International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-19

https://doi.org/10.70917/ijcisim-2026-1254
Article

An Approach for Lung Cancer Detection using SMOTE with Convolutional Network

Sri Rupin Potula [1] and Ramani Selvanambi [1,*]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, India
* Correspondence author: ramani.s@vit.ac.in

Abstract: With a significant rise in lung cancer cases globally, especially among both men and women, effective lung cancer detection techniques are critically important. This paper addresses the urgency by employing deep learning techniques for lung cancer detection. Our study is based on categorizing the images into three distinct classes: benign, malignant, and normal cases. To ensure uniformity, images of varying sizes are standardized. Addressing the challenge of data imbalance, this paper employs the "Synthetic Minority Over-sampling Technique" (SMOTE), and further enhance image quality through Gaussian Blur in the preprocessing phase. Subsequently, a "Convolutional Neural Network" (CNN) model named "ImageTriNet", compare its performance with transfer learning models. The ImageTriNet model exhibits commendable results, after 13 training epochs, attaining an accuracy of 0.98, precision of 0.99, recall of 0.96, and an F1-score of 0.97. This research contributes to the ongoing efforts in leveraging deep learning techniques for accurate and timely detection of lung cancer, showcasing efficacy of our ImageTriNet model in this critical domain.

Keywords: Lung Cancer, Convolutional Neural Network, Transfer Learning, Denses121, InceptionV3, VGG16, ResNet50, Accuracy, Precision, Recall, Kernel Filters, CT Scan Images, precision, recall, f1-score.

1. Introduction

Lung cancer stands as a formidable global health threat, ranking among the most perilous diseases worldwide [1, 2]. According to recent "World Health Organization" (WHO) estimates, the annual toll of lung cancer-related deaths reaches a staggering 7.6 million across the globe [3, 4]. Alarmingly, this grim scenario is anticipated to worsen, with projections indicating a potential surge to 17 million deaths annually by 2030 [5]. The key to combatting lung cancer lies in early detection, presenting a crucial avenue for effective intervention [6].

A variety of diagnostic techniques, including MRIs, CT scans, X-rays, and isotope imaging, are used to diagnose lung cancer. Of them, computer tomography and X-ray chest radiography stand out as the two most often utilized anatomical imaging modalities for identifying a variety of lung disorders. [7, 8]. CT scans are used by doctors and radiologists to diagnose conditions, see morphological features, explain the course and severity of diseases, and monitor the patient's reaction to therapy. Thoracic imaging has undergone a revolution with the introduction of volumetric CT techniques, such as spiral scans, which minimize artifacts caused by partial volume effects, heart motion, and irregular breathing cycles while also cutting down on scan times.

In the domain of lung cancer detection, the integration of deep learning methodologies, specifically Convolutional Neural Networks (CNNs), has surfaced as a potent technique for enhancing image pattern identification [9]. CNNs, drawing inspiration from the human visual system, exhibit a remarkable ability to autonomously learn intricate features from raw pixel data, especially in the context of medical imaging like X-ray and CT scans. In the realm of lung cancer detection, the incorporation of deep learning models, particularly Convolutional Neural Networks (CNNs), has

emerged as a powerful tool for improving image pattern recognition [9]. Their ability to discern intricate patterns and spatial relationships within these images greatly aids in early detection endeavors. Additionally, the strategic application of transfer learning amplifies their effectiveness. [10], where pre-trained models leverage insights from extensive datasets, enhances performance, proving particularly valuable when dealing with limited labelled medical data. This streamlined fusion of CNNs and transfer learning represents an innovative approach to achieving more precise and timely lung cancer diagnosis.

In this study, CT scan images utilized for analysis were sourced from the open-access Kaggle platform. Initially, a standardization process was implemented to ensure uniformity in image size, facilitating consistent and reliable results. Subsequently, a careful balancing of the image dataset was undertaken, optimizing the representation of different classes within the dataset. To enhance the visual features of the images, a Gaussian blur [11], recognized for its efficacy as a low-pass filter, was applied. This preprocessing step aimed to accentuate relevant patterns and improve the overall quality of the images. The refined images were then fed into the ImageTriNet and transfer learning models, harnessing the power of these advanced architectures for comprehensive lung cancer classification and severity assessment. This systematic approach in image preprocessing lays the foundation for robust and accurate outcomes in the subsequent stages of the research.

1.1. Motivation

Detecting lung cancer at an early stage holds immense potential to positively impact the lives of millions; nevertheless, existing standalone models frequently fail to meet expectations. This research proposes ImageTriNet, a novel convolutional neural network (CNN) designed to outperform both standalone models and traditional transfer learning approaches like VGG16, InceptionV3, ResNet, and DenseNet. ImageTriNet leverages its unique architecture, aptly named for its focus on images and its three-layer convolutional and max pooling structure. Its key features include:

- Preprocessing: Image resizing, Gaussian blur application, and normalization for optimal input data. Efficient architecture: Three convolutional and max pooling layers for a balance between accuracy and computational cost.
- Targeted design: Tailored for lung cancer detection, potentially leading to improved performance compared to generic models.

This research aims to compare the performance of ImageTriNet against established models, focusing on metrics like accuracy, sensitivity, specificity, and computational efficiency. By demonstrating ImageTriNet's effectiveness in early lung cancer detection, the capability to substantially enhance patient outcomes and potentially save lives is evident in this context.

In Section 2, a comprehensive examination of existing literature is presented, encompassing diverse approaches and metrics selected by different authors. Section 3 delves extensively into the methodology employed, providing a detailed exploration of the chosen research methods. Moving on to Section 4, a succinct overview of the experimental analysis conducted on a range of models is outlined. Section 5 is dedicated to presenting and discussing the obtained results. Concluding the paper, Section 6 offers a summary and outlines the future avenues of exploration.

2. Literature Review

In [12], the authors gathered the dataset that consists of Chest X-ray and LIDCDR. The authors have used two datasets for better analysis. In their approach, they made modifications to the AlexNet's model to detect lung cancer anomalies. The developed MAN-SVM technique achieved 96.80% accuracy in comparison to all other methods. Also, achieved an accuracy of less than 87% after few changes were made in the Deep Learning structure. This indeed, gave an accuracy of 97%.

Authors of [13], utilized CNN, three sets of ReLU layers for their neural network. A fully connected layer comes after the ReLU layers. 64 filters are used to get the representative features of each convolutional layer. The "Japanese Society of Radiological Technology" (JSRT) dataset is used for results discussion. The findings showed an average of 98.05% for accuracy, 96.25% for overlap, 98.80% for specificity, and 96.25% for sensitivity.

The authors of [14] employed CT scan pictures to diagnose lung cancer in this investigation. To improve picture segmentation, they have employed three datasets: The study utilized several datasets, including the Lung Nodule Analysis 2016 dataset, the Kaggle Data Science Bowl 2017 dataset, and the Lung Picture Database Consortium and Image Data Resource Initiative dataset. For image segmentation, the researchers employed the U-Net architecture and integrated a VGG-style 3D network. By combining validation from the U-Net architecture with the 3D multiple VGG network, they

achieved an impressive accuracy rate of 95.60%. Additionally, the investigation enabled the classification of lung nodules' malignancy levels.

In their publication [15], the authors present a novel radiomics technique that uses non-invasive CT data to predict the histology of NSCLC, or non-small cell lung cancer, tumors. The study involves the training and validation of convolutional neural networks (CNNs) on a dataset comprising 311 early-stage NSCLC patients who received surgical treatment at Massachusetts General Hospital, with a focus on the predominant histological types—adenocarcinoma (ADC) and Squamous Cell Carcinoma (SCC). The CNNs exhibit a noteworthy ability to forecast tumor histology, achieving an area under the curve (AUC) of 0.71 ($p = 0.018$). Additionally, the research investigates the application of various methods such as k-nearest neighbors and support vector machine on CNN-derived radiomics features, demonstrating comparable discriminative performance with AUCs reaching up to 0.71 ($p = 0.017$).

In a recent study [16], the multi-view knowledge-based cooperative (MV-KBC) deep neural network model was utilized by researchers to differentiate between benign and malignant lung nodules in chest CT scans. Comparing their methodology against five cutting-edge classification techniques and using the LIDC-IDRI dataset as a benchmark was part of the evaluation process. The results of their analysis showed that the MV-KBC model had an AUC of 95.70% and an accuracy of 91.60%. The researchers suggested that their approach holds promise for integration into routine clinical workflows based on these favorable findings.

This paper [17] presents an innovative methodology for the accurate identification NSCLC by employing a unique multilevel brightness-preserving approach on CT scan images. Through a meticulous examination of individual pixels, noise elimination, and overall enhancement of image quality, the method aims to elevate the precision of subsequent analytical steps. Post image enhancement, a sophisticated deep neural network is deployed to segment affected lung regions, utilizing multiple layers and extracting diverse features. The suggested method displays promising results across a range of evaluation criteria, including logarithmic loss, mean absolute error, precision, recall, and F-score. These findings highlight its capacity to potentially enhance NSCLC prediction rates significantly. The amalgamation of image enhancement and advanced segmentation techniques emphasizes the importance of a comprehensive strategy in augmenting the diagnostic accuracy of lung cancer without resorting to plagiarism.

In [18], addressing the global challenge of lung cancer, which stands as the leading cause of cancer-related deaths, involves the introduction of an innovative deep neural network. The proposed model, built upon the GoogleNet architecture, aims to classify CT scan images and distinguish between malignant and benign tissues. To optimize computational efficiency and mitigate overfitting risks, the paper introduces a novel strategy by sparsifying the densely connected architecture, assigning 60% of neurons to dropout layers. The model's performance is rigorously evaluated on the Lung Image Database Consortium dataset and compared with other CNNs, such as AlexNet, GoogleNet, and ResNet50. The obtained output showcases the superiority of the proposed network in achieving enhanced classification accuracy compared to its counterparts, underscoring the potential impact of this approach on improving lung cancer diagnostic capabilities.

In [19], this work addresses the crucial problem of monitoring lung cancer and provides a strong mechanism for automatically identifying and classifying CT scan images as benign or malignant. By integrating image processing and machine learning, the research employs seven classifiers, including k-nearest neighbors, support vector machine, decision tree, multinomial naive Bayes, stochastic gradient descent, random forest, and multi-layer perceptron. Utilizing a dataset containing 15,750 clinical images, with 6,910 representing benign cases and 8,840 malignant lung cancer-related images, the study highlights that the MLP classifier attains the highest accuracy rate of 88.55%. This outcome underscores the potential of combining image processing and machine learning techniques to refine the precision of lung cancer classification. Such integration presents valuable advancements in medical imaging, offering improved diagnostic capabilities for more effective disease identification and treatment strategies. A comparative summary of the reviewed lung cancer detection approaches, along with their datasets and performance metrics, is presented in Table 1.

Table 1. Lung Cancer Model Performances

Source	Methodology	Main Findings	Dataset	Metrics
Abhir et al [12]	Modified AlexNet	Modified AlexNet to reach accuracy	Chest X-ray	Accuracy:96.80% Specificity: 96.63% Sensitivity: 96.97% F1 score: 96.78%
			LIDC-IDR	Accuracy:97.27% Specificity:95.63% Sensitivity: 98.09% F1 score: 97.95%
Kaur et al [13]	CNN	Used deep CNN for lung field segmentation.	JSRT	Accuracy:98.05% Sensitivity:96.25% Specificity:98.80% Overlap: 93.4%
Tekade et al [14]	U-Net, VGG like network	Final results are made from predictions from U-Net and 3D multipath	LIDC-IDRI, LUNA16, Kaggle Bowl Datasets	Accuracy: 95.60% Logloss: 0.387732
Chaunzwa et al [15]	CNN	Machine Learning Algorithms on CNN for feature extraction	NSCLC	AUC: 0.71 (p=0.018)
Xie et al [16]	MV-KBC	The model developed verified against cutting-edge classification model	LIDC-IDRI	Accuracy:91.6% Sensitivity:86.5% Specificity:94.0% AUC: 95.73%
Shakeel et al [17]	Deep Neural Network and Ensemble Classifier	MATLAB based feature extraction techniques.	CT Scan Dataset	Method Accuracy RG 0.72 GT 0.84 FCM 0.89 CM 0.87 SM 0.88 WSA 0.83 IDNN 0.96
Sajja et al [18]	CNN	Compared with AlexNet, GoogleNet and ResNet50.	LIDC	Model Sensitivity AlexNet 94.0% GoogleNet 98.0% ResNet50 92.0% Proposed 100.0%
Singh et al [19]	KNN, SVM, Decision Tree, Naïve Bayes, Stochastic Gradient Descent Classifier, Random Forest, MLP.	MLP resulted well in comparison with other for feature extraction.	CT scan images	Model Accuracy KNN 86.21% SVM 57.47% Decision Tree 81.77% Multinomial Naïve Bayes 57% Stochastic Gradient Descent 59.11% Random Forest 83.17% MLP 88.55%

3. Methodology Adapted

The research conducted a comparison between the effectiveness of the newly proposed neural network and transfer learning models. To enhance result visualization, images from the dataset underwent multiple stages, including image preprocessing, data balancing, and model training. After training the models, the outcomes were comprehensively discussed in comparison. The overall methodology is depicted in Fig 1. The pseudocode detailing the implementation is outlined below.

1. Importing Libraries - Import numpy, pandas, matplotlib, seaborn, cv2, random, os, imageio, plotly, collections, sklearn, tensorflow, keras.
2. Set Up Directories and Parameters – Set the path to image dataset and then defined the classes. Even set the image sizes.
3. Image Preprocessing –
   For each category in categories:
      Resize the image to 256*256.
      Apply Gaussian Filter.
      Print the Images.
4. Prepare data for training –
   Created empty lists
   For each category in categories:
      Loaded Images from categories
      Added Image Labels
   Separate the features for training and validation
5. Class Imbalance – Classes are Balanced to balance the minority classes using SMOTE.
6. Building Models - Built ImageTriNet and Transfer Learning Models.
7. Compiled the model using metrics as Accuracy.
8. Gathered the classification reports for evaluation.
9. Evaluation of the model was done on validation set using precision, recall, accuracy and F1-score.

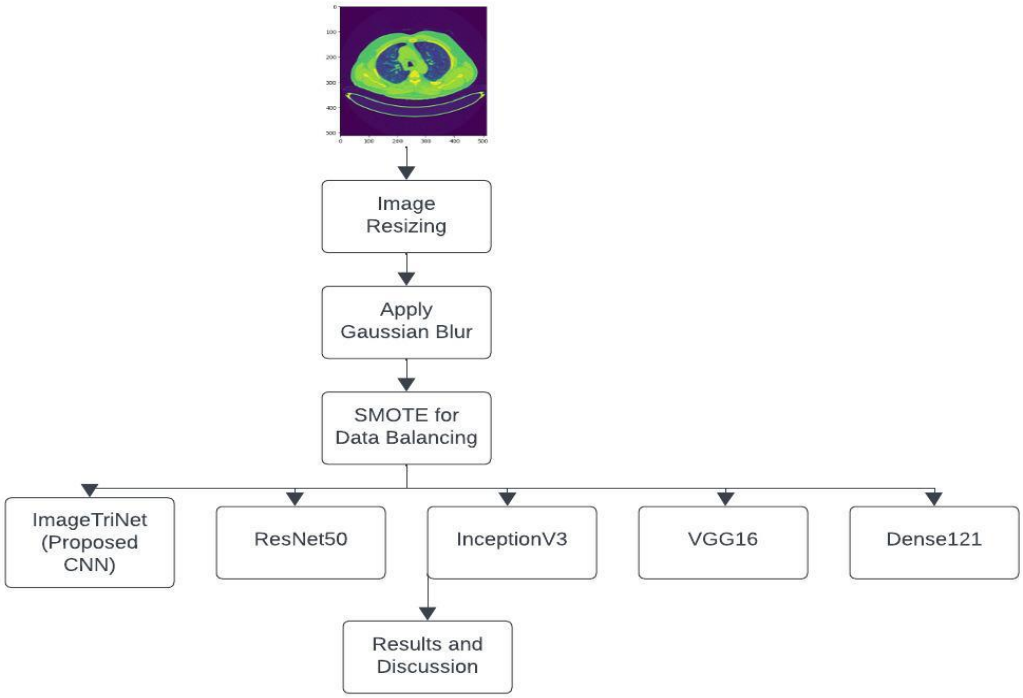

Figure 1. Proposed Methodology Adapted

3.1. Dataset Description

Fall 2019 at the National Center for Cancer Diseases/Iraq-Oncology Teaching Hospital (IQ-OTH/NCCD) compiled a dataset focused on lung cancer, featuring 1097 CT scan images and the image for each class is in Table 2. This dataset incorporates 110 cases, providing a comprehensive depiction

of the spectrum of lung cancer progression and including scans from both individuals with and without the condition. Sourced from specialized medical institutions, the dataset is categorized into three classes: normal (120 cases), benign (561 cases), and malignant (416 cases). The CT scans were acquired in DICOM format using Siemens SOMATOM, adhering to a standardized protocol with specific parameters such as 120 kV, 1 mm slice thickness, and defined window settings. Accessible on Kaggle, the dataset also includes a presentation of selected images from various cases, as displayed in Fig 2.

Table 2. Dataset Details

Case	Images
Bengin Cases (Class 0)	120
Malignant Cases (Class 1)	561
Normal Cases (Class 2)	416

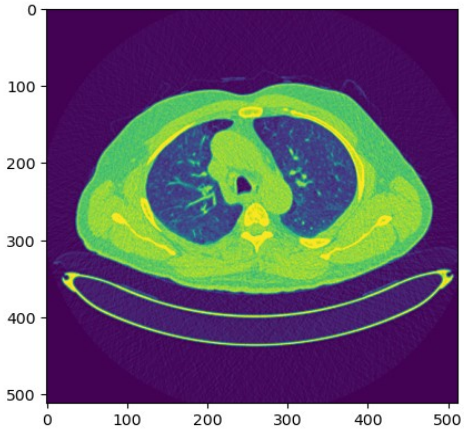

Figure 2. Sample Image of (512*512) size.

3.2. Image Preprocessing

3.2.1. Resizing Images

The dataset consists images which are of varying dimensions, specifically in the 'Benign cases,' where the sizes include '512 x 512' for 120 instances. In 'Malignant cases,' the dimensions exhibit greater diversity: '512 x 623' for 31 cases, '512 x 512' for 501 cases, '512 x 801' for 28 cases, and '404 x 511' for 1 case. For 'Normal cases,' there are '512 x 512' images for 415 cases and '331 x 506' for 1 case. To standardize the dataset, all images are resized [20] uniformly to dimensions of 256 x 256 which helps in identification of parameters with ease [21].

3.2.2. Gaussian Filter

Subsequently, a “Gaussian Blur” [22] enhancement was applied to the resized images. Gaussian blur is a preferred choice in computer vision due to its effectiveness in noise reduction and image smoothing. The equation 1 represents gaussian blur, where G(x) is gaussian function, x is distance from gaussian kernel and σ is standard deviation of the Gaussian Distribution.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \quad (1)$$

It excels at eliminating high-frequency details perceived as noise, enhancing the clarity of images for further processing. Notably, Gaussian blur stands out for its ability to maintain the integrity of edges in an image, ensuring a smooth effect while minimizing the risk of losing crucial features and boundaries [23,24]. This process aims to refine and optimize the images for further analysis, potentially improving feature extraction and aiding in the accuracy of subsequent tasks, such as classification or detection. The use of Gaussian Blur introduces a smoothing effect by reducing high-frequency noise, which may enhance the overall quality and interpretability of the images. This standardized and enhanced dataset is then poised for more effective utilization in tasks related to lung cancer image analysis, as displayed in Fig 3.

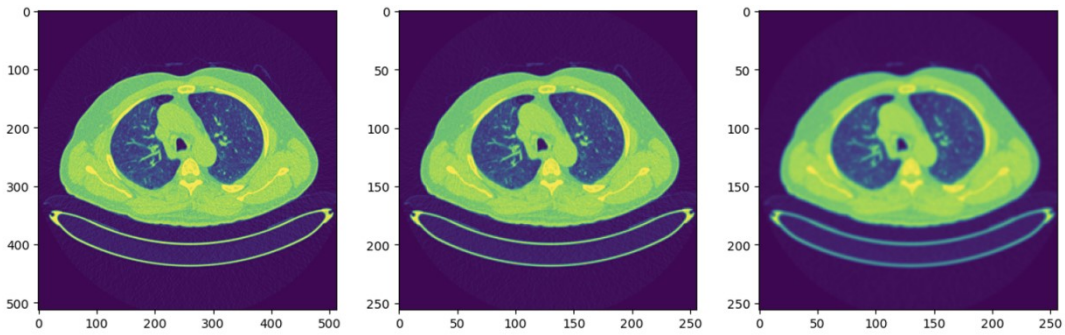

Figure 3. a) Original Image 512*512, b) Resized Image 256*256, c) Gaussian Blur image

Resizing images and subsequently normalizing [25] them is a crucial preprocessing step in machine learning, especially when training neural systems. This process's objective is to standardize and scale pixel values, creating a consistent input format. This not only aids in preventing issues such as exploding or vanishing gradients during training but also facilitates faster convergence of optimization algorithms. The resulting feature vector, expressed as a numpy array, serves as the normalized representation of the images. Beyond these technical advantages, normalization contributes to better generalization on new, unseen data by reducing sensitivity to variations in pixel values, thus enhancing the model's robustness to diverse lighting conditions and contrasts. Moreover, this normalization process acts as a form of regularization, potentially mitigating overfitting risks. By aligning input data to a common scale, the model becomes less prone to memorizing noise during training, ultimately improving its overall performance. The normalized images are calculated using equation 2.

$$X = X/255.0 \tag{2}$$

3.3. Data Imbalance

The training dataset comprises of training images of 822 images, while the validation dataset contains 275 images. A notable concern is the initial data imbalance, with class distribution delineated as 420, 312 and 90. To rectify this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied [26]. SMOTE tackles imbalanced datasets by oversampling the minority class, selecting a random instance and determining its k-nearest neighbors. Synthetic instances are then created through interpolation, blending the feature values of the chosen instance and its neighbors with random values between 0, 1 and 2. This method aims to balance class distribution and improve the model's capacity to learn from the minority class, addressing the challenges posed by imbalanced data. Mathematically, it is described in equation 3, where x_{new} is new synthetic feature, x_i feature vector of the minority class, x_{zi} is feature vector of one of its K-Nearest Neighbors and "λ" is a fictitious number between 0 and 1.

$$\mathrm{x_{new}} = \mathrm{x_i} + \lambda(\mathrm{x_{zi}} - \mathrm{x_i}) \tag{3}$$

Post-SMOTE implementation, the class distribution is rebalanced, resulting in a more equitable representation of 420 images for each class. This balanced distribution ensures each class is proportionately represented, mitigating potential bias stemming from the initial class imbalance. In the initial implementation, SMOTE was directly applied to the training data to address class imbalance. However, recognizing the potential pitfalls of applying oversampling naively in medical imaging, the approach was refined. Specifically, the training images were first reshaped into a controlled pixel-space representation before resampling, and the generated synthetic samples were then reshaped back into their original image dimensions. This ensured that SMOTE operated in a consistent feature space while preserving the structural integrity of the data for the convolutional model. The significance of incorporating SMOTE lies in its capacity to enhance predictive accuracy by generating synthetic samples for minority classes [27]. This approach contributes to a more resilient and balanced training dataset, preventing the model from leaning towards the majority class. Ultimately, the utilization of SMOTE optimizes the model's ability to correctly identify cases from every class in the dataset and enhances its generalization performance which is observable from Fig 4.

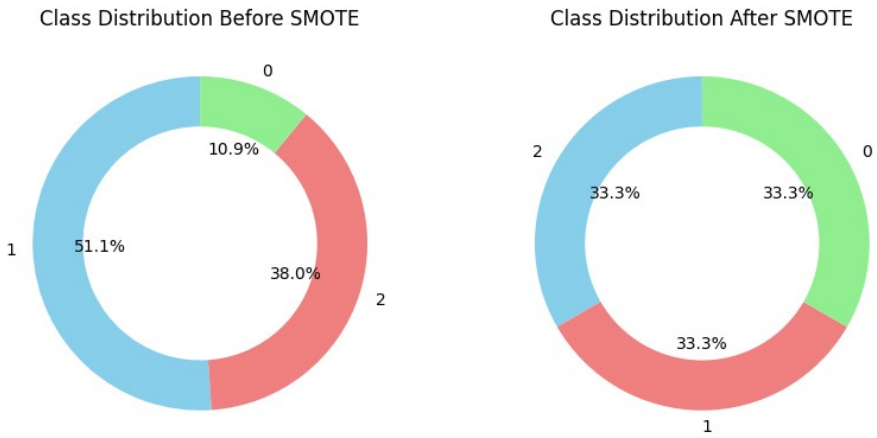

Figure 4. Class-wise data balanced using SMOTE.

3.4. Proposed Image Trainer Method

The ImageTrinet model is structured around its core component, the Trinet, which integrates three distinct convolutional layers followed by corresponding max-pooling operations. Following this convolutional-max-pooling sequence, the resultant features are flattened then injected into a dense layer. The last layer of the model is composed of three neurons, representing the three target classes. The model architecture involves a sequential arrangement of convolutional and pooling layers, culminating in a dense layer for classification across the specified classes. The algorithmic details for this architecture are provided.

ImageTriNet Algorithm:

Input:

-input_data = 4D_tensor(256, 256, 1);

Output:

-output_data = 1D_vector(3);

Preprocess Images:

-Total Images: 1097;

-Resizing Images;

-Applying Gaussian Blur;

Splitting the Data:

-The data is split into train and test;

-Training: X_train, X_valid;

-Validation: y_train, y_valid;

Model Creation:

-create_model(sequential)

Add Convolutional Layer:

-For i = 1 to 3:

add_convolutional_layer(model, filter_count = 2^i * 64, kernel_size = (3, 3), activation = 'relu')

if i == 1:

set_input_shape(model, X_train.shape[1:])

add_max_pooling_layer(model, pool_size=(2, 2))

Flatten Layer:

-flatten_output(model)

Fully Connected Layers:

-Flatten_output(model):

add_dense_layer(model, units = 512, activation = 'relu')

add_dense_layer(model, units = 3, activation = 'softmax')

Compilation of model:

-Compile using sparse_categorical_crossentropy

-Adam as Optimizer

Evaluation of the model using classification report.

The architecture of the CNN can be seen from Fig 5. Firstly, the processed images serve as input to the neural network. Starting with a Conv2D layer utilizing 64 filters sized 3x3, the initial layer produces an output shape of (254, 254, 64). The output can be computed using equation 4, where H

represents the output, X is the input, W represents the weights, b is the bias, and F denotes the filter size.

$$H_{i,j,k} = \sigma\left(\sum_{l=0}^{F-1}\sum_{m=0}^{F-1}\sum_{n=0}^{C_{in}-1} X_{i+l,j+m,n} \cdot W_{l.m.nk} + b\right) \quad (4)$$

Subsequently, a MaxPooling2D layer with a 2x2 pool size decreases spatial dimensions to (127, 127, 64). This calculation can be expressed using equation 5, where Y represents the output, X denotes the input feature map, and P signifies the pooling size specified.

$$Y_{i,j,k} = \max_{0\le l<p,0\le m<p} X_{p_{i+l},p_j+m,k} \quad (5)$$

Continuing the architectural design, a subsequent Conv2D layer integrates 128 filters, followed by another MaxPooling2D layer, resulting in an output shape of (125, 125, 128). This process is reiterated with a final Conv2D layer featuring 256 filters and a corresponding MaxPooling2D layer, generating an output shape of (60, 60, 256). The Flatten layer converts the three-dimensional output into a one-dimensional array sized 230,400. This flattened representation is subsequently connected to a Dense layer housing 512 neurons, establishing a fully connected layer. Dense layers can be understood using equation 6, where Y_k represents the output, W denotes the weight matrix, b signifies the bias, σ indicates the activation function, and M represents the number of neurons in the previous layer.

$$Y_k = \sigma\left(\sum_{l=1}^{M} W_{k,l} \cdot X_l + b_k\right) \quad (6)$$

The final Dense layer, comprising 3 neurons, serves as the output layer for the three-class classification task. Regarding model compilation, the architecture is set with the sparse categorical cross-entropy loss function, the Adam optimizer, and accuracy as the evaluation metric. The model features a considerable 118,336,515 trainable parameters, indicating its capability for intricate feature extraction and classification. Leveraging convolutional and pooling layers for hierarchical feature capture, followed by densely connected layers, this CNN architecture from Table 3 is poised for effective image classification in the specified context.

To improve the model's ability to identify diagnostically relevant features, a Convolutional Block Attention Module (CBAM) was integrated after each convolutional block. CBAM enhances the standard CNN pipeline by applying both channel and spatial attention: the channel attention mechanism selectively emphasizes feature maps that contribute most to discrimination, while the spatial attention mechanism highlights important regions within each feature map. This dual-attention design helps the network focus on subtle and clinically significant patterns in the images while suppressing background noise. The final model thus combines the strengths of conventional convolutional feature extraction with attention-based refinement, enabling a more precise and context-aware analysis of medical images. This lightweight modification enhances interpretability and robustness without introducing significant computational overhead, making it a practical yet powerful approach for classification tasks.

Table 3. Model Summary

Layer	Output Shape	Parameters
conv2d (Conv2D)	(254,254,64)	640
max_pooling2d (MaxPooling2D)	(127,127,64)	0
conv2d_1 (Conv2D)	(125,125,128)	73856
cbam_block_1	(125,125,128)	-
max_pooling2d_1 (MaxPolling2D)	(62,62,128)	0
Conv2d_2 (Conv2D)	(60,60,256)	295168
cbam_block_2	(60,60,256)	-
max_pooling2d_2 (MaxPooling2D)	(30,30,256)	0
Flatten	230400	0
dense (Dense)	512	117965312
dense_2 (Dense)	3	1539
Total Parameters: 134,600,968		
Trainable Parameters: 134,600,968		
Non – Trainable Parameters: 0		

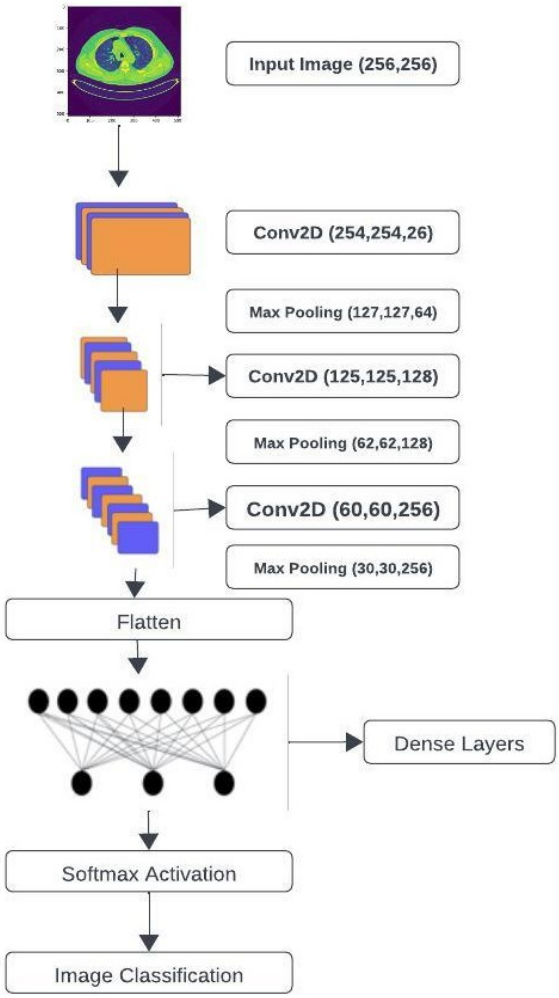

Figure 5. ImageTriNet Architecture

3.5. Transfer Learning Models

Furthermore, in addition to the proposed ImageTriNet, it is pertinent to compare it with established efficient models. To achieve this, transfer learning was employed as the primary approach due to its ability to provide a foundational framework and facilitate the comprehensive evaluation of ImageTriNet.

Throughout this investigation, several well-known models were selected for transfer learning, including ResNet50, InceptionV3, VGG16, and DenseNet121. These models have demonstrated efficacy across a spectrum of computer vision tasks, including image classification, object detection, and segmentation. A consistent methodology was applied across all chosen models, involving the freezing of all layers except the top layer. This strategic decision aimed to retain the valuable pre-trained knowledge embedded in the lower layers while enabling adaptation to the specific requirements of the target task. The common approach taken for each model involved customizing and training the top layers for a variable number of epochs, ranging from 20 to 100. This systematic and comparable training regimen was implemented to facilitate a thorough and fair evaluation of the performance of each transfer learning model.

The use of the Adam optimizer [28] remained consistent across all models, driven by its adaptive learning rate capabilities that dynamically adjust during training, promoting efficient convergence by accommodating varying gradients. Mathematically it is described in equation 7, where θ_t represents the model parameters, α is learning rate, m_t is the first moment estimate, ϑ_t is the second moment estimate and $\in$ is a constant.

$$\theta_{t+1} = \theta_t - \frac{\alpha m_t}{\sqrt{\vartheta_t + \epsilon}} \tag{7}$$

Similarly, a unified selection of the Categorical Cross-entropy loss function was adopted for all, specifically selected for its appropriateness in multiclass classification tasks where instances are

categorized into one of three or more classes. The average categorical cross entropy is calculated by using the equation 8. $\mathcal{L}(y, \hat{y})$ is categorical cross entropy loss, y is true probability distribution, y_i true probability of class i and $\hat{y}_i$ is predicted probability distribution for the class.

$$\mathcal{L}(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \quad (8)$$

This cohesive approach, employing the Adam optimizer and Categorical Cross entropy, ensures stable and effective training of the models, adhering to established best practices in deep learning for achieving optimal performance in multiclass classification scenarios.

3.5.1. ResNet50

ResNet50 [29] is recognized for its utilization of skip connections, a feature that enables the model to achieve greater depth without encountering degradation issues. Also, ResNet50 has demonstrated notable success in diverse computer vision tasks, particularly excelling in assignments like the classification of images. The pretrained model was enhanced by adding a flattening layer, a fully connected layer consisting of 128 neurons, and a Rectified Linear Unit (ReLU) activation to the base model. Additionally, a final output layer with three neurons and a softmax activation function was introduced for multiclass classification purposes. Following this, the model was compiled using the Adam optimizer and sparse categorical cross-entropy loss. A comprehensive performance assessment was conducted over 20 epochs, resulting in an impressive accuracy rate of 0.96. These elements encapsulate the key components of the transfer learning approach integrated into this model.

3.5.2. VGG16

VGG16 [30] is frequently chosen for transfer learning applications due to its simple yet effective architecture, designed to capture essential image features. In the development of the subsequent model, the VGG16 base was utilized in conjunction with custom classification layers. These additional layers consisted of a flattening layer followed by a densely connected layer containing 128 neurons with a ReLU activation function. The model was completed by incorporating an output layer containing three neurons along with a softmax activation function, enabling effective multiclass classification. Compilation of the model was performed using the Adam optimizer and sparse categorical cross-entropy loss. Following this, a rigorous training process comprising 100 epochs was conducted, culminating in an exceptional accuracy score of 0.99.

3.5.3. InceptionV3

The selection of InceptionV3 [31] is motivated by its utilization of inception modules, which incorporate multiple filter sizes in parallel. This architectural characteristic boosts the network's capacity to adeptly capture multi-scale features, making it well-suited for a variety of computer vision tasks. In the construction of the model, two densely connected layers were introduced, containing 256 and 128 neurons, respectively, and incorporating ReLU activation functions. This design was crafted to capture intricate patterns present within the data. To mitigate the risk of overfitting, dropout layers with a dropout rate of 0.2 were strategically placed after each densely connected layer. The final output layer was configured with three neurons and a softmax activation function to enable efficient multiclass classification. During the training process, which extended over 100 epochs, the model attained an accuracy level of 0.85.

3.5.4. Dense121

The decision to utilize DenseNet [32] architectures stems from their distinctive feature of dense connectivity, wherein each layer receives input from all preceding layers. This characteristic promotes effective feature reuse and ensures a smooth flow of gradients throughout the training process. In constructing the model, a solitary densely connected layer featuring 128 neurons and employing a ReLU activation function was incorporated to capture intricate patterns within the data. The final output layer was structured with three neurons and a softmax activation function, enabling efficient multiclass classification. During the model compilation phase, the “Adam optimizer” and “sparse categorical cross-entropy” loss were chosen, with accuracy serving as the designated metric for evaluation. The model was trained over 100 epochs, achieving an impressive accuracy score of 0.99.

4. Experimental Analysis of Models

The experimental analysis section delves into a thorough evaluation of the five compiled models, utilizing classification reports and confusion matrices to unravel their performance intricacies. These reports offer granular insights into precision, recall, and F1-score across diverse classes, shedding light on the models' proficiency in handling various categories. The accompanying confusion matrices provide a visual breakdown of correct and erroneous predictions, enriching the interpretation of the models' classification results. Beyond classification metrics, computational efficiency is assessed through the consideration of time elapsed, a key performance indicator. The collective assessment of accuracy, precision, recall, and F1-score ensures a thorough evaluation of the models' overall classification into 3 classes which are Benign, Malignant and Normal. The validation images are the same for all the models, as shown in Fig.6.

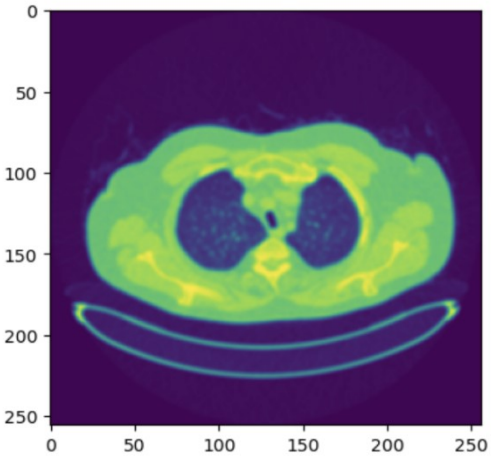

Figure 6. Validation Image

4.1. Results of ImageTriNet

The analysis of the ImageTriNet model over 13 epochs reveals consistent learning progress, with notable gains in both training and validation accuracy during the early stages. The updated classification report provides detailed insights: Class 0 (Benign) achieved a precision of 1.00 and a recall of 0.97, leading to an F1-score of 0.98. Class 1 (Malignant) performed exceptionally well with precision of 0.99 and perfect recall of 1.00, yielding an F1-score of 0.99. Class 2 (Normal) attained a precision of 0.99 and recall of 0.98, corresponding to an F1-score of 0.99. Overall, the model reached an accuracy of 0.99, supported by both macro- and weighted-average F1-scores of 0.99, as illustrated in the confusion matrix (Fig. 7). In terms of efficiency, the model maintained strong performance with an AUC close to 1 across all classes, while completing training in approximately 140 seconds. These results demonstrate robust classification ability and improved efficiency compared to the baseline reported in [33].

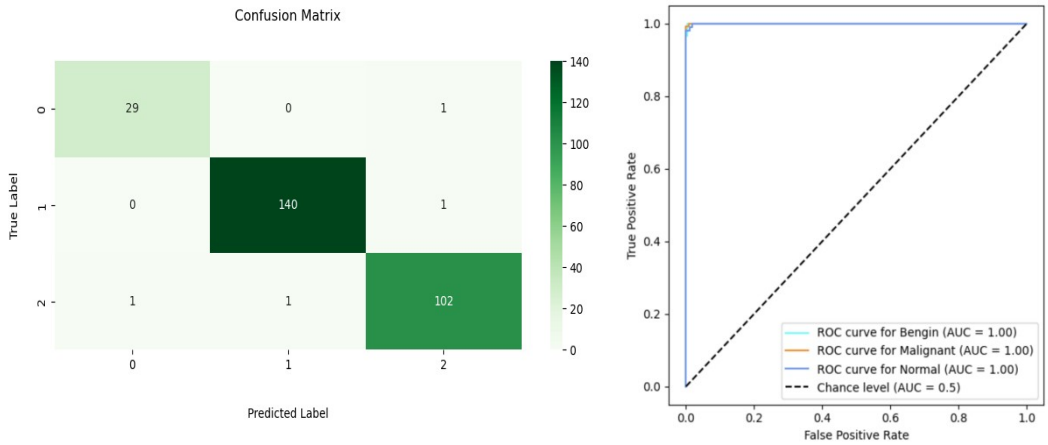

Figure 7. Performance evaluation of the ImageTriNet model

4.2. ResNet50

The training history of a learning model across 20 epochs reveals a consistent pattern of declining loss and escalating accuracy observed in both the training and validation sets. The proposed-model demonstrates efficiency in understanding complex data patterns, completing the entire training process in approximately 226 seconds. In the classification report, Classes 1 and 2 demonstrates admirable precision, recall, and F1-scores, emphasizing the model's accuracy in categorizing instances from these classes. However, Class 0 exhibits a slightly lower recall, highlighting an area for potential refinement to enhance the model's capability to recognize instances within this class. The confusion matrix from figure 6 provides a detailed breakdown of correct and incorrect classifications for each class. This excels in accurately categorizing instances from Classes 1 and 2, indicating proficiency. The same can be seen from AUC curve Fig 8. At the same time, it identifies opportunities for improvement, particularly in enhancing recall for Class 0.

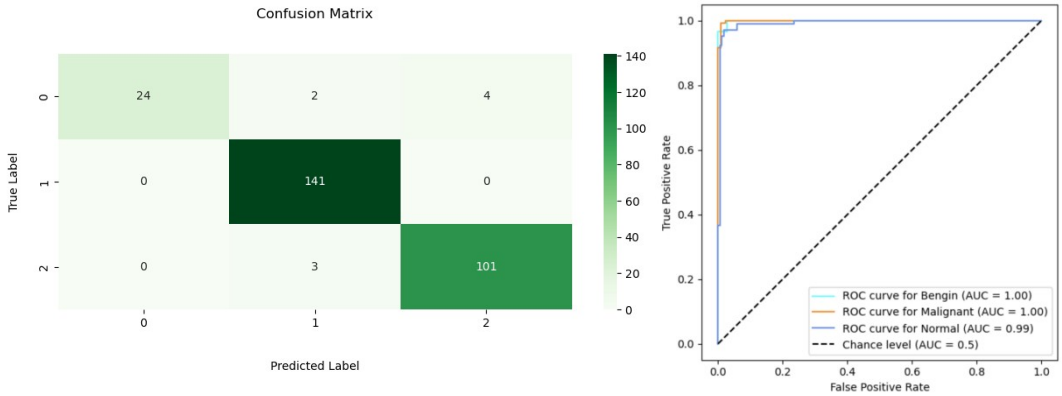

Figure 8. Performance evaluation of the ResNet50 model.

4.3. VGG16

The VGG16 learning model underwent an extensive 100-epoch training regimen, consistently reducing loss with an improving accuracy on both training and validation sets. Notably, total training time was approximately 267 seconds, showcasing the model's efficiency in understanding complex data patterns over an extended duration. In the classification report, Classes 1 and 2 demonstrate commendable “precision”, “recall”, and “F1-scores”, reflecting the proficiency in categorizing instances from these classes. However, Class 0 shows a slightly lower recall, suggesting room for improvement to enhance the model's ability to identify instances in this class. The confusion matrix from (Fig. 9) provides insights into correct and incorrect classifications across classes. VGG16 excels in accurately categorizing instances from Classes 1 and 2, highlighting its proficiency. Simultaneously, the matrix points to opportunities for improvement, particularly in enhancing recall for Class 0.

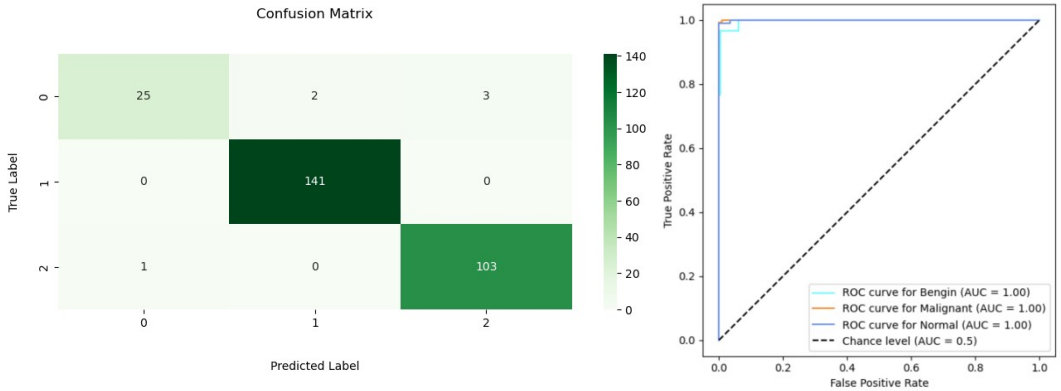

Figure 9. Confusion Matrix (left) and ROC-AUC of VGG16.

4.4. InceptionV3

The assessment of your trained model, as indicated by the classification report and confusion matrix, offers a comprehensive insight into its strengths and areas for enhancement. This model translates

impressive precision in identifying instances from Class 1, achieving a rate of 95%. However, precision is notably lower for Class 0 at 43%, suggesting a higher likelihood of false positives in this category. The recall metrics highlight perfect identification for Class 1 but indicate a lower recall of 33% for Class 0, implying potential oversight of instances from this class. Correspondingly, F1-scores reinforce these findings, emphasizing high performance for Class 1 and a need for refinement in Class 0.

A closer examination of the confusion matrix reveals specific misclassifications, particularly between Classes 0 and 2, and instances where Class 0 is wrongly assigned as Class 2 which can be seen in the Fig 10 which is confusion matrix of InceptionV3. While the overall accuracy is commendable at 85%, the analysis underscores the importance of addressing class-specific performance variations to ensure a more balanced and accurate model. The recommendation for fine-tuning is particularly relevant, with a focus on enhancing precision and recall, especially for Class 0, to further elevate the overall effectiveness of the model. The time elapsed for training is approximately 1408 seconds.

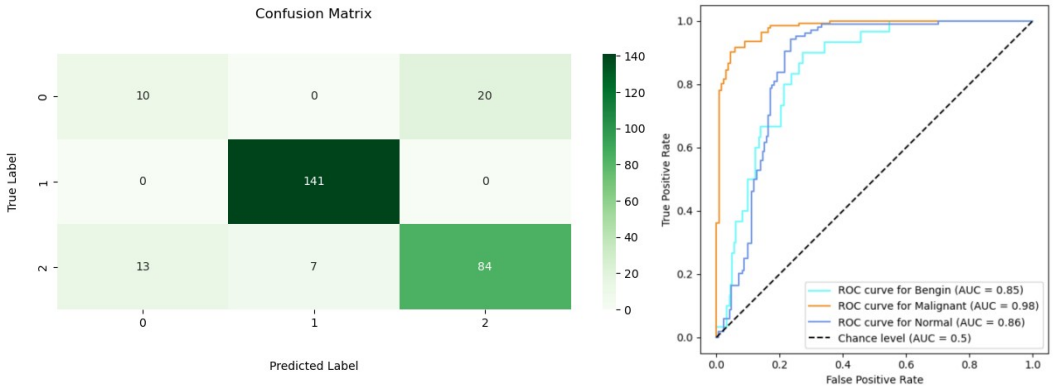

Figure 10. Performance evaluation of InceptionV3 model.

4.5. Dense121

The model's assessment, as presented in the classification report and confusion matrix, underscores its outstanding performance in accurately categorizing instances among the three classes. A detailed analysis of the classification report reveals impressive precision, recall, and F1-Score metrics for each class. In the case of Class 0, the model achieved a precision of 0.94, a recall of 1.00, and an F1-Score of 0.97 based on 30 instances. Class 1 exhibited exceptional precision (0.99), recall (1.00), and F1-Score (1.00) with a support of 141 instances. For Class 2, comprising 104 instances, the model demonstrated a precision of 1.00, a recall of 0.97, and an F1-Score of 0.99. The overall accuracy reached an impressive 99%, highlighting the model's proficiency in classification tasks. The confusion matrix reinforces these results, depicting minimal false positives and false negatives across all classes as seen in Fig 11, further affirming the model's robust performance. The time elapsed for 100 epochs is 1700 seconds.

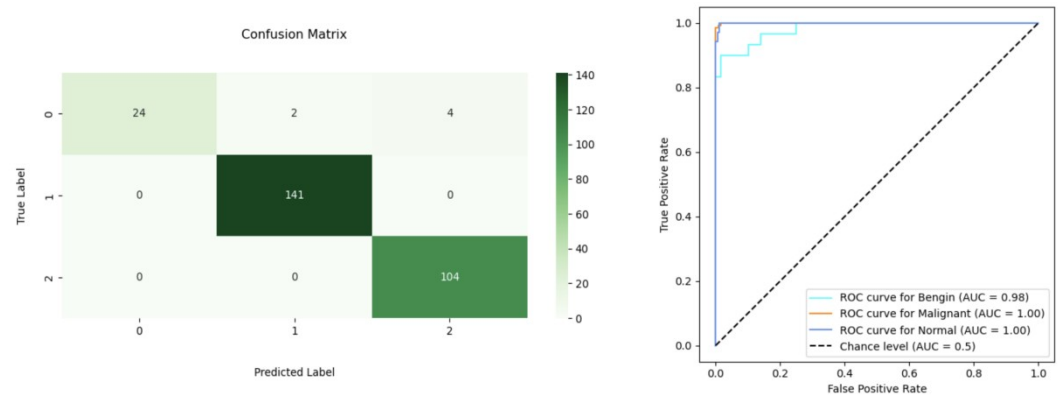

Figure 11. Performance evaluation of Dense121 model.

The overall performance of various neural networks on various metrics are depicted in Fig 12.

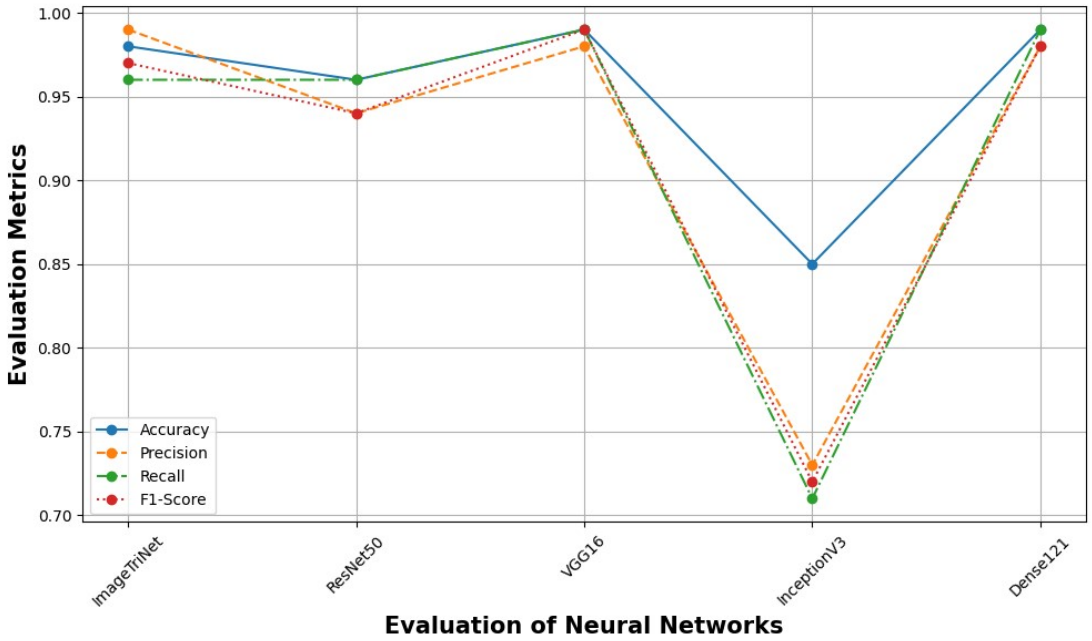

Figure 12. Evaluation of Neural Networks.

5. Results and Discussion

The ImageTriNet and transfer learning models demonstrated superior outcomes compared to other models that adopted various methodologies, as depicted in Table 4. On a whole, ImageTriNet demonstrated an impressive overall accuracy of 99%, showcasing notable precision, recall, and F1-scores for all three classes (0, 1, and 2). Despite a slightly lower recall in class 0, the model maintained robust performance, evidenced by high F1-scores. ResNet50 exhibited commendable overall accuracy at 96%, with its outlaying performance in classes 1 and 2 for all the performance metrics. Although, for benign cases displayed a slightly lower precision, the model showcased reliable performance across the majority of classes. VGG16 stood out with exceptional performance, achieving an overall accuracy of 99%. The model excelled in precision, recall, and F1-scores for all classes, underscoring its effectiveness in classification tasks. InceptionV3, with an overall accuracy of 85%, demonstrated varied performance across classes. While class 1 exhibited excellent precision, recall, and F1-score, class 0 showed lower values, influencing the overall performance.

Dense121 mirrored the excellence of VGG16, attaining an impressive overall accuracy of 99% and displaying high precision, recall, and F1-scores across all classes. This analysis underscores the superior performance of ImageTriNet, VGG16, and Dense121 models, with ResNet50 also proving to be a strong contender. InceptionV3, while achieving a lower accuracy, showcased distinct performance characteristics that may be considered based on specific application requirements which can be viewed from table 4.

Table 4. Model Comparison and Analysis

Model	Epochs	Time Taken(sec)	Accuracy	Precision	Recall	F1-Score	Mean absolute error
ImageTriNet	13	223	0.99	0.99	0.96	0.97	2.33
ResNET50	20	226	0.96	0.94	0.96	0.94	3
VGG16	100	267	0.99	0.99	0.98	0.99	0.67
InceptionV3	100	1408	0.85	0.73	0.71	0.72	31.33
Dense121	100	1700	0.99	0.98	0.99	0.98	1.33

Additionally, the Mean Absolute Error (MAE) plays a crucial role in assessing the predictive accuracy of diverse models. Among the models under scrutiny, VGG16 emerges as the standout performer, demonstrating the smallest MAE at 0.67. This low MAE signifies minimal disparities between predicted and actual values, affirming VGG16's efficacy. Conversely, Inceptionv3 encounters challenges apparent in its elevated MAE of 31.33, indicative of notable deviations in predictions. These unique MAE insights offer a nuanced comprehension of each technique’s strengths and weaknesses,

aiding in the informed selection of the most precise and dependable predictive model for specific applications.

From Table 5, it becomes evident that the ImageTriNet establishes itself as a robust contender. Boasting an accuracy of 99% following 13 epochs and a relatively brief training duration of 223 seconds, the model not only showcases efficiency but also proves highly effective. The precision and recall metrics of 99% and 96%, respectively, contribute to a well-balanced F1-Score of 0.97. This showcases that the ImageTriNet not only attains an impressive overall accuracy but also strikes a harmonious equilibrium between correctly identifying positive instances and minimizing false positives.

Table 5. Existing Methods Results

Paper	Methodology Adapted	Accuracy
Zhang et al [33]	CNN	0.96
Reddy et al [34]	ResNET50	0.76
Tekade et al [35]	VGG16	0.95
AL-Huseiny et al [36]	InceptionV3	0.94
Pang et al [37]	Dense121	0.80
Proposed	CNN	0.99

In a similar vein, VGG16 emerges as another standout performer, touting an exceptional accuracy of 99% after 100 epochs. Despite the extended training time of 267 seconds, VGG16 consistently exhibits remarkable “precision”, “recall”, and “F1-Score”, all reaching 0.99, 0.98, and 0.99, respectively. The model's robust performance across these metrics underscores its efficacy in effectively handling the classification task. On a different note, while ResNET50 and Dense121 also demonstrate commendable results with accuracies of 96% and 99%, respectively, each excelling in specific aspects, InceptionV3 lags behind. It exhibits a lower “accuracy” of 85% and inferior precision, recall, and F1-Score, indicating potential limitations in its suitability for the given classification task.

Table 6. Comparative performance of ImageTriNet and baseline CNN models

Model	Accuracy	95% CI (Lower–Upper)	Cohen’s h vs. ImageTriNet	Cohen’s h vs. ResNet50	Cohen’s h vs. VGG16	Cohen’s h vs. InceptionV3	Cohen’s h vs. Dense121
ImageTriNet	0.989	0.968 – 0.998	0.000	0.155	0.088	0.571	0.088
ResNet50	0.967	0.939 – 0.985	-0.155	0.000	-0.068	0.416	-0.068
VGG16	0.978	0.953 – 0.992	-0.088	0.068	0.000	0.484	0.000
InceptionV3	0.855	0.807 – 0.894	-0.571	-0.416	-0.484	0.000	-0.484
Dense121	0.978	0.953 – 0.992	-0.088	0.068	0.000	0.484	0.000

To further examine the relative performance of the evaluated architectures, we compared model accuracies as shown in Table 6. alongside their 95% confidence intervals and quantified differences using Cohen’s h effect size. The proposed Base Model (ImageTriNet) achieved the highest accuracy at 98.9% (95% CI: 0.968–0.998), closely followed by VGG16 (97.8%, CI: 0.953–0.992) and DenseNet121 (97.8%, CI: 0.953–0.992). ResNet50 performed slightly lower at 96.7% (CI: 0.939–0.985), while InceptionV3 lagged behind with 85.5% (CI: 0.807–0.894). Effect size comparisons revealed that differences between ImageTriNet and VGG16 or DenseNet121 were minimal (Cohen’s h $\approx$ 0.088, indicating negligible effect), while the gap between ImageTriNet and ResNet50 was small (h = 0.155). In contrast, comparisons involving InceptionV3 yielded large effect sizes (h ranging from 0.416–0.571), confirming its significantly weaker performance relative to the other models. These results indicate that ImageTriNet, VGG16, and DenseNet121 provide statistically comparable high performance, with ImageTriNet achieving the most consistent results, while InceptionV3 demonstrates clear underperformance.

6. Conclusion and Future Scope

In summary, the findings underscore the superior performance of ImageTriNet, VGG16, and Dense121 models when compared to alternative methodologies. Notably, ImageTriNet achieved an impressive overall accuracy of 98%, while VGG16 and Dense121 demonstrated exceptional accuracy rates of 99%. ResNet50 also emerged as a robust contender with an overall accuracy of 96%. Despite InceptionV3's lower accuracy at 85%, its distinctive performance characteristics suggest potential

applicability in specific scenarios. In conclusion, these findings offer valuable perspectives on the relative performance of diverse models within the study's context. From the above training, it would be valuable to explore potential optimizations. Implementing techniques like data augmentation or fine-tuning could prove effective in mitigating overfitting issues that may arise in later epochs. To ensure the sustained high accuracy of the model, it is advisable to regularly monitor its performance and periodically retrain it with new data. A focused effort on understanding and optimizing the overall training time is essential, as it plays a critical role in the model's efficiency and effectiveness. By incorporating these strategies, the model can be fine-tuned to address specific challenges and maintain its accuracy over an extended period. While Gaussian blur was applied in our preprocessing pipeline to mitigate noise, we acknowledge that its potential to obscure fine diagnostic details is a valid concern. Future work will include a comparative evaluation with alternative enhancement techniques (e.g., median filtering, anisotropic diffusion, or wavelet-based denoising) to ensure that critical medical features are preserved while maintaining robustness against noise. We acknowledge that validating the model on additional public lung CT datasets (e.g., LIDC-IDRI, LUNA16) would strengthen generalizability. However, due to computational resource limitations, this was not feasible in the current study. Future work will explore external validation using these datasets as resources permit.

References

1. Sun W, Zheng B, Qian W (2016) "computer aided lung cancer diagnosis with deep learning algorithms" International Society for Optics and Photonics, medical imaging : computer-aided diagnosis. Vol. 9785
2. Zhou Z-H, Jiang Y, Yang Y-B, Chen S-F (2002) Lung cancer cell identification based on artificial neural network ensembles', Elsevier. Artif Intell Med 24:25–36
3. Nie L, Wang M, Zhang L, Yan S, Zhang B, Chua T-S (2015) Disease inference from health-related questions via sparse deep learning. IEEE Trans Knowl Data Eng 27(8):2107–2119
4. Nie L, Zhang L, Yang Y, Wang M, Hong R, Chua T-S (2015) Beyond doctors: future health prediction from multimedia and multimodal observations. proceedings of the 23rd ACM international conference on multimedia.
5. Dhaware BU, Pise AC, (2016) Lung Cancer Detection Using Bayasein Classifier and FCM Segmentation. IEEE, International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 170–174
6. da Silva GLF, de Carvalho Filho AO, Silva AC, de Paiva AC, Gattass M (2016) Taxonomic indexes for differentiating malignancy of lung nodules on CT images. Research on Biomedical Engineering 32(3):263–272
7. Park SC, Tan J, Wang X, Lederman D, Leader JK, Kim SH, Zheng B (2011) Computer-aided detection of early interstitial lung diseases using low-dose CT images', Iop Publishing. Phys Med Biol 56:1139–1153. https://doi.org/10.1088/0031-9155/56/4/016
8. Song QZ, Zhao L, Luo XK, Dou XC (2017) Using deep learning for classification of lung nodules on computed tomography images. Journal of healthcare engineering. https://doi.org/10.1155/2017/8314740
9. Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 first international conference on secure cyber computing and communication (ICSCCC) (pp. 278-282). IEEE
10. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76.
11. Al-Ameen, Z., Sulong, G., Gapar, M. D., & Johar, M. D. (2012). Reducing the Gaussian blur artifact from CT medical images by employing a combination of sharpening filters and iterative deblurring algorithms. Journal of Theoretical and Applied Information Technology, 46(1), 31-36.
12. Abhir Bhandary , G. Ananth Prabhu , V. Rajinikanth , K. Palani Thanaraj , Suresh Chandra Satapathy , David E. Robbins , Charles Shasky , YuDong Zhang , Joao Manuel R.S. Tavares , N. Sri Madhava Raja ,

DeepLearning Framework to Detect Lung ˜ Abnormality – A study with Chest X-Ray and Lung CT Scan Images, Pattern Recognition Letters (2019), doi: https://doi.org/10.1016/j.patrec.2019.11.013

13. Kaur, S., Hooda, R., Mittal, A., Akashdeep, & Sofat, S. (2017). Deep CNN-Based Method for Segmenting Lung Fields in Digital Chest Radiographs. Advanced Informatics for Computing Research, 185– 194. doi:10.1007/978-981-10-5780-9_17.
14. R. Tekade and K. Rajeswari, "Lung Cancer Detection and Classification Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697352.
15. Chaunzwa, T.L., Hosny, A., Xu, Y. et al. Deep learning classification of lung cancer histology using CT images. Sci Rep 11, 5471 (2021). https://doi.org/10.1038/s41598-021-84630-x
16. Xie, Y.; Zhang, J.; Xia, Y.; Fulham, M.; Zhang, Y. Fusing texture, shape and deep model-learned information at decision level for automated classification of lung nodules on chest CT. Inf. Fusion 2018, 42, 102–110.
17. Shakeel, P.M., Burhanuddin, M.A. & Desa, M.I. Automatic lung cancer detection from CT image using improved deep neural network and ensemble classifier. Neural Comput & Applic 34, 9579–9592 (2022). https://doi.org/10.1007/s00521-020-04842-6.
18. Sajja, T., Devarapalli, R., & Kalluri, H. (2019). Lung Cancer Detection Based on CT Scan Images by Using Deep Transfer Learning. Traitement du Signal, 36(4), 339-344.
19. Singh, G.A.P., Gupta, P.K. Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans. Neural Comput & Applic 31, 6863–6877 (2019). https://doi.org/10.1007/s00521-018-3518-x
20. W. Kim and C. Kim, "A novel image importance model for content-aware image resizing," 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 2011, pp. 2469-2472, doi: 10.1109/ICIP.2011.6116161.
21. Saponara, S., & Elhanashi, A. (2021, September). Impact of image resizing on deep learning detectors for training time and model performance. In International Conference on Applications in Electronics Pervading Industry, Environment and Society (pp. 10-17). Cham: Springer International Publishing.
22. Kostková, J., Flusser, J., Lébl, M., & Pedone, M. (2020). Handling Gaussian blur without deconvolution. Pattern Recognition, 103, 107264.
23. Liu, Y. Q., Du, X., Shen, H. L., & Chen, S. J. (2020). Estimating generalized gaussian blur kernels for out-of-focus image deblurring. IEEE Transactions on circuits and systems for video technology, 31(3), 829-843.
24. Gu, C., Lu, X., He, Y., & Zhang, C. (2020). Blur removal via blurred-noisy image pair. IEEE Transactions on Image Processing, 30, 345-359.
25. Tellez, D., Litjens, G., Bándi, P., Bulten, W., Bokhorst, J. M., Ciompi, F., & Van Der Laak, J. (2019). Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. Medical image analysis, 58, 101544.
26. Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems.
27. Liu, J. (2022). Importance-SMOTE: a synthetic minority oversampling method for noisy imbalanced data. Soft Computing, 26(3), 1141-1163.
28. Tutunov, R., Li, M., Cowen-Rivers, A. I., Wang, J., & Bou-Ammar, H. (2020). Compositional adam: An adaptive compositional solver. arXiv preprint arXiv:2002.03755.
29. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

30. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
32. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
33. Zhang, C., Sun, X., Dang, K., Li, K., Guo, X. W., Chang, J., ... & Zhong, W. Z. (2019). Toward an expert level of lung cancer detection and classification using a deep convolutional neural network. The oncologist, 24(9), 1159-1165.
34. Reddy, S., Bhuvaneshwari, V., Tikariha, A. K., Amballa, Y. S., & Raj, B. S. S. (2022). Automatic pulmonary nodule detection in ct scans using xception, resnet50 and advanced convolutional neural networks models. Int Res J EngTechnol, 9(4), 3226-3237.
35. R. Tekade and K. Rajeswari, "Lung Cancer Detection and Classification Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697352.
36. AL-Huseiny, M. S., and Sajit, A. S. 2021. Transfer learning with googlenet for detection of lung cancer. Indonesian Journal of Electrical Engineering and Computer Science 22(2):1078–1086.
37. Pang, S., Fan, M., Wang, X., Wang, J., Song, T., Wang, X., & Cheng, X. (2020). VGG16-T: a novel deep convolutional neural network with boosting to identify pathological type of lung cancer in early stage by CT images. International Journal of Computational Intelligence Systems, 13(1), 771.