the power of deep learning techniques, specifically utilizing established neural network architectures, offers a promising solution. This study focuses on the classification of lung images from computed tomography (CT) scans into cancerous and non-cancerous categories. By employing prevalent deep learning models, transfer learning, and rigorous evaluation metrics, this study aims to assess the efficiency of these models in accurately diagnosing lung cancer. The study uses a publicly available dataset and employs preprocessing and segmentation techniques to prepare the images for analysis. The performance of the deep learning models is evaluated on the basis of parameters such as accuracy, sensitivity, specificity, and F1 score. The results demonstrate remarkable accuracy rates, with specific architectures such as ResNet-152V2 and the proposed deep convolutional neural network architecture achieving a staggering 99.1% accuracy. These findings underscore the potential of deep learning techniques in revolutionizing lung cancer diagnosis, offering valuable support to healthcare professionals, and paving the way for more efficient and accurate diagnostic practices.

Keywords: Lung Cancer, Deep Learning, Classification, Transfer Learning

Categories: I.2.1, I.4.

DOI: 10.3897/jucs.116916

1 Introduction

Cancer, which characterized by uncontrolled cell growth and division, is a disease that inflicts physical, emotional, and financial burdens on those it afflicts. A leading cause of mortality worldwide, cancer demands innovative solutions, particularly in the realm of accurate and early diagnosis. Among its numerous variants, lung cancer is the most prevalent and lethal [Siegel et al. 22], often diagnosed in advanced stages when treatment options are limited, and prognosis is bleak. Despite significant advances in cancer research and treatment, traditional diagnostic techniques have limitations that prevent timely and accurate detection of cancer. According to a study [Bogoni et al. 12], it usually takes 2–3.5 min for an experienced radiologist to examine a single slice of a CT scan. This time may vary depending on the complexity and characteristics of

the image and the experience and expertise of the radiologist. The results of another study [Al Mohammad et al. 17] showed that only 68% of lung cancer nodules were diagnosed correctly when a single radiologist reviewed the scan, whereas the accuracy rate increased to 82% when two radiologists examined the same scan. This suggests that detecting early-stage lung cancer nodules is a challenging and time-consuming task for radiologists, who must carefully review multiple scans to identify small and potentially cancerous nodules. The high workload and the need for precise, careful analysis make this a difficult and error-prone process, highlighting the need for more effective and efficient tools and techniques for the early diagnosis of lung cancer. Therefore, the imperative to revolutionize cancer diagnostics, especially lung cancer, has triggered a paradigm shift in the field and is driving the field toward the integration of advanced technologies such as artificial intelligence (AI), machine learning, and deep learning. AI offers the advantage of processing vast amounts of data quickly and accurately. This capability is crucial in cancer diagnosis, where numerous imaging and molecular data are generated for each patient. AI algorithms can automatically analyse these data, identifying complex patterns that might be challenging for humans to perceive. This increases the accuracy and timeliness of cancer diagnoses, leading to more effective treatments and outcomes and providing healthcare professionals with an unattainable level of convenience. At the heart of these technological benefits is the application of deep learning, a subset of machine learning that utilizes artificial neural networks, particularly deep neural networks with multiple hidden layers, to discern intricate patterns within data. In the context of cancer diagnosis, especially lung cancer, deep learning algorithms demonstrate unparalleled prowess. Deep learning models, trained on meticulously curated datasets comprising lung tissue images, can automatically identify and classify abnormalities, including cancerous nodules, with remarkable accuracy. Transfer learning, which is also a powerful machine learning technique, enables the adaptation of pre-trained models to tackle related problems, bridging the gap when limited data or computational resources impede conventional training. By repurposing knowledge acquired from analogous tasks, these models can swiftly and efficiently process new data, thereby enhancing diagnostic accuracy and expediting the path toward tailored treatment strategies.

This study aims to develop methods that can classify lung images in DICOM format obtained from CT scans as cancerous or non-cancerous by using deep learning and transfer learning technologies in the field of artificial intelligence, thus improving patient outcomes and reducing the burden on healthcare professionals by increasing the accuracy and speed of lung cancer diagnosis.

The paper is structured as follows: Section 2 provides a brief overview of related works in the field. Section 3 elaborates on the methodology, including dataset description, preprocessing and segmentation stage, and the deep learning models utilized in the study. Section 4 presents the results of the research, followed by the conclusion in Section 5, which discusses the implications of the findings and potential avenues for future research in the domain of AI-driven cancer diagnosis.

2 Literature Review

Pulmonary nodules, which are often indicative of lung cancer, pose a significant challenge in medical diagnosis. In recent years, the widespread adoption of deep

learning systems has revolutionized the automated classification of these nodules, owing to their efficiency and end-to-end design [Greenspan et al. 16]. Several studies have showcased the effectiveness of these systems in accurately identifying and classifying pulmonary nodules, showcasing their potential for improving patient outcomes and streamlining healthcare practices. Researchers have harnessed the power of deep convolutional neural networks (DCNNs) for pulmonary nodule classification, building upon well-established architectures such as LeNet-5 [LeCun et al. 98], AlexNet [Krizhevsky et al. 12], VGGNet [Simonyan and Zisserman 14], GoogLeNet [Szegedy et al. 15], and ResNet [He et al. 16]. These architectures have been widely adopted due to their proven efficacy and ease of use. While many studies have opted to modify existing CNNs for enhanced performance, others have ventured into designing novel CNNs from scratch, offering increased flexibility and customization, albeit at the cost of additional time and effort in development and training.

Ibrahim et al. [Ibrahim et al. 21] developed a multi-class deep learning model to diagnose COVID-19, pneumonia, and lung cancer using a combination of chest X-ray and CT images. The model was implemented in three stages: data preprocessing, deep learning models for feature extraction, and classification. They took chest images from CT and X-ray scans as input and classified the images into four categories: COVID-19, normal, pneumonia, or lung cancer. The preprocessing stage included tasks such as resizing, image augmentation, and splitting the data into training and validation sets (70% and 30% respectively). Techniques such as flipping, rotation, and tilting were used to increase the number of training images, resulting in a total of 75,000 images, augmented from the original 33,676 images. They evaluated the performance of four models: VGG-19+CNN, ResNet-152V2, ResNet-152V2+Gated Recurrent Unit, and ResNet-152V2+Bidirectional Gated Recurrent Unit. The VGG-19+CNN model outperformed the other three models, achieving an accuracy of 98.05%, sensitivity of 98.05%, precision of 98.43%, specificity of 99.5%, F1 score of 0.9824, and an area under the curve (AUC) of 0.9966.

In their study, Baranwal et al. [Baranwal et al. 21] performed a three-class classification of lung cancer images (normal, adenocarcinoma, and squamous cell carcinoma) using ResNet-50, VGG-19, InceptionResNet-V2, and DenseNet-121 for feature extraction. They applied a triplet loss to guide the CNN, increasing the inter-cluster distance and reducing the intra-cluster distance. The performance of all four CNN models, including VGG-19, ResNet-50, InceptionResNet-V2, and DenseNet-121, was improved by fine-tuning hyperparameters. InceptionResNet-V2 achieved the highest accuracy rate at 99.7%, while the accuracy rates for VGG-19, ResNet-50, and DenseNet-121 were 92.1%, 99%, and 99.4%, respectively. When the triplet neural network model was trained on these pre-trained models, DenseNet-121 obtained the highest test accuracy among these four models with a test accuracy of 99.08%. The test accuracies for the other three models, VGG-19, ResNet-50 and InceptionResNet-V2, were 97.69%, 96.2% and 97.04%, respectively.

Bahat and Görgel [Bahat and Görgel 21] introduced a lung cancer diagnosis system using deep learning techniques to classify cancerous and healthy lung images. They enhanced accuracy by preprocessing images before training them with deep learning. At this stage, the images were subjected to the Gabor filter method with different orientation values. The resulting images were normalized to a size of 32×32 and fed into the deep learning architecture. The study used a total of 25,125 lung images, including 21,600 cancerous lung images and 3,525 healthy lung images, in the

experiments. 20% of this data was set aside as test data. The Gabor filter was applied separately with orientation values of 60 and 90 for comparison. When the orientation was set to 60, the accuracy was 96.48%, sensitivity was 85.07%, specificity was 98.44%, and the F1 score was 0.8763. When the orientation was set to 90, the accuracy increased to 98.37%, sensitivity to 96.20%, specificity to 98.74%, and F1 score to 0.9453.

Mastouri et al. [Mastouri et al. 21] introduced a novel approach called the bilinear convolutional neural network (BCNN) to classify lung nodules in CT images. This method utilizes two CNNs (VGG-16 and VGG-19) for feature extraction and an SVM for false positive reduction. The researchers tested three different combinations of BCNN and found that the combination of VGG-16 and VGG-19 with SVM outperformed the other two architectures in terms of performance. When evaluated on a dataset comprising 3186 images from the publicly available Lung Nodule Analysis 2016 (LUNA16) database, the BCNN [VGG-16, VGG-19] combination achieved an accuracy rate of 91.99% and an AUC of 0.959. In comparison, the [VGG-16]² and [VGG-19]² architectures achieved accuracies of 91.84% and 90.58% with AUC values of 0.948 and 0.94, respectively.

Kawathekar and Areeckal [Kawathekar and Areeckal 22] proposed a machine learning approach for the multi-class classification of lung nodules into solid, semi-solid and Ground Glass Object tissue classes. The researchers employed feature extraction techniques such as gray-level co-occurrence matrix, Gabor filters, and local binary patterns. They validated the performance on the lung nodule database (LNDb) dataset. The best-performing classifier, K-Nearest Neighbors (KNN), achieved an accuracy of 94% and an F1 score of 0.92. Both support vector machine (SVM) and logistic regression attained an accuracy of 91% and an F1 score of 0.89.

Al-Shabi et al. [Al-Shabi et al. 22] proposed a novel Progressive Growing Channel Attentive Non-Local (ProCAN) network for lung nodule classification. This method addresses the challenge of accurate classification in three ways. First, the researchers developed a non-local network by adding channel-wise attention capability. Second, they applied the principles of curriculum learning by training the model on easy examples before moving on to more challenging ones. Third, during the curriculum learning process, the model was progressively enlarged to enhance its ability to perform the task as the classification task became more difficult. The ProCAN model was tested on two publicly available datasets and compared with state-of-the-art methods. The results showed superior performance, achieving an AUC of 0.9805 and an accuracy of 95.28% on the LIDC-IDRI dataset, outperforming these methods.

Vaiyapuri et al. [Vaiyapuri et al. 22] proposed a computer-aided diagnostic model named CSO-CADLCC that uses cat swarm optimization (CSO) for lung cancer classification. The model begins by preprocessing the data using a noise removal technique based on Gabor filtering. Feature extraction is performed using the NASNetLarge model. The extracted features are then optimized using the CSO algorithm and fed into a weighted extreme learning machine (WELM) model. The optimized WELM model was used for lung nodule classification. The performance of the CSO-CADLCC model was evaluated on a comparison dataset, achieving high accuracy, precision, sensitivity, specificity and F1 score with values of 98.89%, 98.41%, 98.25%, 99.17% and 0.9829 respectively.

Rawat et al. [Rawat et al. 22] addressed the early prediction of lung cancer by exploring four machine learning algorithms - Bayes Net, Naive Bayes, Decision Tree,

and Random Forest - as well as an artificial neural network as a deep learning algorithm. They evaluated these algorithms on a lung cancer dataset using state-of-the-art parameters, with accuracy as the primary evaluation metric. According to the experimental results, the artificial neural network achieved the best performance with an accuracy rate of 92.23%. The accuracy rate was also evaluated for artificial neural networks with one, two, and three hidden layers, and single-layer artificial neural networks obtained the highest accuracy.

Singh et al. [Singh et al. 22] proposed a convolutional neural network model based on the pre-trained ResNet-50 for classifying lung CT scans into cancerous or non-cancerous categories. They evaluated the performance of the proposed system using 2478 lung CT scan images collected from the LUNA16 dataset. Due to a significant class imbalance in the dataset, they addressed this issue by augmenting cancerous images using data augmentation techniques, including rotation, cropping, flipping, and zooming. After the augmentation process, the dataset was split into training and testing sets with a 70–30 ratio. The images were resized to 50×50 and converted to grayscale for further processing. The proposed transfer learning-based model achieved an accuracy rate of 99.1% and an F1 score of 0.9901.

3 Materials and Methods

3.1 Dataset

The Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI) dataset was utilized in this study. LIDC-IDRI is a prominent and extensively used dataset for lung disease research and is widely recognized for its vast collection of high-quality lung imaging data [Armato et al. 11]. The dataset encompasses 1010 patients and comprises a total of 244527 medical images, including computed tomography, digital radiography, and computed radiography images. Various axial slices were acquired for each patient, resulting in a diverse set of imaging data. Each individual image is standardized to 512×512 pixels and is represented in a single-channel greyscale format. The images within the LIDC-IDRI dataset were meticulously annotated by four experienced thoracic radiologists. These annotations provide valuable ground truth data, indicating regions of interest and abnormalities within lung images. In addition, the dataset includes relevant XML files containing detailed descriptions. When combined with high-resolution images, these rich annotations enable the development and evaluation of advanced machine learning algorithms. In this study, a total of 10991 images were collected from the LIDC-IDRI dataset, which included 110 patients. Of these, 5,491 images were labelled as cancerous, and 5500 images were labelled as non-cancerous (healthy).

3.2 Preparing Data

The CT scans exploited in this study underwent a series of preprocessing and segmentation steps such as Sobel filters and watershed segmentation to ensure suitability for deep learning model training and evaluation.

The images were resampled to achieve a uniform resolution, enhancing consistency across the dataset. Subsequently, resizing operations were applied to standardize the

images to specific dimensions, facilitating seamless processing during deep learning model training.

The Sobel filter is a type of edge detection algorithm widely implemented in image processing. In the context of lung cancer diagnosis and classification, Sobel filters can be employed as part of the image processing stage to enhance the quality of CT images. This is attributed to their ability to increase the contrast between different tissues and structures, facilitating their identification and segmentation into distinct regions. For instance, in a CT scan of the lungs, a Sobel filter may be applied to emphasize the edges between healthy lung tissue and air-filled spaces such as the trachea and bronchi. The Sobel operator performs a two-dimensional spatial gradient measurement on an image, highlighting regions with high spatial frequency corresponding to edges. Typically, it is used to find the approximate absolute gradient magnitude at each point in a grayscale image. The operator theoretically consists of a pair of 3×3 convolution kernels, one being the simple transpose of the other. These kernels are designed to operate vertically and horizontally with respect to the pixel grid, responding maximally to edges running in both vertical and horizontal directions. When applying the vertical gradient mask (S_x) to the image, it enhances the vertical edges, functioning similar to a first-order derivative by calculating the difference in pixel intensities around the edges. The center column, which is zero, focuses on the difference between the values of pixels to the right and left of the edge. Additionally, the center values in both the first and third columns are 2 and -2, respectively, giving more weight to the pixel values around the edge region. This intensifies the edge density, resulting in an enhanced image compared to the original. The same process applies to the horizontal gradient (S_y). These kernels can be separately applied to the input image in each orientation to produce separate measurements of the gradient components. These measurements were then combined to determine the absolute magnitude and direction of the gradient at each point in the image.

Watershed segmentation is a region-based segmentation technique employed in image processing to identify and separate different objects or regions within an image. It is based on the concept of a watershed, a land region that separates two areas draining into different rivers or water bodies [Bankman, 09]. To perform watershed segmentation, preprocessing of the image is initially required to enhance the contrast between different tissues and structures, which can be achieved through techniques such as histogram equalization or Sobel filters. After image preprocessing, a seed must be selected for each object or region of interest. In the context of lung cancer diagnosis, seeds can be placed along the edges of the lungs and within tumors. The watershed algorithm then gradually separates different objects or regions in the image by overflowing the seeds. This process continues until all seeds are entirely surrounded by their respective regions, creating clear boundaries between them.

The following preprocessing and segmentation, the images were transformed into numerical arrays, a format suitable for feeding into the deep learning model. The preprocessed images were divided into training and testing sets, with 80% of the images allocated for training and the remaining 20% for testing. The training set was used to train the deep learning model, allowing it to learn the underlying patterns, while the testing set was employed to assess the model's accuracy and generalization ability. These meticulous data preparation steps were crucial in enhancing the quality of the dataset, enabling the deep learning model to effectively diagnose lung cancer based on the distinct features extracted from the preprocessed and segmented images.

3.3 Deep Learning Models

AlexNet is a convolutional neural network designed to be deeper and wider than previous CNN models. It consists of eight layers and more than 60 million parameters, including five convolutional layers and three fully connected layers. It was specifically developed to enhance the performance of image classification tasks and won the Large-Scale Visual Recognition Challenge (ILSVRC) in 2012, establishing itself as a pioneering model in the field of deep learning. One of the key innovations of AlexNet is its use of a Rectified Linear Unit (ReLU) activation function and dropout to prevent overfitting. Moreover, AlexNet made significant strides by utilizing Graphics Processing Units (GPUs) for training, which greatly accelerated the training process compared with previous models that were trained on Central Processing Units (CPUs). This breakthrough allowed AlexNet to train deeper and wider models efficiently. In the ILSVRC competition, AlexNet achieved a top-5 error rate of 15.3%, marking a significant improvement over the previous technology. This success demonstrated the potential of deep learning models in achieving high accuracy in challenging image recognition tasks. The architecture of AlexNet includes two GPUs, each of which is responsible for specific layers. One GPU processes the upper-layer portions, while the other handles the lower-layer portions. GPUs communicate only at certain layers. The first convolutional layer filters the input image ($224 \times 224 \times 3$) with 96 kernels of size 11×11 using a stride of 4 pixels. Subsequently, the first max-pooling layer with a size of 3×3 and a stride of 2 is applied. This produces a feature map of size $27 \times 27 \times 96$. The second convolutional layer filters the output with 256 kernels of size 5×5 , resulting in a feature map of size $13 \times 13 \times 256$. Again, a max-pooling layer with a size of 3×3 and a stride of 2 is applied, producing a feature map of size $6 \times 6 \times 256$. The third, fourth, and fifth convolutional layers are connected directly without any pooling or normalization layers. The third and fourth convolutional layers have 384 kernels of size 3×3 , and the fifth convolutional layer has 256 kernels of size 3×3 . Following the last convolutional layer, a third max-pooling layer with a size of 6×6 and a stride of 1 is applied, resulting in a feature map of size $1 \times 1 \times 256$. There is a dropout layer with a dropout rate of 0.5, followed by the first fully connected layer with 4096 neurons. Another dropout layer with a dropout rate of 0.5 follows the first fully connected layer. This is succeeded by a second fully connected layer with 4096 neurons. Finally, there is an output layer with 1000 neurons corresponding to the 1000 classes in the ImageNet dataset. The SoftMax activation function is applied in this output layer for classification. All convolution and fully connected layers, except the output layer, use the Rectified Linear Unit (ReLU) activation function. In addition, the first and second convolutional layers are followed by normalization layers.

In this study's version of AlexNet, enhancements were made to optimize its performance for specific tasks. Batch normalization was added to the fifth convolutional layer, enhancing internal representations and speeding up training. The output layer used the sigmoid function, customizing the model for precise binary classification. With inputs standardized at $227 \times 227 \times 3$, the model demonstrated adaptability across diverse datasets. Impressively complex, it comprised 58,287,873 parameters, with 58,286,657 being trainable, demonstrating its capability for nuanced binary decision-making. Figure 1 summarizes the proposed AlexNet model.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 55, 55, 96)	34944
batch_normalization (Batch Normalization)	(None, 55, 55, 96)	384
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
conv2d_1 (Conv2D)	(None, 27, 27, 256)	614656
batch_normalization_1 (Batch Normalization)	(None, 27, 27, 256)	1024
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_2 (Conv2D)	(None, 13, 13, 384)	885120
conv2d_3 (Conv2D)	(None, 13, 13, 384)	1327488
conv2d_4 (Conv2D)	(None, 13, 13, 256)	884992
batch_normalization_2 (Batch Normalization)	(None, 13, 13, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 256)	0
dropout (Dropout)	(None, 6, 6, 256)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 4096)	37752832
dropout_1 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dense_2 (Dense)	(None, 1)	4097
=====		
Total params: 58,287,873		
Trainable params: 58,286,657		
Non-trainable params: 1,216		

Figure 1: Summary of the proposed AlexNet model

VGGNet is a standard deep convolutional neural network architecture with multiple layers, named after the Visual Geometry Group (VGG) at Oxford University where the research was conducted. It is known for using small 3×3 convolutional filters with a step size of 1 and has proven to be effective in extracting features from images. The use of small filters allows the network to capture fine-grained details, which are crucial for accurate object recognition in images. Additionally, it employs max-pooling layers to subsample the feature maps and reduce the computational complexity of the network. VGGNet comes in different architectures, such as VGG-16 and VGG-19, where the numbers represent the count of weight layers (convolutional layers) in the model. For example, VGG-19 has three more convolutional layers than VGG-16. VGGNet takes an input image of size 224×224 . At a single test scale, VGGNet achieved a top-5 error

rate of 8.0%. In multiple test scales, it achieved a top-5 error rate of 7.5% (Simonyan and Zisserman, 2014). The VGGNet also secured second place in the 2014 ILSVRC competition with a top-5 error rate of 7.3%. The VGG-19 consists of 19 layers, including 16 convolutional layers and 3 fully connected layers. It includes five 2×2 max-pooling layers with a step size of 2. The last three layers of VGG-19 are fully connected. The first two fully connected layers have 4096 neurons each, and the third (output) layer has 1000 channels, employing the SoftMax activation function. All hidden layers in the VGG network use the ReLU activation function. The VGG-19 architecture has approximately 144 million parameters.

In the refinement process of the VGG-19 model, a meticulous approach involving transfer learning and fine-tuning techniques was employed. Initially, the model with ImageNet weights was loaded with the "include_top" argument set to "False," ensuring that the original fully connected output layers were not loaded. This strategic move facilitated the addition and training of a new output layer tailored to the specific task. Subsequently, all other layers of the original model were frozen, preserving their learned features. To enhance the model's predictive capabilities, a single-unit dense layer was added, incorporating global average pooling and binary classification, with the sigmoid activation function utilized in the output layer. This configuration resulted in a model trained with 513 parameters, paving the way for nuanced binary classification. The pivotal phase followed, in which the entire previously frozen model was unfrozen and retrained end-to-end. This comprehensive retraining, conducted with an exceptionally low learning rate of 0.00001, encompassed 20,024,897 parameters, offering the prospect of incremental improvements in model performance. Crucially, these operations were executed after ensuring the convergence of the model with frozen layers, aligning with best practices to prevent the disruption of pre-trained features and ensuring the integrity of the refined VGG-19 model. In Figure 2, the summary of the proposed VGG-19 model is shown.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
vgg19 (Functional)	(None, 7, 7, 512)	20024384
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 1)	513
=====		
Total params: 20,024,897		
Trainable params: 513		
Non-trainable params: 20,024,384		

Figure 2: Summary of the proposed VGG-19 model

Residual Network (ResNet) is a type of convolutional neural network developed to address the problem of training very deep networks. This is achieved by using residual blocks and skip connections. A residual block is a stack of layers in which the output of one layer is taken and added to a deeper layer in the block. Non-linearity is applied after the addition of the corresponding layer's output from the main path. This bypass connection is known as a "shortcut" or "skip connection". In this architecture, the model has a direct connection that skips some layers. This skip connection forms the backbone of the residual blocks. Due to this skip connection, the output is not the same as before. The calculation of the output without a skip connection is shown in (1).

$$H(x) = f(wx + b) \text{ or } H(x) = f(x) \quad (1)$$

Here, x is multiplied by the layer's weights (w), and then a bias term (b) is added. The activation function is then applied, and the output is taken as $H(x)$. With the introduction of the skip connection technique, the output $H(x)$ has changed, as shown in (2).

$$H(x) = f(x) + x \quad (2)$$

However, the input dimension might be different from the output dimension, which can occur in convolutional or pooling layers. This problem can be tackled in two ways: zero-padding with the skip connection to increase the dimensions or adding 1×1 convolutional layers to match the dimensions. In the second approach, the output is calculated as shown in (3).

$$H(x) = f(x) + w1. x \quad (3)$$

Here, an additional parameter ($w1$) is added, whereas no extra parameters are added when using the first approach. This skip connection technique in ResNet addresses the problem of vanishing gradients in deep CNNs by allowing an alternative shortcut path for the gradient flow. In addition, skip connections assist in bypassing any architectural performance degradation caused by regularization. ResNet achieved a first-place finish in the ILSVRC 2015 classification competition with a top-5 error rate of 3.57% (He et al., 2016). There are three different architectures, each with its own variations: ResNet-50, ResNet-101, and ResNet-152. Additionally, there is a second version for each architecture. The difference between the original ResNet and ResNetV2 is that V2 uses batch normalization before each weight layer. It accepts images of size 224×224 as input and has approximately 60 million parameters. In the custom configuration of the ResNet-152V2 model, a strategic blend of transfer learning techniques and tailored adjustments was implemented. When initializing the model with ImageNet weights, the "include_top" argument was set to "False," ensuring the exclusion of the original model's fully connected layers. Unlike previous approaches, only the last 200 layers were frozen, allowing adaptability in the remaining parts of the network. New layers were seamlessly integrated, starting with a flattening layer, followed by two fully connected layers, each housing 128 neurons, with a sparsity layer (0.5) introduced in between. For the binary classification task, a single-unit dense layer exploiting a sigmoid activation function was used as the output layer. This meticulous configuration resulted in a model with 71,193,473 parameters, notably, 44,603,777 of which were trainable, emphasizing the model's ability to finely tune its internal representations for

nuanced binary decision-making. Figure 3 summarizes the proposed ResNet-152V2 model.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
resnet152v2 (Functional)	(None, 7, 7, 2048)	58331648
flatten (Flatten)	(None, 100352)	0
fc1 (Dense)	(None, 128)	12845184
dropout (Dropout)	(None, 128)	0
fc2 (Dense)	(None, 128)	16512
dense (Dense)	(None, 1)	129
=====		
Total params: 71,193,473		
Trainable params: 44,603,777		
Non-trainable params: 26,589,696		

Figure 3: Summary of the proposed ResNet-152V2 model

The developed deep convolutional neural network represents a sophisticated tool for classifying lung images, each measuring $512 \times 512 \times 1$. With meticulous design, the model comprises six convolutional layers, featuring a gradual increase in the number of kernels from 32 to 64, each with a size of 3×3 . These layers play a pivotal role in extracting intricate features from the input images and creating a hierarchical representation of the data. Following each convolutional layer, a 2×2 max-pooling layer is applied, effectively reducing the spatial dimensions of feature maps while retaining crucial information. This process not only aids in feature extraction but also mitigates overfitting by minimizing the model's parameters, enhancing its generalization ability. Post-convolution and pooling layers, the model incorporates a smoothing layer that transforms the output into a one-dimensional vector. This vector undergoes further processing through three fully connected layers, each equipped with 128 neurons and employing the ReLU activation function. The integration of ReLU introduces the necessary nonlinearity, enabling the model to discern intricate patterns within the data. The final layer, which features a single neuron and a sigmoid activation function, culminates the network, providing the ultimate binary classification decision. To fortify the model against overfitting, dropout layers with a dropout ratio of 0.5 are strategically inserted after the third convolutional layer and the first fully connected layer. This deliberate dropout helps maintain model integrity by preventing overreliance on specific features and fostering a well-rounded understanding of the data. In total, the

model encompasses 439,361 parameters, all of which are meticulously trainable, representing the intricate neural connections fundamental to its exceptional classification performance. In Figure 4, the summary of the proposed deep convolutional neural network model is depicted.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 510, 510, 32)	320
max_pooling2d (MaxPooling2D)	(None, 255, 255, 32)	0
conv2d_1 (Conv2D)	(None, 253, 253, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 126, 126, 32)	0
conv2d_2 (Conv2D)	(None, 124, 124, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 62, 62, 32)	0
dropout (Dropout)	(None, 62, 62, 32)	0
conv2d_3 (Conv2D)	(None, 60, 60, 64)	18496
max_pooling2d_3 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_4 (Conv2D)	(None, 28, 28, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_5 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 128)	295040
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 1)	129
=====		
Total params: 439,361		
Trainable params: 439,361		
Non-trainable params: 0		

Figure 4: Summary of the proposed deep convolutional neural network model

3.4 Performance Metrics

Performance metrics serve as measurements used to assess the effectiveness and functionality of a model. They offer valuable insights into the model's accuracy, reliability, and generalizability. Commonly used metrics such as accuracy, precision, and F1 score enable researchers and practitioners to compare diverse models while ensuring their work's authenticity and originality. Accuracy (4) is a measure of how well a model or system can produce the correct or expected outcomes. It is typically calculated by comparing the model's predicted output with the actual output. For example, in a classification problem, the accuracy of a model is the percentage of correct predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Here, TP (true positive) represents cases where the person has cancer (lung nodule is present) and TN (true negative) represents cases where the person is healthy (no lung nodule is present). FP (false positive) indicates cases in which a healthy person is incorrectly identified as having cancer and FN (false negative) represents cases in which a person with cancer is incorrectly identified as healthy.

In the context of machine learning, loss (5) is a measure of how well a model fits the training data. This is a numerical value representing the error or difference between the predicted output of a model and the actual output. In many machine learning algorithms, the goal is to minimize loss, which typically involves finding the set of model parameters that results in the smallest possible loss. The loss function used in a machine learning model depends on the type of problem being solved. For example, in a classification problem, the loss function might be cross-entropy, which measures the difference between the predicted probabilities and the actual labels. Binary cross-entropy is a loss function used in machine learning for binary classification tasks where the goal is to predict one of two classes (e.g., positive or negative). It measures the difference between the predicted probability and the true probability of the class.

$$Loss = \frac{1}{N} \sum_{i=1}^N -(y_i \times \log(p_i) + (1 - y_i) \times \log(1 - p_i)) \quad (5)$$

where p_i is the probability of class 1 and $(1-p_i)$ is the probability of class 0. When the observation belongs to class 1, the first part of the formula becomes active and the second part disappears, and vice versa if the actual class of the observation is 0.

Precision (6) is a measure of how consistent and accurate a model can produce results. Sensitivity (7), also known as true positive rate or recall, is a measure of how well a model can identify all positive examples in a dataset. Specificity (8), also known as the true negative rate, is a measure of how well a model can identify all negative examples in a dataset. The F1 score (9), also known as the F-measure, is a measure of the accuracy and precision of a model. The F1 score is calculated as the harmonic mean of the model's precision and sensitivity. This means that the F1 score takes into account both the percentage of correct predictions made by the model (precision) and the percentage of positive examples correctly predicted (sensitivity). The F1 score is often used as a single, composite metric to evaluate a model's performance because it provides a balanced measure of accuracy and precision.

A receiver operating characteristic (ROC) curve is a graphical representation of the performance of a binary classification model. It is a useful tool for evaluating the

accuracy and precision of a model and for comparing the relative performance of different models. The ROC curve for the binary classification model is constructed by plotting the true positive rate against the false positive rate at different classification thresholds. The resulting curve provides a visual representation of the model's performance, with a higher curve indicating a higher level of accuracy and precision.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$Specificity = \frac{TN}{TN+FP} \quad (8)$$

$$F1\ Score = \frac{2(Precision \times Sensitivity)}{(Precision + Sensitivity)} \quad (9)$$

The area under the receiver operating characteristic curve (AUC) is used to evaluate the performance of a binary classification model. In a binary classification model, the true positive rate is plotted against the false positive rate (ROC curve) at different classification thresholds. The area under the curve (AUC) is then calculated. A model with a higher AUC is generally considered to be more accurate and reliable than a model with a lower AUC.

A confusion matrix is a table used to evaluate the performance of a machine learning model. It is a useful tool for understanding the accuracy and precision of a model and identifying the specific types of errors that the model makes. For a binary classification model, the confusion matrix is constructed by counting the number of true positives, true negatives, false positives, and false negatives. These counts are then used to calculate various metrics such as precision, sensitivity, and accuracy.

Cross-validation is a useful tool for evaluating the accuracy and precision of the model and identifying potential problems with the model, such as overfitting. Cross-validation involves splitting existing data into multiple sets, such as training, validation, and testing sets. The model is then trained on the training set and evaluated on the validation and test sets. This process is repeated many times using different parts of the data to obtain a more accurate and reliable estimate of the model's performance. Cross-validation is a useful technique for evaluating the performance of a machine learning model.

K-fold cross-validation is particularly useful when the size of available data is limited. In the context of machine learning, K-fold cross-validation involves splitting existing data into k different clusters. The model is then trained and evaluated k times, each time using a different fold as the validation set and the remaining folds as the training set. Because this process uses all available data for training and evaluation, it provides a more accurate and reliable prediction of the model's performance. By using all available data for training and evaluation, k-fold cross-validation can provide a more accurate and reliable estimate of the model's performance than other techniques.

4 Results

In this study, deep learning architectures, transfer learning, and fine-tuning techniques were used to classify lung cancer from CT images. The dataset consists of 10991 images, with 5491 images labeled as cancerous and 5500 images labelled as non-cancerous. The dataset is divided into the training set (80%) and testing set (20%). Images were randomly assigned to the training and test sets.

The proposed model was implemented on the Google Colab platform using Pydicom, SciPy, Scikit-learn, NumPy, Pandas, Matplotlib, TensorFlow, and Keras libraries and the Python programming language. For each model, a sigmoid activation function was used in the output layer, binary cross-entropy was used as the loss function in the compilation stage, and ImageNet weights were used for ResNet-152V2 and VGG-19 models in the transfer learning stage. The input size is set to $227 \times 227 \times 3$ for the AlexNet model, $512 \times 512 \times 1$ for the proposed DCNN model and $224 \times 224 \times 3$ for the other models. All training phases were carried out with 250 epochs and 64 batch sizes. Only in the fine-tuning phase (second training round) was the VGG-19 model implemented with 50 epochs. All models include the Adam optimization algorithm. The VGG-19 model was trained at a learning rate of 0.00001 in the fine-tuning phase (second training), and all other training was conducted at a learning rate of 0.001. Table 1 shows the hyperparameters used in this study.

Activation Function	Sigmoid
Loss Function	Binary Cross-Entropy
Weights	ImageNet
Input Size	AlexNet - $227 \times 227 \times 3$ DCNN - $512 \times 512 \times 1$ ResNet-152V2 & VGG-192 - $24 \times 224 \times 3$
Epoch	250 (VGG-19 w/ Fine-Tuning 50)
Batch Size	64
Learning Rate	VGG-19 w/ Fine-Tuning (0.00001) AlexNet, DCNN, ResNet-152V2 (0.001)
Optimization Algorithm	Adam

Table 1: Hyperparameters used in the study

Model	Accuracy	Sensitivity	Specificity	F1 Score
AlexNet	98,05%	98,55%	96,64%	0,976
DCNN	99,1%	98%	98,82%	0,984
ResNet-152V2	99,1%	98,64%	98,55%	0,986
VGG-19	98,1%	98,37%	97,82%	0,981

Table 2: Comparison of the proposed models with accuracy, sensitivity, specificity, and F1 score metrics

Model	Loss	Precision	AUC
AlexNet	0,1107	98,67%	0,996
DCNN	0,0368	99,17%	0,998
ResNet-152V2	0,0401	99,18%	0,997
VGG-19	0,0691	98,69%	0,998

Table 3: Comparison of the proposed models with loss, precision, and AUC metrics

Model	Accuracy	10-Fold Accuracy
AlexNet	98,05%	95,69%
DCNN	99,1%	98,32%
ResNet-152V2	99,1%	96,74%
VGG-19	98,1%	92,29%

Table 4: Comparison of the average accuracy values of the proposed models obtained after applying 10-fold cross-validation with the original accuracy values

Table 2 shows the performance results of the models proposed in this study under the parameters of accuracy, sensitivity, specificity, and F1 score. When the results were examined, it was observed that the ResNet-152V2 model and the proposed DCNN model reached the highest overall accuracy of 99.1%. In addition, the ResNet-152V2 model reaches the highest F1 score and sensitivity, and the proposed DCNN model reaches the highest specificity value. Table 3 shows a comparison of the proposed models with loss, precision, and AUC metrics. The lowest loss value was obtained in the proposed DCNN model, the highest precision was obtained in the ResNet-152V2 model, and the highest AUC values were precured in the proposed DCNN and VGG-19 models. While AlexNet and VGG-19 models performed well, ResNet-152V2 and the proposed DCNN model showed the best performance in all metrics. In addition, Table 4 shows a comparison of the average accuracy values obtained after applying 10-fold cross-validation of the proposed models with the original accuracy values. The proposed DCNN model also achieves the highest accuracy value.

Figure 5 shows the complexity matrices of the proposed models. While the highest true negative and lowest false positive values were obtained in the DCNN model, the highest true positive and lowest false negative values were obtained in the ResNet-152V2 model.

The accuracy chart and ROC curve of the AlexNet model are shown in Figure 6, and the accuracy chart and ROC curve of the proposed DCNN model are presented in Figure 7. Similarly, the accuracy chart and ROC curve of the ResNet-152V2 model are

illustrated in Figure 8, and the accuracy chart and ROC curve of the VGG-19 model are displayed in Figure 9.

When looking at the accuracy graph of the AlexNet model, deviations are seen in the "validation" rates, whereas the validation and training accuracies of other models show a stable and regular increase. When the ROC curves are examined, classifiers that give curves closer to the upper left corner show better performance.



Figure 5: Confusion matrices of the proposed models

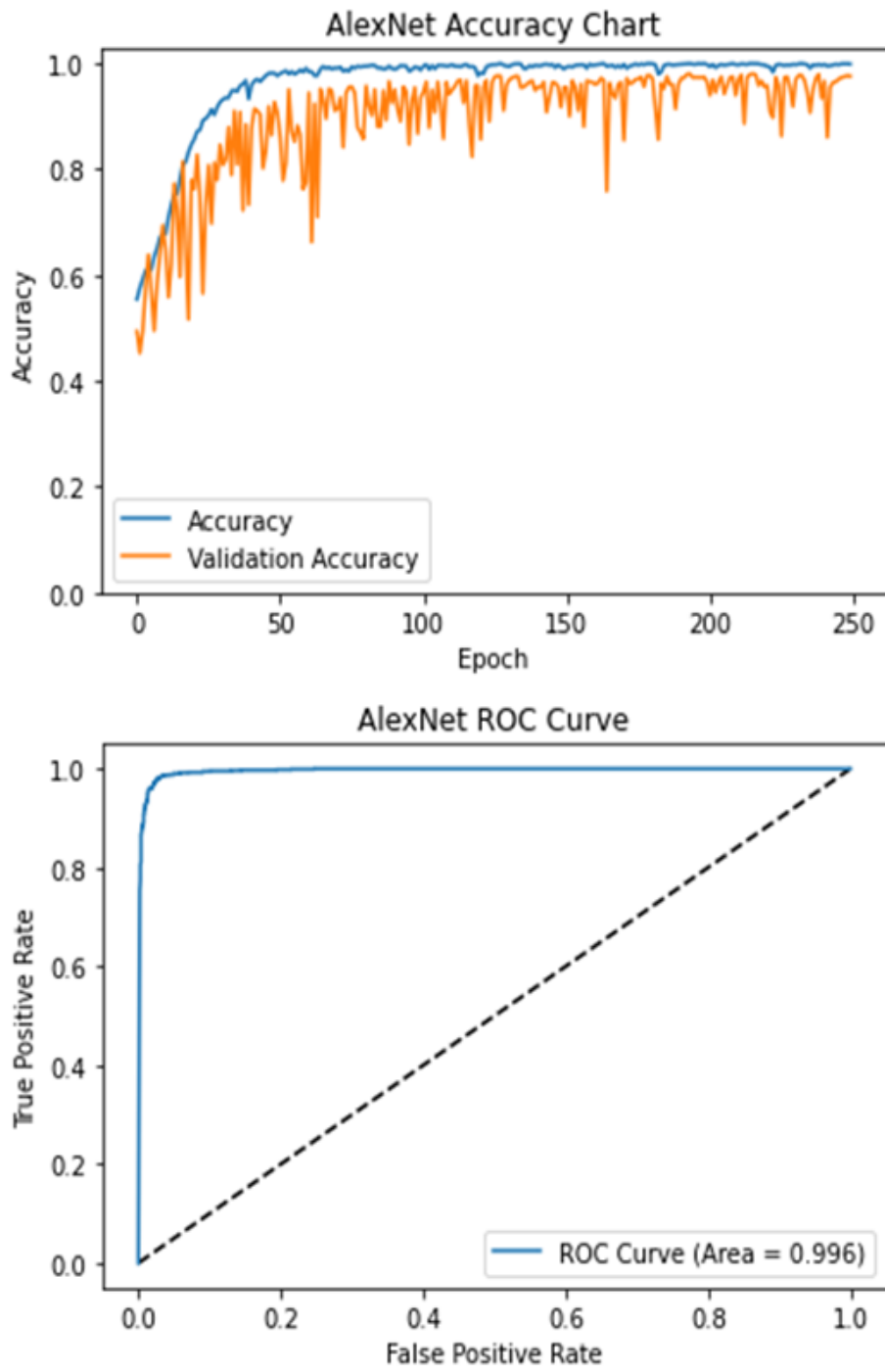


Figure 6: Accuracy chart and ROC curve of the proposed AlexNet.

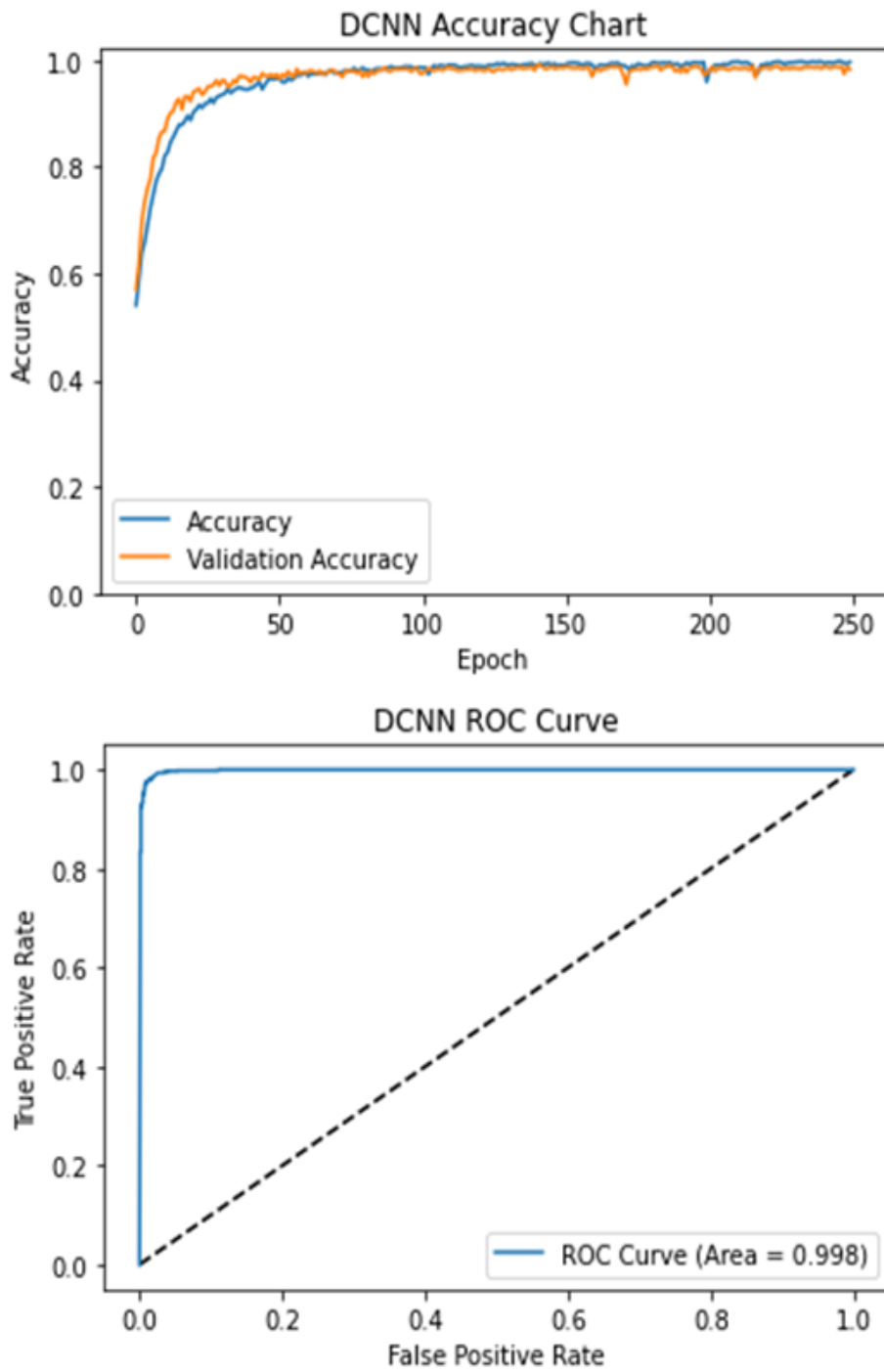


Figure 7: Accuracy chart and ROC curve of the proposed DCNN

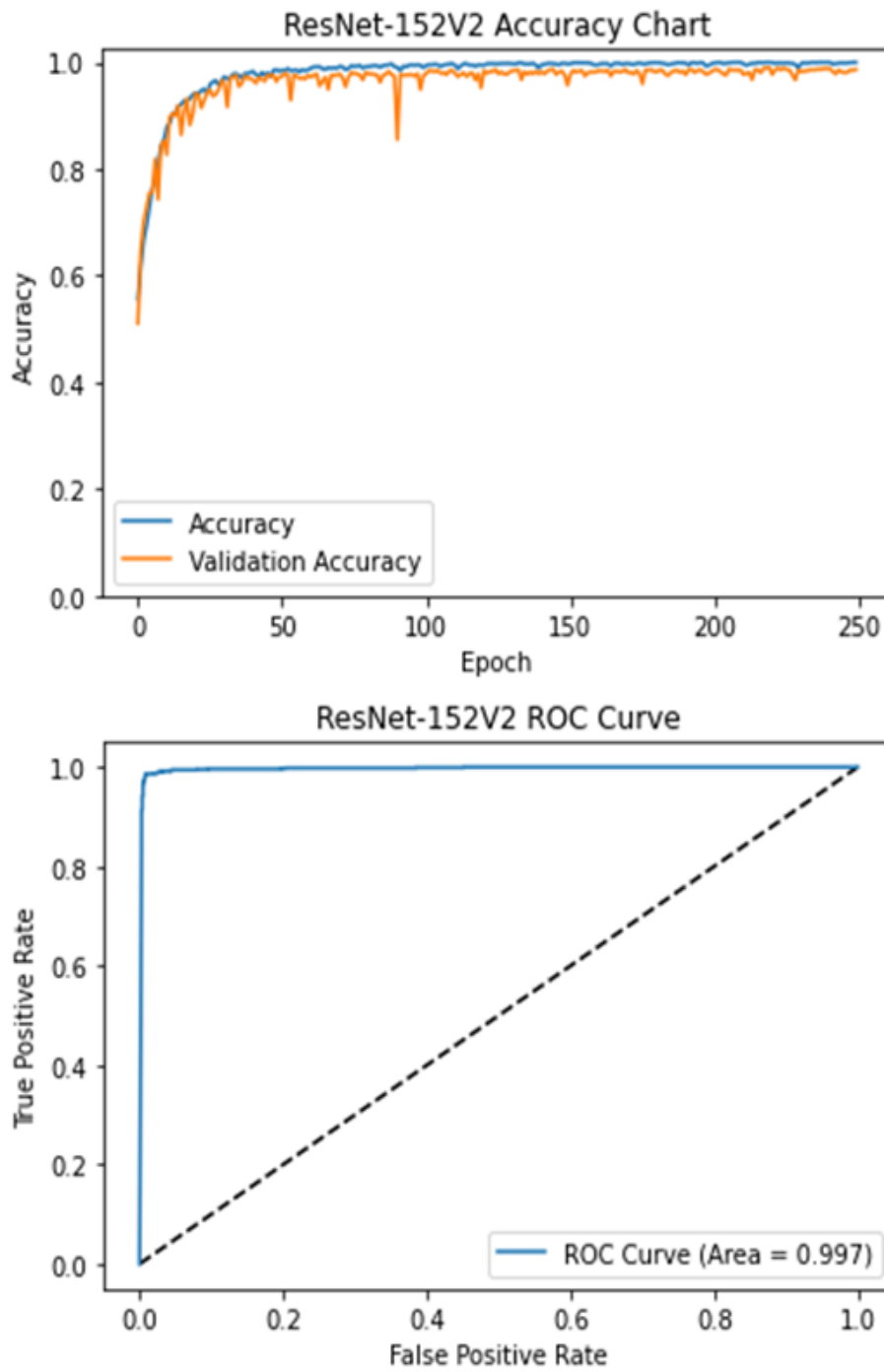


Figure 8: Accuracy chart and ROC curve of the proposed ResNet-152V2.

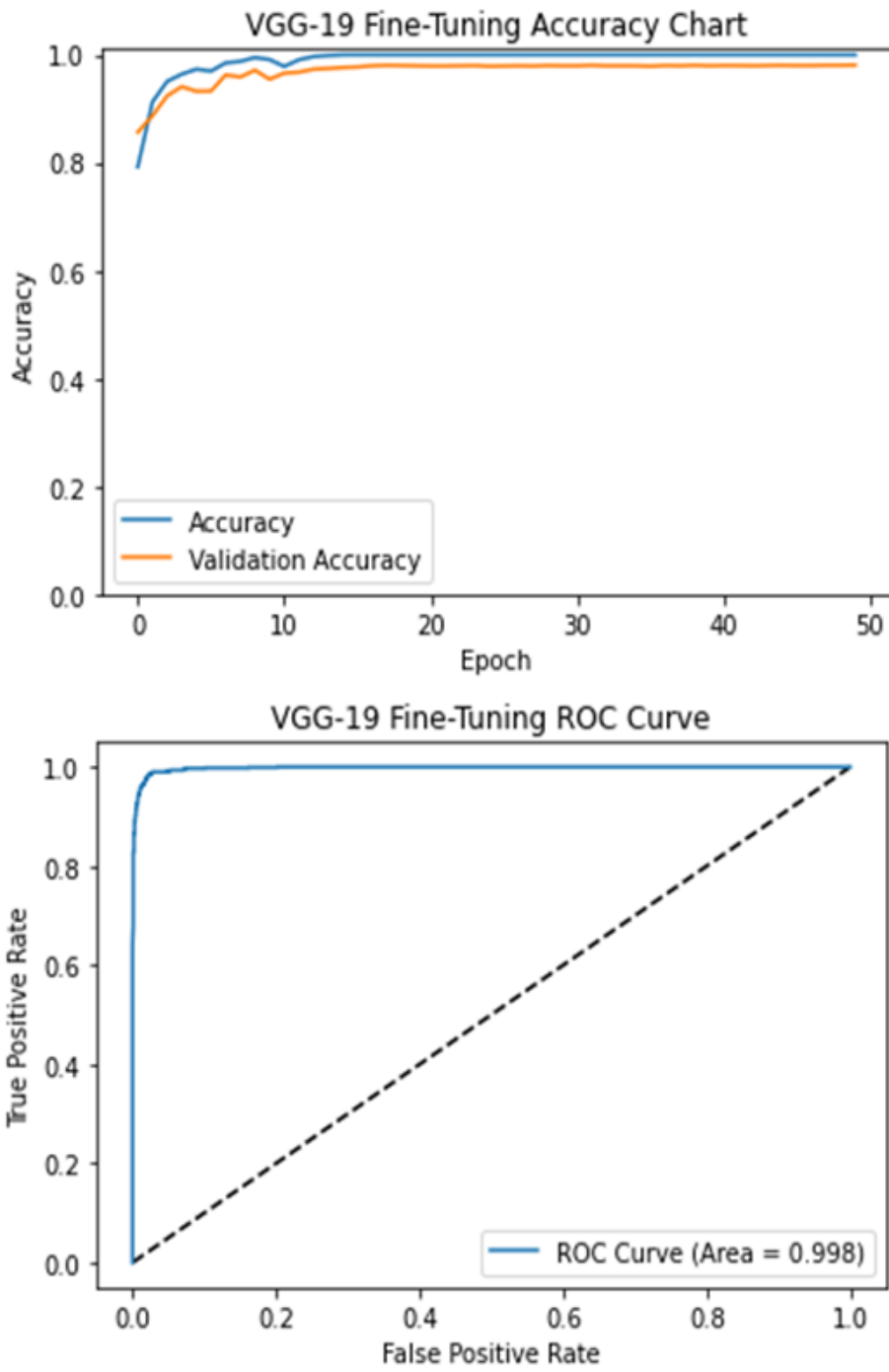


Figure 9: Accuracy chart and ROC curve of the proposed VGG-19 (Fine-Tuning)

5 Conclusion

In the realm of diagnosing lung cancer, the traditional methods employed by doctors involve a painstaking examination of multiple tests, leading to a time-consuming process. This study addresses this challenge by leveraging advanced deep learning techniques to classify lung images obtained from CT scans as cancerous or non-cancerous. Our comparative analysis delved into the performance of various architectures, including the proposed deep convolutional neural network, AlexNet, ResNet-152V2, and VGG-19. The outcomes of our evaluation, encompassing accuracy, loss, sensitivity, specificity, precision, AUC, and F1 score, highlighted the superior performance of ResNet-152V2 and the proposed deep convolutional neural network. These architectures exhibited the highest accuracy, sensitivity, and F1 score, indicating their prowess in accurately identifying lung cancer cases. In contrast, the older model (AlexNet) displayed comparatively lower performance metrics than the other models.

It is important to note that the limitations of this study, including constraints on resources and time, resulted in the exploration of deeper architectures with a limited dataset. Our future endeavours will focus on augmenting the dataset, enabling the exploration of even deeper networks, and utilizing more advanced computational resources. These steps are important for enhancing the accuracy and robustness of our models. Looking ahead, our research indicates promising prospects. Not only can we determine the presence of cancer in patients, but we also anticipate advancing our methodologies to pinpoint the precise location of cancerous nodules. By continually refining and expanding our techniques, we strive to contribute significantly to the field of medical diagnostics, facilitating quicker, more accurate, and efficient identification of lung cancer for the benefit of patients and healthcare providers alike.

References

- [Al Mohammad et al., 2017] Al Mohammad, B., Brennan, P. C., Mello-Thoms, C. (2017). A Review of Lung Cancer Screening and the Role of Computer-Aided Detection. *Clinical Radiology*, 72(6): 433–442.
- [Al-Shabi et al., 2022] Al-Shabi, M., Shak, K., Tan, M. (2022). ProCAN: Progressive Growing Channel Attentive Non-Local Network for Lung Nodule Classification. *Pattern Recognition*, 122, 108309.
- [Armato et al., 2011] Armato, S. G., 3rd, McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, et al. (2011). The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT scans. *Medical Physics*, 38(2): 915–931.
- [Bahat et al., 2021] Bahat, B., Görgel, P. (2021). Lung Cancer Diagnosis via Gabor Filters and Convolutional Neural Networks. In *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1-6, IEEE.
- [Bankman 2009] Bankman, I. N. (2009). Overview and Fundamentals of Medical Image Segmentation. Bankman, I. N. (Ed.), *Handbook of Medical Image Processing and Analysis (2nd ed.)* (73-90). Academic Press.

- [Baranwal et al. 2021] Baranwal, N., Doravari, P., Kachhoria, R. (2021). Classification of Histopathology Images of Lung Cancer Using Convolutional Neural Network (CNN). *ArXiv*, abs/2112.13553.
- [Bogoni et al., 2012] Bogoni, L., Ko, J. P., Alpert, J., Anand, V., Fantauzzi, J., Florin, C. H., Koo, C. W., Mason, D., Rom, W., Shiau, M., Salganicoff, M., Naidich, D. P. (2012). Impact of a Computer-Aided Detection (CAD) System Integrated into a Picture Archiving and Communication System (PACS) on Reader Sensitivity and Efficiency for the Detection of Lung Nodules in Thoracic CT Exams. *Journal of Digital Imaging*, 25(6): 771–781.
- [Greenspan et al., 2016] Greenspan, H., Van Ginneken, B., Summers, R. M. (2016). Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. *IEEE Transactions on Medical Imaging*, 35(5): 1153-1159.
- [He et al., 2016] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778, IEEE.
- [Ibrahim et al., 2021] Ibrahim, D. M., Elshennawy, N. M., Sarhan, A. M. (2021). Deep-Chest: Multi-Classification Deep Learning Model for Diagnosing COVID-19, Pneumonia, and Lung Cancer Chest Diseases. *Computers in Biology and Medicine*, 132: 104348.
- [Kawathekar et al., 2021] Kawathekar, I. D., Areeckal, A. S. (2022). Performance Analysis of Texture Characterization Techniques for Lung Nodule Classification. *Journal of Physics: Conference Series*, 2161.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25: 1097-1105.
- [LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11): 2278-2324.
- [Mastouri et al., 2021] Mastouri, R., Khelifa, N., Neji, H., Hantous-Zannad, S. (2021). A Bilinear Convolutional Neural Network for Lung Nodules Classification on CT Images. *International Journal of Computer Assisted Radiology and Surgery*, 16(1): 91–101.
- [Rawat et al., 2022] Rawat, D., Meenakshi, Pawar, L., Bathla, G., Kant, R. (2022). Optimized Deep Learning Model for Lung Cancer Prediction Using ANN Algorithm. In *3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 889-894, IEEE.
- [Siegel et al., 2022] Siegel, R. L., Miller, K. D., Fuchs, H. E., Jemal, A. (2022). Cancer Statistics, 2022. *CA: A Cancer Journal for Clinicians*, 72(1): 7–33.
- [Simonyan et al., 2014] Simonyan, K., Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.
- [Singh et al., 2022] Singh, L., Choudhary, H. K., Singh, S., Bisht, A. K., Jain, P., Shukla, G. (2022). Automated Detection of Lung Cancer using Transfer Learning based Deep Learning. In *2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, 500-504, IEEE.
- [Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going Deeper with Convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9, IEEE.
- [Vaiyapuri et al., 2022] Vaiyapuri, T., Liyakathunisa, Alaskar, H., Parvathi, R., Pattabiraman, V., Hussain, A. (2022). Cat Swarm Optimization-Based Computer-Aided Diagnosis Model for Lung Cancer Classification in Computed Tomography Images. *Applied Sciences*, 12(11): 5491.