RESEARCH PAPER

A Hybrid Texture-Guided Learning Approach for Lung Cancer Segmentation and Classification from CT Images

Shashank Yadav[1,2]* and Upendra Kumar[3]

[1]*Department of Computer Science & Engineering, Institute of Engineering & Technology, Lucknow Affiliated with Dr. A. P. J. Abdul Kalam Technical University, Lucknow, India,*
[2]*Assistant Professor, Department of Information Technology, Krishna Institute of Engineering & Technology (KIET), Ghaziabad, India,*
[3]*Department of Computer Science & Engineering, Institute of Engineering & Technology, Lucknow, India*
[1]**shashankyadav48@gmail.com, [2]2410@ietlucknow.ac.in and [3]ukumar@ietlucknow.ac.in*

Received: 14th June, 2026; Revised: 20th July 2026; Accepted: 20th August, 2026; Available Online: 05th September 2026

ABSTRACT
Lung cancer is the key cause of death for mankind worldwide. Indications of lung cancer generally are not seen in primary stages. For identification of cancer at initial stage, manually medical imaging segmentation is a tedious task. For sorting out the issue, this work suggests creating a comprehensive method for recognizing the lung cancer in CT scan images at initial stage. With suitable segmentation technique, suitable features should be extracted. In this work, Watershed segmentation technique is applied on Data Science Bowl 2017 dataset for segmenting lung CT images and different types of features are taken out utilizing different ways such as Gray-Level Co-occurrence Matrix, Gray Level Size Zone Matrix, Gray Level Run Length Matrix, and Neighborhood Gray Tone Difference Matrix. The purpose of extracting different types of features is to check which features set will be more suitable for proper classification of lung images. For classification, Random Forest technique is utilized with all extracted features separately and results are compared among these features set. By comparing the result we conclude that Random Forest technique performs best with Neighborhood Gray Tone Difference Matrix features set. Data augmentation enhances training datasets by applying transformations to existing images. We processed data augmentation method on the same dataset and this augmented dataset is applied on two deep learning models hybrid VGG-16 and hybrid DenseNet201 with Adam and SGD optimizers. Finally conclusion is that the deep learning model DenseNet201 with SGD optimizer gives 99% accuracy on training data.

Keywords: *Watershed, Random Forest, VGG-16, DenseNet201, Adam, SGD*
How to cite this article: Yadav S and Kumar U, A Hybrid Texture-Guided Learning Approach for Lung Cancer Segmentation and Classification from CT Images. Int J Drug Deliv Technol. 2026;16(7): 1322-1342; DOI: 10.25258/ijddt.16.7.143
Source of support: Nil.
Conflict of interest: None

Abbreviation

CT	Computed Tomography
GLCM	Gray-Level Co-occurrence Matrix
GLRLM	Gray Level Run Length Matrix
GLSZM	Gray Level Size Zone Matrix
NGTDM	Neighborhood Gray Tone Difference Matrix
VGG	Visual Geometry Group
SGD	Stochastic Gradient Descent
GLOBOCAN	Global Cancer Statistics
UNETR	UNEt TRansformers

1. INTRODUCTION AND LITERATURE SURVEY

Cancer is increasingly becoming one of the major reasons of demise of humankind worldwide. As stated in the most recent data from GLOBOCAN, globally, 19.3 million fresh cases of cancer were identified. In 2020 alone, 9.96 million people lost their lives to cancer [1]. Segmentation of lung cancer is a important zone of research, with many studies conducted in this field. Various treatment methods have been established to manage malignant tumors and improve the lifestyle of cancer patients. Other than surgical intervention, these methods include chemotherapy, radiation, thermotherapy, and immunotherapy. Deep learning techniques have newly shown exceptional performance across different fields of artificial intelligence and computer vision. Their application in the medical field, particularly in organ cancer segmentation, has seen significant advancements. These models have proven highly effective in automating

**Author for Correspondence: shashankyadav48@gmail.com*

the segmentation of medical images by autonomously learning feature representations and leveraging high-dimensional abstractions to finish segmentation processes without manual input. However, one common challenge in applying deep learning architectures is the need for large datasets, as small datasets often fail to yield optimal results when using these models.

Said et al. [1] suggested a model for identifying the lung cancer. They established segmentation method by using UNETR network in their first section of the system, and classification method to classify the segmented result in their second part of the system. They used 3D-input CT scan dataset for recognizing lung cancer in their powerful tool. They succeeded with 97.83% segmentation accuracy and 98.77% classification accuracy. Riaz et al. [2] proposed an advanced modified neural network for the segmenting malignant lung tumors semantically from images of CT scan by using combination of two models MobileNetV2 and UNet. They trained the model by using dataset from the Medical Segmentation Decathlon (MSD) 2018 Challenge and tested model on 25% of the dataset. As a result they found 0.8793 dice score. Abdillah et al. [3] utilized Marker Control Watershed and Region Growing method for segmentation of CT images. They also enhanced the images by using technique like Gabor filter. Kumar et al. [4] worked on five methods for optimization like particle swarm optimization, k-median clustering etc. to detect the tumor from image of lung. They investigated all methods performance and decided that Guaranteed Convergence Swarm Optimization (GCPSO) was the best optimization method with 95.89% accuracy. Joon et al. [5] proposed the system that processes acquired lung X-ray images, starting with noise detection through a median filter in the preprocessing step. Next, image segmentation is performed, followed by feature extraction with the help of K-means as well as fuzzy C-means clustering. This work extends existing image processing techniques for lung cancer detection, producing final feature extraction results after segmenting X-ray images. This approach is established by applying the Support Vector Machine (SVM) method for classifying the images.

Nadkarni et al. [6] suggested self-operating technique for detecting cancer in lung CT scan images. The proposed method involves image preprocessing using median filtering, trailed by segmenting the lung area of interest through mathematical morphological operations. After then geometrical features are calculated from the segmented area and utilized to categorize the images as affected or unaffected using classification technique like support vector machine (SVM). Uzelaltinbulat et al. [7] presented an approach for diagnosing of medical image aimed at segmenting lung tumors in CT images, addressing the shortage of such algorithms and methodologies for tumor detection. Mostly research in this area utilizes machine learning to tackle the complications of segmentation. The effort employs various image processing tools, which, when combined and applied successively, successfully achieve the desired outcomes. The segmentation system progresses through several stages to reach its goal of segmenting the lung tumor. Initially, image preprocessing is performed, incorporating enhancement techniques to improve image quality and decrease noise. After that different image regions are separated to facilitate tumor segmentation. Automatic thresholding is applied to ensure correct selection across all images, accounting for the varying gray-level intensities of the tumor. An additional technique is then utilized to isolate the tumor from the thresholded image. Ultimately, the lung tumor is exactly segmented by deducting the thresholded image from the processed image.

Primakov et al. [8] have designed a model for identifying and volumetrically segmenting non-small cell lung cancer (NSCLC) utilizing 1,328 lung CT scans which were taken from eight organizations. The model is quicker and more reproducible than professional assessments, with 56% of cases preferred by radiologists and radiation oncologists. Tumor volume measurements and the RECIST criteria are used to evaluate the prognostic value of the automated contours. Compared to hand outlines, the method's segmentations more statistically significantly divide patients into groups with low and high survival rates. An algorithm was proposed by Shimazak et al. [9] based on deep learning to distinguish lung cancer chest radiographs by segmentation operation. They utilized training and test datasets from a hospital between 2006 and 2018. The method's sensitivity and median false positive indications per image were calculated utilizing an independent test dataset. This method found a sensitivity of 0.73 with an mFPI of 0.13 during testing step. Sensitivity was lesser for lung cancers overlapping blind spots (0.50-0.64) than non-overlapping areas (0.87). The average dice coefficient was 0.52 for the 159 malicious lesions. Finally, the DL-based model effectively identified lung cancers with less mFPI.

Miah et al. [10] proposed system involves several steps, including acquiring image, preprocessing, converting image into binary image, thresholding, segmenting image, extracting features etc. Initially, input lung CT images were processed using various image processing methods. In the first phase, the image was converted into a binary format, which was then compared to a threshold value to identify lung cancer. In the second phase, segmentation was executed on the CT images, followed by a robust process to take out key features from the segmented images. These features were provided to train a neural network, which is then tested to distinguish between images that are cancerous and those that are not. The system demonstrates satisfactory performance, achieving an accuracy of 96.67%. In order to separate the cancerous lung regions from the adjacent chest region, Cifci [11] presented the integration of U-Net with a channel attention module (CAM). The suggested SegChaNet method used a number of encoders to transform CT slices of the lungs into feature maps. Furthermore, a multi-scale approach was created expressly to extract dense features at various

scales from the encoded feature maps. The lungs segmentation map was then generated by the decoders, and SegChaNet technique was matched with state-of-the-art techniques. Iterative down-sampling and up-sampling made sure the network was constant regardless of the magnitude of dense abnormalities, and the method successfully learned extracted dense features for lung problems. The suggested method worked accurately and efficiently according to experiment results, providing precise lung region segmentation in complex scenarios without the need for post-processing.

Different researchers used different types of classification techniques with various types of texture features. Patil et al. [12] also used SVM classifier for lung cancer identification by utilizing features from Gray-Level Co-occurrence Matrix (GLCM). Vironicka et al. [13] used the four different classifiers Random Forest, SVM, Decision tree, and Logistic Regression with GLCM features of X-rays images of bones for identifying abnormalities. Muthukumaran et al. [14] worked with Probabilistic Neural Network to classify the brain images' tumor. The network was fed with GLRLM texture features. Thibault et al. [15] presented a study for classification of cells nuclei for diagnosing Progeria disease affected patients. For this classification they used GLSZM texture features. Chen et al. [16] calculated the discriminant power of dual time point imaging (DTPI) PET/CT in distinguishing between benign and malignant FDG-avid solitary lung nodules using NGTDM texture characteristics.

Jiang et al. [17] suggested Improved Visual Geometry Group-13 that is an improved VGG16 method for classifying pneumonia X-rays images. They used augmentation methods to improve unbalanced data. They trained original dataset as well as augmented dataset by using LeNet Alexnet, GoogleNet and VGG convolutional neural networks. They found that process with augmented dataset improved recall, precision, and F1-score values. Liu et al. [18] recommended a model for classifying the breast X-ray mammography images. The model was built upon VGG16 convolutional neural network. By this model, they improved classification effectiveness compared to previous models of this area. In order to develop a method for identifying and categorizing heel bone disorders deprived of the need for data augmentation. Taher et al. [19] suggested an enhanced capsule network that combines an improved DenseNet201 with the main structure. Benhassine et al. [20] studied the potential impact of data augmentation on the performance of transfer learning models in medical imaging. The overall accuracy of the suggested network architecture was 94.5%.

In this research, Random Forest machine learning technique and deep learning models hybrid VGG-16 and hybrid DenseNet201 are applied on the Data Science Bowl 2017 dataset and their results are compared. The Watershed algorithm is utilized for segmenting lung images after preprocessing steps. After segmenting the lung CT images, different types of features GLCM, GLRLM, GLSZM, and NGTDM method are applied on segmented lung images. All features set are fed into Random Forest method one by one for classification. After calculating classification results, we find that NGTDM features work well with Random Forest method than other features. After that we used same dataset for two deep learning Models hybrid VGG-16 and hybrid DenseNet201. Before applying the dataset on deep learning models, lung CT images are processed with image data augmentation methods for enhancement. After augmenting the dataset, this dataset is applied on deep learning models with two optimizers Adam and SGD. There are four combination of models with optimizes (VGG-16 with Adam, VGG-16 with SGD, DenseNet201 with Adam, and DenseNet201 with SGD). After calculating classification results, the conclusion is that DenseNet201 model with SGD optimizer provides better results than others.

The overall content of this work is ordered in some sections as follows:

- **Section-1:** Introduction section discusses the main contribution of different researches related to different segmentation and classification techniques of medical images for identifying the different diseases.
- **Section-2:** Methodology section provides the information about dataset and methods which are utilized for segmentation and classification of cancer in lung CT scan images. This section discusses about the various features methods like GLCM, GLRLM, GLSZM, and NGTDM; machine learning method Random Forest and deep learning models VGG16 and Denset201.
- **Section-3:** Result and Discussion section shows the experimental results and discussion that mostly focus on classification results with accuracy, recall, precision, and F1-score, using combination of four different models.
- **Section-4:** This section tells about the conclusion of this work and future scope.

2. METHODOLOGY

This part covers the data collection, preprocessing steps, segmentation technique, feature extraction methods, classification methods of lung CT scan images. The Fig. 1 demonstrations the block-diagram of overall architecture of experimental machine and deep learning methods utilized in this research work.

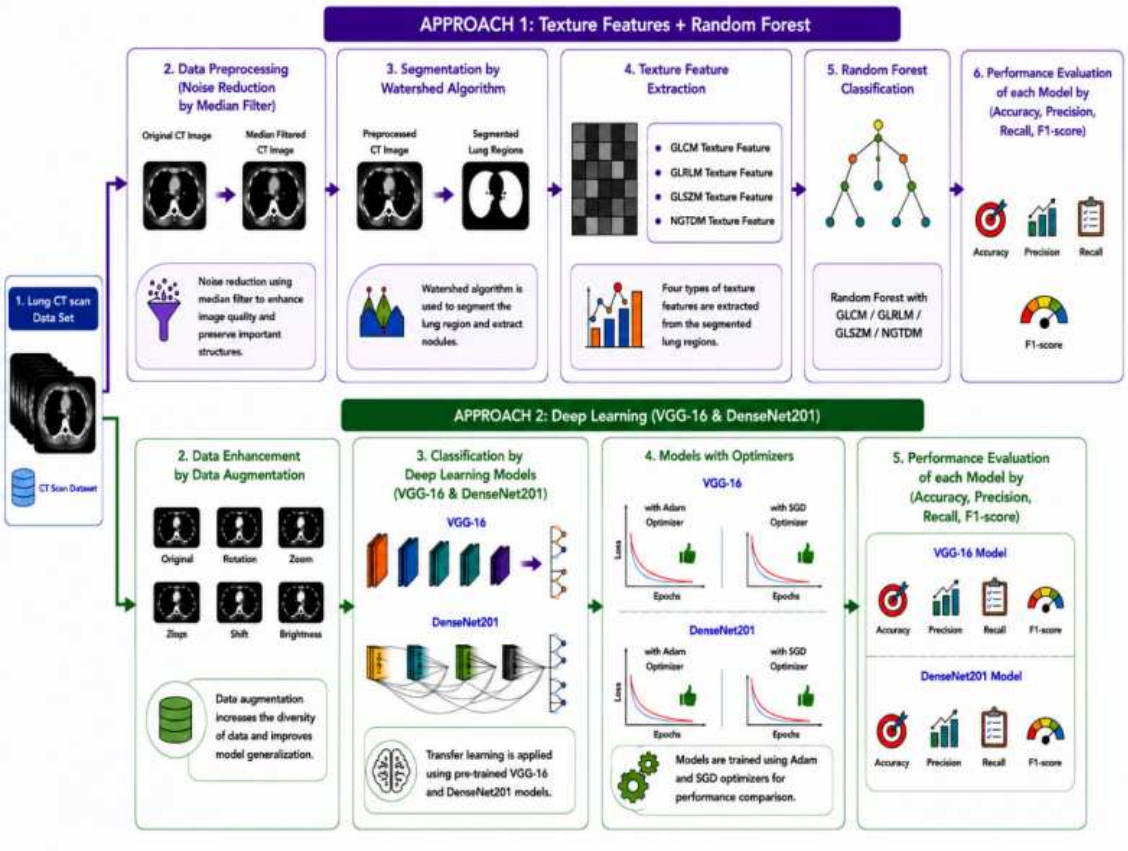


Figure 1. Overall structure of experimental machine learning model used in this work which includes the clear understanding of image preprocessing methods used, then feature extraction methods used and 08 different machine and deep learning models utilized in this work for classification of lung images.

2.1 Dataset

Data Science Bowl 2017 dataset from Kaggle [21] is utilized in this work. The lung images are two types, cancer and non-cancer. The images of dataset are in DICOM image format. DICOM stands for Digital Imaging and Communications in Medicine. DICOM format is an internationally granted standard for all medical imaging modalities. This dataset contains more than a thousand DICOM-formatted low-dose CT scans of high-risk individuals. A number of axial slices of the chest cavity are included in each image. The number of 2D slices in each image varies depending on the patient and the machine doing the scan. The header of the DICOM files includes the slice thickness and other scan characteristics, along with the required patient ID information.

2.2 Details of experimental setup

For the experimental work in this investigation, the following configuration was created:

Operating system: Windows 11 Home Single Language, version 23H2

RAM: 12 GB

System type: 64-bit operating system

Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz

Software used for programming: Python (Jupyter Notebook (anaconda3))

2.3 Segmentation and Classification of lung CT scan image

Before segmentation and classification of lung CT scan images, preprocessing steps like noise reduction by median filter are processed. After preprocessing, different methods are applied for mining the features. Finally Random Forest technique is utilized for classifying cancer disease in lung CT scan images.

2.3.1 Image Acquisition and Preprocessing

The initial part of our model is to acquire the images from the Data Science Bowl 2017 dataset. All images are in DICOM format. By applying image processing operations, the useful information are enhanced or extracted from them. The basic image processing tasks are removing noises, adjusting contrast and brightness. Image preprocessing is an important step which is applied before segmentation of images of lung. The core objective of

preprocessing is to grow the clarity of images, eliminate noise, and expand contrast and normalize intensity of images. Following steps are given which are used in our model.

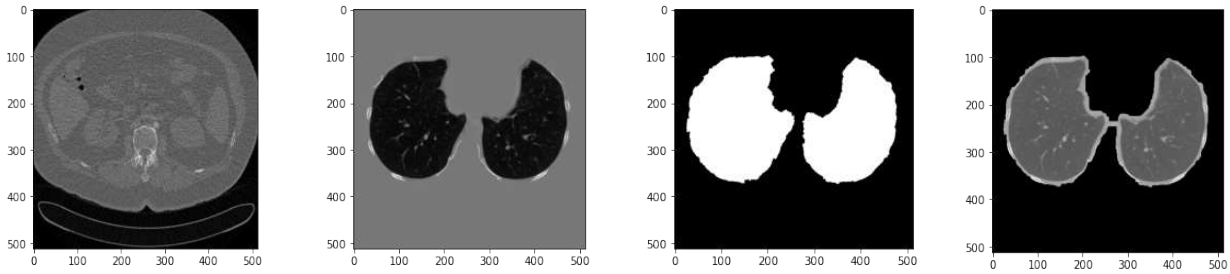
2.3.1.1 Noise Reduction: There are many kinds of noises which can affect lung images. Some examples of such noises are random pixel variations and imaging artifacts. For removing the noise we can apply noise reduction techniques like filtering. Some standard filters are median filter, Gaussian filter, and bilateral filter. These filters can eliminate noise from input images with preserving the structures and edges of images. In this approach, median filter is used to extract noise from lung images [22]. Median filter calculates the median of all the pixels within the kernel region and change the mid pixel value with median value.

After applying the filtering process, next step is segmenting the lung images. Segmentation of image is a way of splitting an image into multiple sections and these sections show the different objects of the image. The main purpose of the segmentation process is to make a simple representation of image; which is better expressive and easier to examine. To diagnose the lung cancer, image segmentation outlines the lung tumors or suspicious areas within lung tissue. By segmentation features of lung tissue's intensity, texture and shape are analyzed. The main goal of segmentation algorithms is to separate healthy lung tissues from the sections which designate the existence of a tumor. In this work, Watershed Image segmentation algorithm is applied for segmenting the lung images.

2.3.2 Watershed Image Segmentation:
Watershed Image segmentation technique is pertaining to the idea of treating the image as a topographic landscape with valleys and the ridge [23]. This technique can be used for 2D images as well as 3D images. The key idea of the watershed technique is to transform the image into a "gradient image" by which the edges in the image are heightened. After then the gradient image is flooded from the local minima and the highest level pixels are labeled as the object region and the leftover pixels are labeled as the background. The watershed algorithm is a standard image segmentation method utilized in computer vision and image processing. It is particularly effective for separating overlapping objects in an image, such as touching cells in biological images or adjacent regions in satellite imagery. The algorithm is named after the concept of a watershed in geography, where it represents the dividing line between two drainage basins. This algorithm works as follows [24]:

- **Gradient Image Calculation:**
  - The algorithm usually starts with a grayscale image, where the intensity values represent elevation (like a topographic map).
  - A gradient image is often computed from the primary image to highlight the edges of objects. In this context, edges correspond to the high-gradient areas, while the interior of objects has lower gradient values.
- **Marker Assignment:**
  - Markers are assigned to regions of the image that are known to belong to different objects or the background. Markers can be manually set or automatically determined based on certain criteria (like local minima).
  - These markers act as the "seeds" or starting points for the flooding process.
- **Flooding Process:**
  - Imagine the gradient image as a landscape, where the intensity represents the height. Water is "poured" simultaneously from each marker, and it flows downhill to fill up basins.
  - The algorithm simulates the process of water flooding the landscape, starting from the markers and gradually filling the basins. When water from different basins meets, a "dam" is built, which corresponds to the boundaries of the segmented regions.
- **Segmentation:**
  - The process continues until all the regions are flooded. The resulting dams represent the boundaries between different objects or regions in the image.
  - The output is a segmented image where each region corresponds to a different object or background.
  -

Fig. 2 shows the output of watershed algorithm.



(a) Original Lung Image (b) Lung Nodule (c) Watershed Image (d) Segment Lung

Figure 2. (a) is showing original lung image, (b) is showing the lung nodule after applying Watershed algorithm, (c) is showing watershed image, and (d) is the final segmented image

2.3.3 Feature Extraction

During the process of mining features using the following methods, dataset is transformed into a simplified form for further analysis. The objective of the extracting features is to decrease data dimensionality while maintaining the important information that is required to resolve a specific problem. The following feature extraction techniques are utilized in our model.

2.3.3.1 Extracting features from preprocessed lung CT scan images using GLCM

The GLCM is a popular second-order statistical texture analysis technique utilized in image processing to study the texture of an image. It is particularly effective for identifying patterns and structures in images by examining the spatial correlation between pixel pairs at various offsets and directions [25]. Firstly it was suggested by Haralick et al. [26] in 1973. Here's a breakdown of what GLCM is and how it works:

- **Gray Levels:** These are the different shades of gray that a pixel in a grayscale image can have, ranging from 0 to 255 in an 8-bit image. The GLCM is depended on the occurrence of pairs of these gray levels at a certain spatial relationship in the image.
- **Co-occurrence Matrix:** Basically the GLCM is a matrix where number of gray levels in the image is represented by number of rows and columns of matrix. Each element in the matrix represents the frequency with which a particular pair of pixel intensities (i, j) occurs in the image, given a specific spatial correlation denoted by a certain distance and angle.
- **Spatial Relationship:** The spatial relationship is explained by a specific distance (e.g., one pixel apart) and a direction (e.g., horizontal, vertical, diagonal). The direction can be specified by angles such as 0°, 45°, 90°, and 135°.
- **Matrix Construction:** For each pixel in the image, the algorithm examines the pixel at a certain distance and direction. If the pair of pixels has the gray level values (i, j), the corresponding matrix element GLCM (i, j) is incremented.
- **Normalization:** Often, the GLCM normalization is conducted by dividing each element by the total number of co-occurrences in the matrix, making it a probability distribution.
- **Texture Features:** From the GLCM, several statistical measures can be derived to describe the texture of the image [27], including:
  - Contrast: Computes the local variations in the gray-level values.

$$contrast = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(i-j)^2.P(i,j) \quad (1)$$

  - Dissimilarity: Computes the average difference in intensity between neighboring pixels. Its high values indicate higher heterogeneity in texture.

$$dissimilarity = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|i-j|.P(i,j) \quad (2)$$

  - Homogeneity: Computes the closeness of the distribution of elements in the GLCM to the diagonal.

$$homogeneity = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\frac{P(i,j)}{1+|i-j|} \quad (3)$$

  - Energy: Also known as uniformity, measures the addition of squared elements in the GLCM.

$$energy = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}P(i,j)^2 \quad (4)$$

  - Correlation: Computes how correlated a pixel is to its neighbor over the entire image.

$$correlation = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j} \quad (5)$$

Table 1. A sample of GLCM based texture feature set of a single lung image

Distance b/w pixel pairs	Angle	Contrast	Dissimilarity	Homogeneity	Energy	Correlation
1	0	232.3005	2.5010	0.9322	0.9138	0.7915
	π/4	307.4175	3.3235	0.9268	0.9122	0.7246
	π/2	263.4986	2.7853	0.9301	0.9125	0.7635
	π/4	298.7585	3.1711	0.9236	0.9121	0.7323
2	0	331.3655	3.6268	0.9192	0.9129	0.7031
	π/4	307.4175	3.3235	0.9226	0.9122	0.7246
	π/2	387.3451	4.0949	0.9156	0.9103	0.6530
	π/4	298.7585	3.1711	0.9236	0.9121	0.7323
3	0	386.5970	4.2008	0.9141	0.9119	0.6543
	π/4	419.7722	4.4712	0.9125	0.9096	0.6253
	π/2	446.6577	4.7269	0.9101	0.9080	0.6006
	π/4	397.0958	4.2265	0.9130	0.9095	0.6455

*Author for Correspondence: shashankyadav48@gmail.com

2.3.3.2 Extracting features from preprocessed lung CT scan images using GLRLM

GLRLM is considered as a statistical method used in image processing and analysis, particularly for **texture analysis** in 2D or 3D images. GLRLM describes the texture of an image by counting consecutive occurrences (runs) of a specific gray level along a specific direction in the image [28].

- **Gray Levels**: The intensity values in the image, usually normalized or quantized into a certain range (e.g., 0-255 in an 8-bit grayscale image).
- **Run Length**: The quantity of consecutive pixels that have the similar gray level in a specific direction (e.g., horizontally, vertically, and diagonally).
- **Matrix Formation**: For each direction (e.g., 0°, 45°, 90°, 135°), a matrix is created where:
  - Rows represent gray levels.
  - Columns represent the length of the run.
  - Each entry in the matrix counts how many runs of a particular gray level and run length are present in the image.

Once the GLRLM is generated, various features can be extracted from it to characterize the texture, such as [27]: Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray Level Non-uniformity (GLN), Run Length Non-uniformity (RLN), Run Percentage (RP), Low Gray-Level Run Emphasis (LGRE), High Gray-Level Run Emphasis (HGRE), Short Run Low Gray-Level Emphasis (SRLGLE), Short Run High Gray-Level Emphasis (SRHGLE), Long Run Low Gray-Level Emphasis (LRLGLE), and Long Run High Gray-Level Emphasis (LRHGLE). GLRLM is often utilized in medical imaging (e.g., radiomics) to analyze the texture of tissues in MRI, CT, and other imaging modalities. It helps to differentiate between normal and pathological tissues based on their texture properties.

$$SRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j)}{j^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (6)$$

$$LRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}j^2\,P(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (7)$$

$$GLN = \frac{\sum_{i=1}^{N_g}(\sum_{j=1}^{N_r}P(i,j))^2}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (8)$$

$$RLN = \frac{\sum_{j=1}^{N_r}(\sum_{i=1}^{N_g}P(i,j))^2}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (9)$$

$$RP = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)}{N_p} \quad (10)$$

$$LGRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j)}{i^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (11)$$

$$HGRE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}i^2\,P(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (12)$$

$$SRLGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{P(i,j)}{i^2j^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (13)$$

$$SRHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}(i^2/j^2)P(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (14)$$

$$LRLGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}(j^2/i^2)P(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (15)$$

$$LRHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}i^2j^2P(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}P(i,j)} \quad (16)$$

Where N_g is the number of gray levels, N_r is the maximum run length, $P(i, j)$ represents an entry in the matrix that describes the frequency of runs with a specific gray level i and run length j, and N_p is the total number of pixels in the image.

Table 2. A sample of GLRLM based texture feature set of two lung images

	SRE	LRE	GLN	RLN	RP	LGRE	HGRE	SRLGLE	SRHGLE	LRLGLE	LRHGLE
Lung Image 1	0.8311	4171.10	212.00	13377.64	0.15	6144367556045.27	14845.08	6144367556045.27	13018.81	1.0	9223343.73

*Author for Correspondence: shashankyadav48@gmail.com

Lung Image 2	0.7525	177228.39	181.81	3071.18	0.04	432341899512229.0	16249.15	432841899512229.0	14314.11	1.0	38078053.94

2.3.3.3 Extracting features from preprocessed lung CT scan images using GLSZM

GLSZM is considered as a texture examination method used in image processing, especially for analyzing the texture of grayscale images. Similar to the GLRLM, GLSZM captures spatial relationships between pixel intensities, but instead of focusing on consecutive runs of the same gray level, it looks at 'zones of connected pixels' with the same intensity [15]. The working of this method as follows.

- **Gray Levels**: Like in GLRLM, the image is quantized into a set number of gray levels (for example, 8, 16, or 256 gray levels).
- **Zone**: A zone is a group of connected pixels that all use the same gray level. A zone can be of any shape, not limited to a specific direction like in GLRLM. The size of the zone states to the number of pixels that use the same gray level and are spatially connected (either 4-connected or 8-connected).
- **Matrix Construction**:
  - **Rows** of the GLSZM represent gray levels.
  - **Columns** represent the zone sizes (number of connected pixels of the same gray level).
  - Each element in the matrix counts how many zones of a particular size and gray level exist in the image.

Once the GLSZM is generated, various features can be extracted from it to characterize the texture, such as: Small Zone Emphasis (SZE), Large Zone Emphasis (LZE), Gray Level Non-Uniformity (GLN), Zone Size Non-Uniformity (ZSNU), Zone Percentage (ZP), Low Gray-Level Zone Emphasis (LGZE), High Gray-Level Zone Emphasis (HGZE), Small Zone Low Gray-Level Emphasis (SZLGLE), Small Zone High Gray-Level Emphasis (SZHGLE), Large Zone Low Gray-Level Emphasis (LZLGLE), Large Zone High Gray-Level Emphasis (LZHGLE). Unlike GLRLM, GLSZM does not rely on specific directions (like horizontal, vertical, or diagonal) but considers the entire 2D image and captures the size of the gray-level zones. GLSZM is valuable because it provides a rotation-invariant analysis of image textures, meaning that it doesn't depend on specific directions, making it robust in a variety of applications.

$$SZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{S(i,j)}{j^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (17)$$

$$LZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}j^2\,S(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (18)$$

$$GLN = \frac{\sum_{i=1}^{N_g}(\sum_{j=1}^{N_s}S(i,j))^2}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (19)$$

$$ZSNU = \frac{\sum_{j=1}^{N_s}(\sum_{i=1}^{N_g}S(i,j))^2}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (20)$$

$$ZP = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)}{N_p} \quad (21)$$

$$LGZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{S(i,j)}{i^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (22)$$

$$HGZE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}i^2\,S(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (23)$$

$$SZLGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{S(i,j)}{i^2j^2}}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (24)$$

$$SZHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}(i^2/j^2)S(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (25)$$

$$LZLGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}(j^2/i^2)S(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (26)$$

$$LZHGLE = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}i^2j^2S(i,j)}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}S(i,j)} \quad (27)$$

Where N_g is the number of gray levels, N_s is the number of zone sizes, $S(i, j)$ describes frequency of zones in an image with a specific gray level i and zone size j, and N_p is the total number of pixels in the image.

Table 3. A sample of GLSZM based texture feature set of two lung images

	SZE	LZE	GLN	ZSNU	ZP	LGZE	HGZE	SZLGLE	SZHGLE	LZLGLE	LZHGLE

Lung Image 1	0.1981	-6246 31.27	0.0068	0.5155	3.3620	0.0020	16147.37	0.0003	3187.91	-1671.18	-489586.58
Lung Image 2	0.1886	-3230 97.16	0.0043	0.4651	0.2350	0.0077	19623.23	0.0009	3740.62	-10079.91	-2192248.14

2.3.3.4 Extracting features from preprocessed lung CT scan images using NGTDM

NGTDM is considered as a texture examination technique utilized in image processing to evaluate the gap in gray levels between a pixel and its nearby pixels. This method helps in analyzing the spatial relationships and texture patterns within an image [16]. The working of this method is as follows.

- **Gray Levels**: The image is measured into a set number of gray levels (e.g., 256 levels for an 8-bit grayscale image).
- **Neighborhood**: For each image pixel, a neighborhood is stated around it. This neighborhood is usually a square region (e.g., 3x3, 5x5 pixels) centered on the pixel of interest.
- **Difference Calculation**: The NGTDM computes the dissimilarity in gray levels between the central pixel and each of its neighboring pixels inside the defined neighborhood.
- **Matrix Construction**:
  - **Rows** show the gray levels of the central pixel.
  - **Columns** show the gray tone differences calculated from the neighborhood.
  - Each entry in the matrix counts how many times a particular gray level difference occurs for a specific gray level of the central pixel.

Once the NGTDM is generated, various features can be extracted from it to characterize the texture, such as: Coarseness, Busyness, Contrast, Strength and Complexity [29]. NGTDM is particularly valuable because it captures the local texture variations around each pixel, providing insight into the fine details and patterns within an image.

- **Coarseness:** computes the uniformity of intensity differences in the image.

$$Coarseness = \frac{1}{\sum_{i=1}^{Ng} S(i)} \quad (28)$$

- **Contrast:** computes the intensity variation across different gray levels.

$$Constrast = \frac{\sum_{i=1}^{Ng} n(i)\, S(i)\, (i-\bar{g})^2}{N} \quad (29)$$

Where $\bar{g}$ is the mean gray level of the image.

- **Busyness:** denotes the level of rapid intensity changes in the image.

$$Busyness = \frac{\sum_{i=1}^{Ng} S(i)}{\sum_{i=1}^{Ng} n(i)\, |i-\bar{g}|} \quad (30)$$

- **Complexity:** reflects the amount of detail in the image.

$$Complexity = \frac{\sum_{i=1}^{Ng} n(i)\, S(i)}{\sum_{i=1}^{Ng} S(i)} \quad (31)$$

- **Strength:** denotes the distinction between high and low gray levels in the image.

$$Strength = \frac{\sum_{i=1}^{Ng} n(i)\, (i-\bar{g})^2}{\sum_{i=1}^{Ng} S(i)} \quad (32)$$

Table 4. A sample of NGTDM based texture feature set of two lung images

	Coarseness	Contrast	Busyness	Complexity	Strength
Lung Image 1	2.7275	8.1356	0.000032	1174491.79	133847.09
Lung Image 2	2.8524	8.1140	0.000029	1166756.92	133151.87

2.3.4 Classification of lung CT scan images using texture based features and Random Forest algorithm

After extracting the features Random Forest is applied on all features matrices as shown in Fig. 3. It is a flexible and robust machine learning method designed for both classification and regression tasks. It constructs multiple decision trees and combines their outputs to achieve more accurate and reliable predictions. Here's a breakdown of how Random Forest works and its key characteristics:

- **Bootstrap Aggregating (Bagging)**: Random Forest uses a technique called bagging to build multiple decision trees [30]. In bagging, different subsets of the training data are utilized to train each tree. These subsets are created by sampling the training data with

replacement, meaning some instances may be repeated in the subsets.

- **Building Trees**: Every node in the decision tree is connected to two child nodes and one parent node, making it a directed graph. Parent to child is the direction of the connection. With the binary inquiry φ_n (a feature test), the sample x begins its journey at the root node $n = 0$. The response determines whether it is forwarded to the root node's left child ($\varphi_n(f) = 1$) or right child ($\varphi_n(f) = 0$) [31]. At the new node, the procedure is repeated, and x continues down the tree until it hits a leaf node. Each decision tree is generated by a random subset of features at each split in the tree. This process helps in making the trees less correlated with each other, which improves the overall performance and reduces over-fitting.
- **Voting/Averaging**: For classification tasks, the Random Forest algorithm takes a vote from each decision tree to decide the final class label. The final prediction is made for the class that receives the most votes. For regression jobs, the Random Forest algorithm calculates median of the predictions from all decision trees to generate the finishing output.

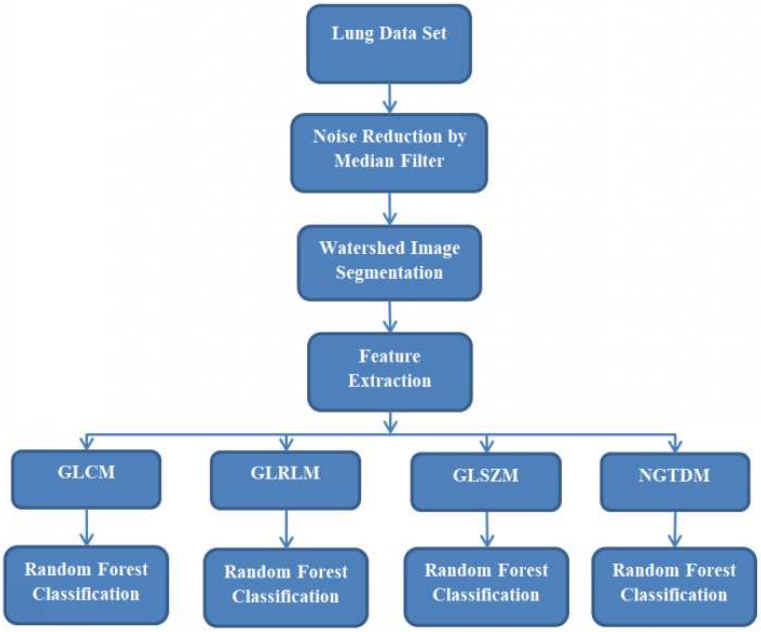


Figure 3. Lung Images Segmentation and Classification by Random Forest with different features GLCM, GLRLM, GLSZM, and NGTDM

2.3.5 Dataset Enhancement and VGG16 and DenseNet Classification

In this work two deep learning models are also processed with same dataset. Before applying deep learning models, dataset is enhanced by data augmentation method.

2.3.5.1 Lung Image Enhancement by Data Augmentation

The job of artificially expanding a training dataset's size and diversity is recognized as data augmentation. In machine learning, it is frequently employed, particularly for deep learning models, which depend on big and varied datasets to function well. Additionally, over-fitting is a prevalent issue in machine learning that can be lessened with the aid of data augmentation [32]. If a model learns the training data too well and is unable to generalize to new data, it is said to be over-fit. Data augmentation helps keep the model from over-fitting by growing the diversity of the training data. A model's capacity to generalize can be enhanced via data augmentation, which increases the model's resilience to various data variances. On every training and test sets, data augmentation frequently results in increased accuracy. For situations where gathering additional data is costly or time-consuming, data augmentation can be an affordable method of expanding the size and a training dataset's diversity. For enhancing the lung images we augment all images by using ImageDataGenerator python function. We augmented images by including rotation range, zoom range, shear range, horizontal flip, height shift range, and width shift range [33].

2.3.5.2 Application of VGG16 Architecture for classification using Adam and SGD optimizers

The **VGG16** architecture is considered as a deep convolutional neural network (CNN) model utilized for classifying the images. This model was established by the Visual Geometry Group (VGG) at the Oxford University and was introduced in their 2014 paper for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [34]. This model is famous for its easiness and effectiveness, and it has been broadly utlized as a benchmark in computer vision tasks.

2.3.5.2.1 Key Characteristics of VGG16:

- **Depth**: VGG16 is considered as a 16 weighted layers deep network. The names of layers are convolutional layers, fully connected layers, and pooling layers.
- **Architecture**: 13 convolutional layers are used in this model [35]. 3x3 kernels with a stride of 1 and padding are used in these layers to keep the spatial dimensions unchanged. This design choice allows the network to maintain high resolution through the layers. There are some max-pooling layers to decrease the spatial dimensions of the feature maps with holding significant features. Pooling layers used a 2x2 window with a stride of 2. The network ends with 3 fully connected layers. 4096 neurons are presented in the first two fully connected layers each, and the final fully connected layer has as many neurons as there are classes (e.g., 1000 for ImageNet).
- **Layer Structure:** VGG16 layer structure is shown in Fig. 4.

Input Layer: It is fed by an input image of size 224×224×3 (height × width × channels).

Convolutional Layers with pooling layers: Organized as follows [36]:

- First two convolutional layers contain 64 filters and every layer is followed by max-pooling layer.
- Next 2 convolutional layers contain 128 filters followed by max-pooling layer.
- Next 3 convolutional layers contain 256 filters followed by max-pooling layer.
- Next 3 convolutional layers with 512 filters followed by max-pooling layer.
- Next 3 convolutional layers with 512 filters followed by max-pooling layer.

Fully Connected Layers: las 3 convolutional layers followed by max-pooling layer are followed by three dense layers. The first two dense layers contais 4096 neurons, and the number of neurons in the final layer is equal to the number of classes in the classification task.

- **Activation Function**: we utilized rectified Linear Unit (ReLU) activation function for all convolutional and fully connected layers. But for the output layer we used a softmax function for classification.

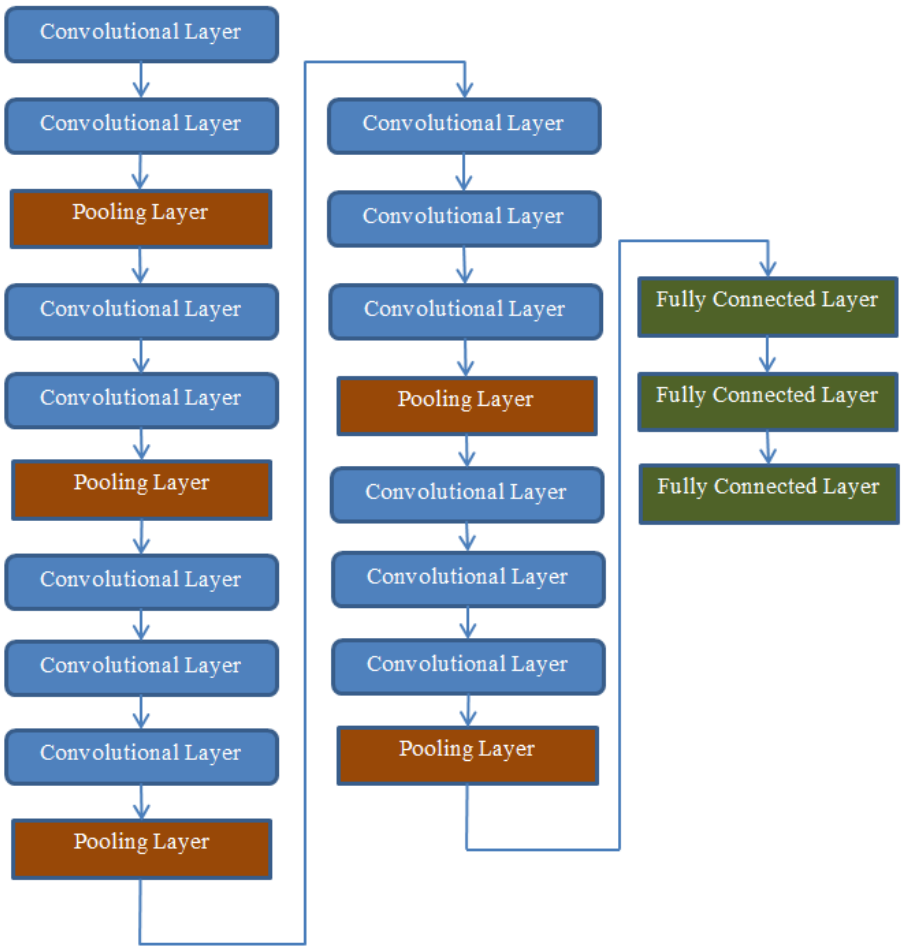


Figure 4. VGG16 Design with 13 convolutional layers, 3 fully connected layers, and 5 max-pooling layers

2.3.5.2.2 Adam Optimizer:

- **Overview**: Adam (Adaptive Moment Estimation) is a standard optimization optimizer utilized to update network weights based on the training data.
- **Functionality**: It merges the advantages of two popular optimizers, **AdaGrad** (which alters the learning rate for every parameter) and **RMSProp** (which applies moving averages of gradients to scale the learning rate).
- **Formula**: Initialize 1st moment vector $mv1_t$, 2nd moment vector $mv2_t$, and time stamp t by zero. At each iteration t=t+1, update biased moments:

$$mv1_t = \beta_1 mv1_{t-1} + (1 - \beta_1) g_t \quad (33)$$

$$\mathrm{mv2_t} = \beta_2 mv2_{t-1} + (1 - \beta_2) g_t^2 \quad (34)$$

Now correct bias by following formulas.

$$\hat{\mathrm{m}}_\mathrm{t} = \frac{\mathrm{m_t}}{1-\beta_1^\mathrm{t}} \quad (35)$$

$$\hat{\mathrm{v}}_\mathrm{t} = \frac{\mathrm{v_t}}{1-\beta_2^\mathrm{t}} \quad (36)$$

After then, update the parameters by the following formula.

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (37)$$

Here:

$\beta 1$: Decay rate for the 1st moment (default value: 0.9).

$\beta 2$: Decay rate for the 2nd moment (default value: 0.999).

η: Learning rate (default: 0.001).

ϵ: Small constant to prevent division by zero (default value: 10^{-8}).

- **Key Features**:
  - **Momentum**: Maintains track of an exponentially decaying average of past gradients (first moment, mtm_tmt) to accelerate learning.
  - **Adaptive Learning Rates**: Maintains a moving average of the squared gradient (second moment, vtv_tvt) for each parameter.
  - **Hyperparameters**: Typical values for β1=0.9, β2=0.999 and $\epsilon=10^{-8}$
- **Advantages**:
  - Adam converges faster and is more efficient with sparse data and noisy gradients.
  - It requires less tuning compared to other optimizers like stochastic gradient descent (SGD).

2.3.5.2.3 Combining VGG16 and Adam:
When using VGG16 with the Adam optimizer, Adam helps improve the convergence speed and model accuracy by dynamically balancing learning rates for each weight. This combination is commonly used in tasks like transfer learning [37] where VGG16 is pre-trained on large datasets (i.e. ImageNet), and the Adam optimizer fine-tunes the model on a new dataset.

2.3.5.2.4 Stochastic Gradient Descent (SGD):

- **Overview**: SGD is one of the mostly utilized optimization algorithms for training deep learning models [38]. It improves the model's weights depended on the gradient of the loss function with respect to the model parameters.
- **Gradient Descent Variants**:
  - **Batch Gradient Descent**: Utilizes the whole dataset to calculate the gradient and update weights, but can be slow and resource-heavy for large datasets.
  - **Stochastic Gradient Descent**: Uses one sample at a time to update weights, making it faster and more efficient, but more noisy and with higher variance.
  - **Mini-batch Gradient Descent**: A compromise where a small batch of data points is used to compute gradients and update weights.
- **SGD Update Rule**:

$$\theta = \theta - \eta \nabla_\theta J(\theta) \quad (38)$$

Where:

- θ represents the model parameters (weights).
- η represents the learning rate.
- $\nabla_\theta J(\theta)$ represents the gradient of the loss function J(θ) with respect to the parameters.

2.3.5.2.5 Combining VGG16 and SGD:

Training VGG16 with SGD is standard practice. SGD, often with mini-batches, is used to optimize the weights of VGG16 during training. It helps to decrease the loss function by adjusting the network's parameters.

2.3.5.3 Application of DenseNet201 Architecture for classification using Adam and SGD optimizers
DenseNet201 is a deep CNN architecture, a variant of the DenseNet family, known for its densely connected convolutional layers. One research paper was presented in the 2017 named "Densely Connected Convolutional Networks" published by Gao Huang et al. [39] aim to address the matter of vanishing gradients and decrease the number of parameters while retaining high accuracy. The diagram of DenseNet201 is shown in Fig. 5.

*Author for Correspondence: shashankyadav48@gmail.com

2.3.5.3.1 Key Features of DenseNet201:

- **Densely Connected Layers**: In simple convolutional neural networks, every layer connects simply to the next layer; but in DenseNets, every layer is associated to each other layer in a feed-forward fashion. This signifies the outcome of every layer is concatenated with all subsequent layers, ensuring that gradients flow directly and averting the vanishing gradient problem.
- **201 Layers**: The "201" denotes to the whole number of layers in this specific model [40], which includes convolutional, pooling, batch normalization, and fully connected layers.
- **Reduced Parameters**: By leveraging feature reuse, DenseNets can use less number of parameters without sacrificing performance if we compare it to other deep networks like ResNet.
- **Growth Rate**: DenseNet uses a parameter called the growth rate, which defines how many new feature maps each layer contributes to the network.
- **Bottleneck Layers**: DenseNet201 employs bottleneck layers to improve computational efficiency. These layers use 1x1 convolutions to decrease the number of input feature maps, making the model less computationally expensive.
- **Applications**: DenseNet201 is commonly used in image classification tasks, and can also be adapted for object detection, segmentation, and other computer vision applications. It's often used with datasets like ImageNet.

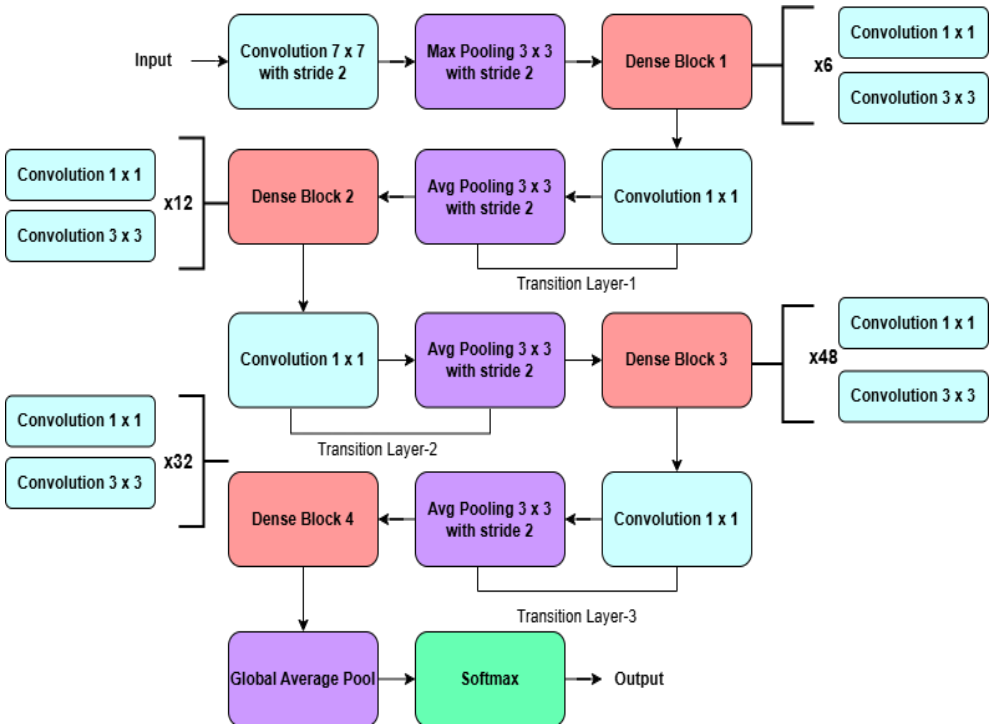


Figure 5: DenseNet201 Architecture

2.3.5.3.2 Adam with DenseNet201:

- **Adaptive learning rates**: Adam adjusts the learning rates depends on the gradient information, which can improve convergence for deep networks like DenseNet201.
- **Faster convergence**: Adam often converges faster compared to optimizers like SGD, especially with deeper architectures.
- **Works well with large datasets**: Adam, combined with DenseNet201's deep architecture, can handle large-scale datasets efficiently.

2.3.5.3.3 SGD with DenseNet201:

Densenet201 is a type of convolutional neural network (CNN) architecture in which 201 deep layers are there, part of the DenseNet family, known for its dense connections between layers. Utilizing **Stochastic Gradient Descent (SGD)** optimizer for DenseNet201 can be a good choice for improving convergence during training.

All combinations of deep learning models with optimizers are presented in Fig. 6.

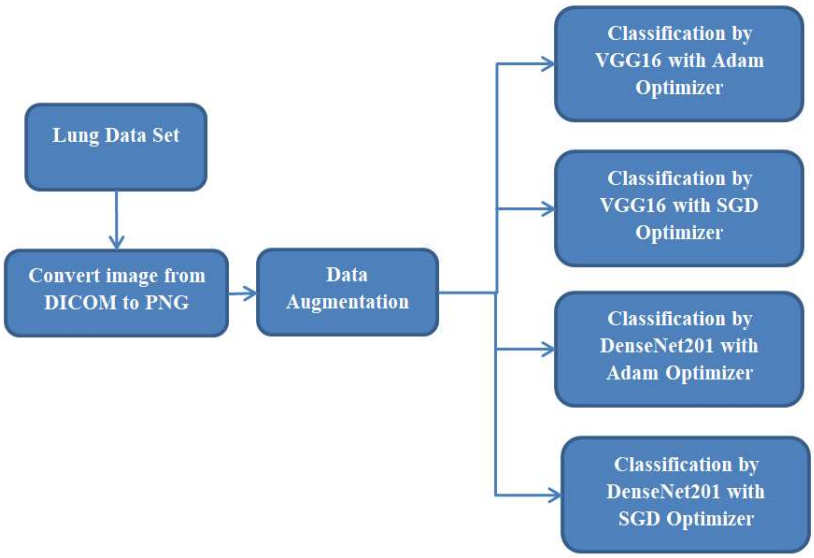


Figure 6. Lung Image Enhancement by Data Augmentation and classification by VGG16 and Densenet201 with Adam and SGD optimizers

2.4 Performance metrics to compute the quality of our suggested models

To assess the model's quality, performance metrics are used. These performance measures help us better understand how skillfully our model performs for the specified data. In this way, the model's performance can be upgraded by altering the hyper-parameters. These performance measures evaluate the model's capacity to accomplish good generalization on new or unknown data, which is the goal of all machine learning models.

2.4.1 Confusion Matrix (CM)

An M×M matrix (M is the number of anticipated classes) can be used to define a confusion matrix. It is a statistic used to evaluate how effectively classification issues in machine learning perform when the output consists of two or more classes. As explained below by Table 5, a confusion matrix is a table that contains four predicted and actual values, TP, FP, FN, and TN. It is very helpful for computing precision, accuracy, recall, and F1-score.

Table 5. Metrics used to measure the quality of the model obtained from confusion matrix of our models

	Target Output			
CM	**0**	**1**		
0	True Positive (TP)	False Positive (FP)	**Precision**	TP/ (TP+FP)
1	False Negative (FN)	True Negative (TN)		
	Recall=TP/(TP+FN)		**Accuracy** = (TP+TN)/(TP+FP+FN+TN)	
			F1-score = 2×Prec×Rec/(Prec + Rec)	

The following metrics were utilized to assess the experimented models in this work:

- **Recall:** the percentage of real positive cases that our model accurately detects. Recall is a useful indicator when False Positive is more worrisome than False Negative.
- **Precision:** the percentage of samples that tested positive and that our model accurately detected. Precision is useful when False Positives are more worrying than False Negatives.
- **F1 score:** It is a specific type of single score that combines precision and recall. Thus, by taking the harmonic mean of recall and precision and assigning equal weight to each, the F1 Score may be calculated.
- **Accuracy**: the percentage of our model's total correct predictions those were accurate.

3. RESULTS:

The cancer lung CT-scan dataset utilized in this study is discussed in this section, along with the experimental outcomes from applying several models to the dataset in order to classify and identify cancer.

3.1 Dataset used:

Data Science Bowl 2017 dataset from Kaggle is applied in the different deep and machine learning models in our work. Since the size of this dataset is very large, therefore we did not work on whole dataset. For Random Forest classification, 60 patients (30 from cancer data and 30 from non-cancer data) are selected and each patient has 40 images; therefore total 2400 images are used. For deep learning models total 3914 lungs images are chosen from the dataset.

3.2 Classification of Lung images by Random Forest using different texture features:

Before applying Random Forest, we applied preprocessing technique like median filter on the lung images and then applied Watershed segmentation technique for segmenting the images and then extract features. Table 5 is showing results of random forest classifier with GLCM features. Table 6 is showing results of random forest classifier with GLRLM features. Table 7 is showing results of random forest classifier with GLSZM features and table 8 is showing results of random forest classifier with NGTDM features.

For showing results we calculated recall, accuracy, F1-score, and precision. The data is distributed into training and testing data in many proportions (90:10, 85:15, 80:20, 75:25, 70:30, and 65:35) and then apply random forest for each data ratio. The table 6 is showing the result of Random Forest with GLCM texture based method. These results are least from other features.

Table 6. Lung Cancer Classification results using Random Forest model and GLCM features

Training and Testing Ratio	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
90:10	70.99	74.59	71.65	73.09
85:15	70.23	73.74	70.21	71.93
80:20	69.41	72.31	70.28	71.28
75:25	69.44	70.78	71.71	71.24
70:30	66.71	69.03	66.76	67.88
65:35	68.23	70.20	67.86	69.01
Average	**69.16**	**71.77**	**69.74**	**70.73**

In table 7, we can see that the performance of GLRLM is better than GLCM features (accuracy, recall, precision and f1-score).

Table 7. Lung Cancer Classification results using GLRLM features and Random Forest model

Training and Testing Ratio	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
90:10	74.45	75.76	78.74	77.22
85:15	76.01	77.20	79.26	78.22
80:20	76.78	77.95	79.52	78.73
75:25	75.34	75.63	78.62	77.10
70:30	73.95	74.34	77.20	75.74
65:35	73.44	73.84	75.95	74.88
Average	**74.99**	**75.78**	**78.21**	**76.98**

In table 8, GLSZM parameters are better than GLRLM parameters.

Table 8. Lung Cancer Classification results using GLSZM features and Random Forest model

Training and Testing Ratio	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
90:10	81.38	81.82	85.04	83.40
85:15	84.10	83.42	88.30	85.79
80:20	82.21	82.75	84.74	83.73
75:25	80.20	79.87	83.55	81.67
70:30	78.87	78.24	82.97	80.53
65:35	78.78	77.61	83.33	80.37
Average	**80.92**	**80.61**	**84.65**	**82.58**

Table 9 is showing the result of classification by random forest with NGTDM features. These results are better than GLSZM feature set. Finally we can say that from GLCM, GLRLM, GLSZM, and NGTDM feature sets, NGTDM features results gave best result with Random Forest classification method.

Table 9. Lung Cancer Classification results using NGTDM features and Random Forest model

Training and Testing Ratio	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
90:10	85.28	81.63	94.49	87.59
85:15	85.54	83.82	90.96	87.24
80:20	84.59	83.46	89.16	86.21
75:25	84.72	83.33	88.82	85.99
70:30	83.64	82.10	88.19	85.03

65:35	84.49	83.60	87.38	85.45
Average	**84.71**	**82.99**	**89.83**	**86.25**

3.2 Classification of Lung images by hybrid VGG16 and hybrid DenseNet using Adam and SGD classifier: Before applying the VGG16 and DenseNet, dataset is enhanced by data augmentation. Data was divided into three parts (training (70%), test (20%), and validation data (10%)). After then we applied VGG16 with Adam optimizer, VGG16 with SGD optimizer, DenseNet201 with Adam optimizer, and DenseNet201 model SGD optimizer. In table 10, we can see that if we apply DenseNet201 model with SGD optimizer, then results are better than other combinations. When DenseNet201 is applied with SGD optimizer, then for training data the model achieved (Accuracy= **99.38**%, Precision= **98.88**%, Recall= **99.53**%, and F1-Score=**99.20**%), for testing data the model achieved (Accuracy= 94.25%, Precision= 87.87%, Recall= 97.10%, and F1-Score=92.25%), and for validation data the model achieved ((Accuracy= 89.03%, Precision= 77.78%, Recall= 92.97%, and F1-Score=84.70%) .

The comparison between training and validation results of VGG16 with Adam optimizer in respect of accuracy, loss, precision, and recall is shown in Fig. 7. The same comparison is shown for VGG16 with SGD in Fig. 8. The comparison between training and validation results of DenseNet201 with Adam optimizer in respect of accuracy, loss, precision, and recall is shown in Fig. 9. The same comparison is shown for DenseNet201 with SGD in Fig. 10. Table 11 shows the comparison of our model result with results of different authors' works.

Table 10. Comparison of hybrid VGG-16 and hybrid DenseNet201 models with Adam and SGD optimizers

Models with optimizer	Data Type	Accuracy	Precision	Recall	F1-Score
Hybrid VGG16-Adam	Training Data	95.33	95.20	97.25	96.21
	Test Data	88.11	85.42	97.06	90.87
	Validation Data	82.65	80.00	95.40	87.02
Hybrid VGG16-SGD	Training Data	91.02	92.79	92.46	92.62
	Test Data	86.19	84.88	94.13	89.27
	Validation Data	82.14	83.67	87.87	85.72
Hybrid DenseNet201-Adam	Training Data	91.06	79.05	97.58	87.34
	Test Data	81.59	86.56	71.93	78.57
	Validation Data	83.93	60.13	97.87	74.49
Hybrid DenseNet201-SGD	Training Data	99.38	98.88	99.53	99.20
	Test Data	94.25	87.87	97.10	92.25
	Validation Data	89.03	77.78	92.97	84.70

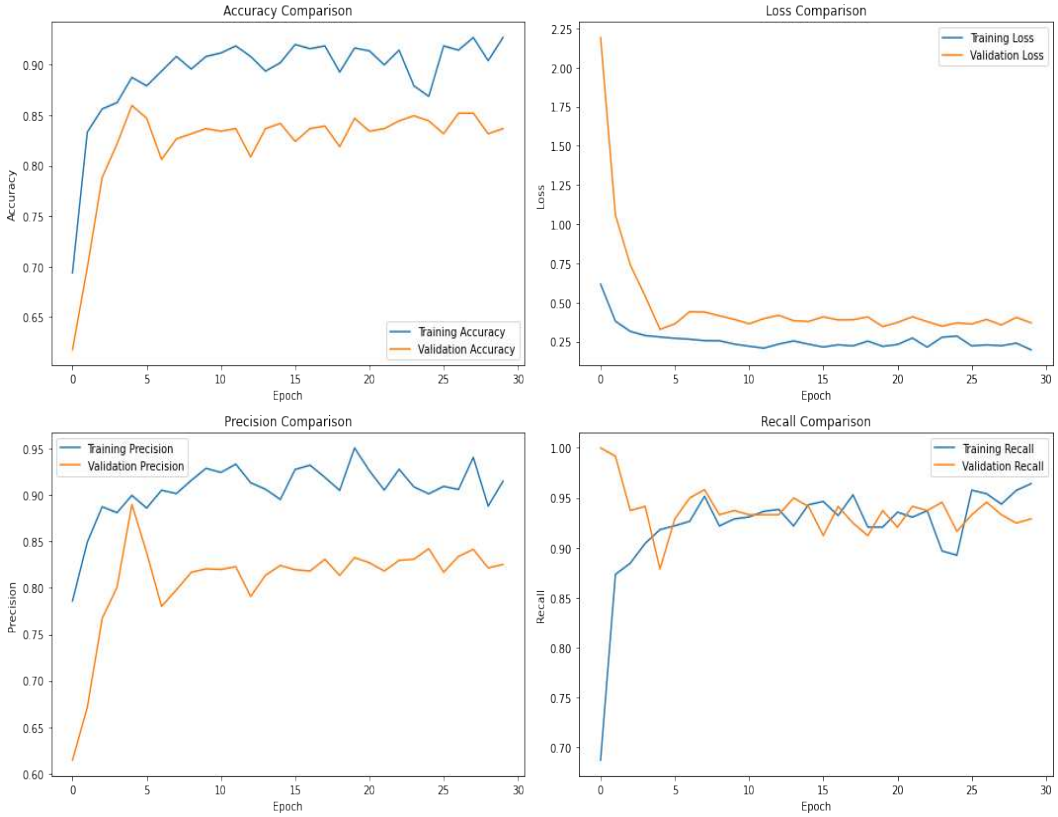


Figure 7. Comparison diagram of training and validation accuracy, loss, precision, and recall of VGG16 method with Adam optimizer

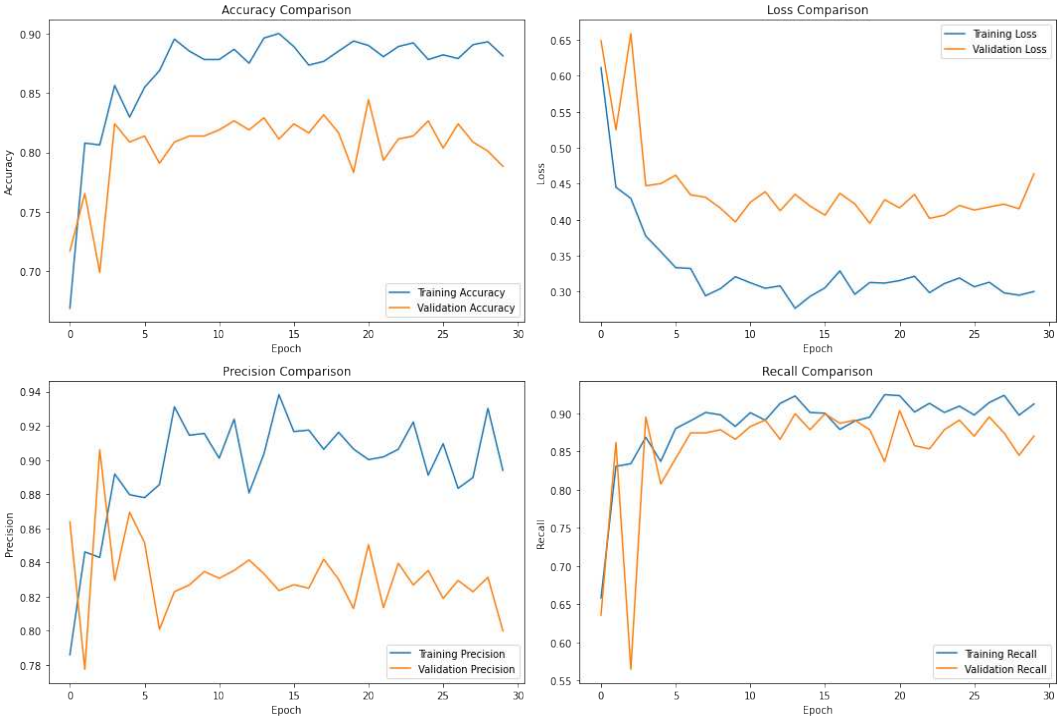


Figure 8. Comparison diagram of training and validation accuracy, loss, precision, and recall of VGG16 method with SGD optimizer

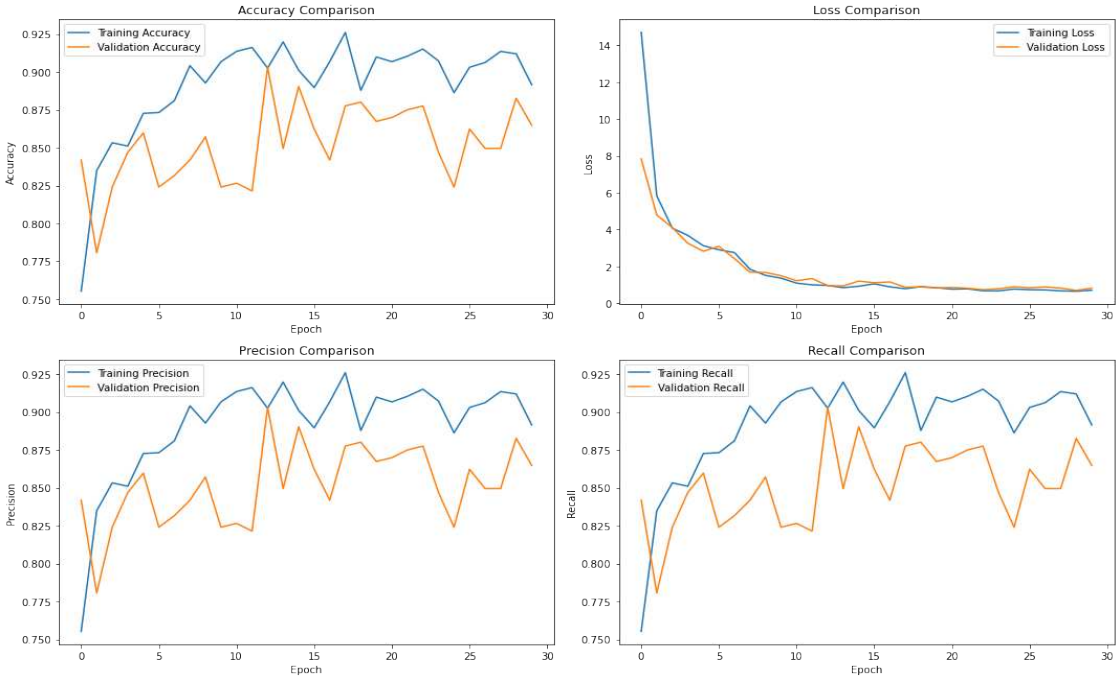


Figure 9. Comparison diagram of training and validation accuracy, loss, precision, and recall of DenseNet201 method with Adam optimizer

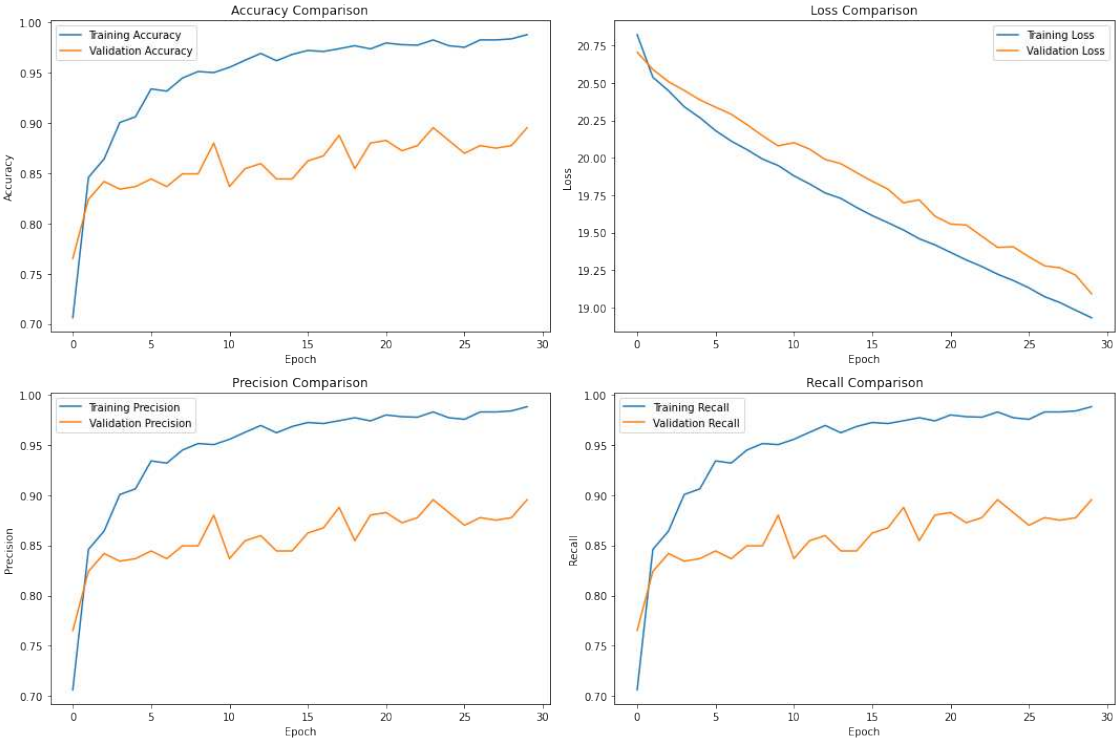


Figure 10. Comparison diagram of training and validation accuracy, loss, precision, and recall of DenseNet201 method with SGD optimizer

Table 11. Comparative results for Lung-Cancer detection among the literature based models and our proposed model

S. No.	Author	Method	Dataset with number of images	Accuracy
1	Said et al. (2023) [1]	Self-Supervised Network	Decathlon (96 sets of segmented 3D CT scans)	98.77%
2	Kumar et al. (2018) [4]	Guaranteed Convergence Particle Swarm	Lung CT-Diagnosis repository (20 sample	95.89%

		Optimization	images)	
3	Jiang et al. (2021) [17]	Improved Visual Geometry Group-13	X-rays images (5856 images)	89%
4	Makaju et al. (2018) [22]	Support Vector Machine	Lung Image Database Consortium (LIDC) (15 nodules)	86.6%
5	Kumaran et al. (2024) [41]	Unified framework of VGG16, ResNet50, Inception V3	IQ-OTH/ NCCD Lung dataset (1097 images)	98.18
6	Chutia et al. (2024) [42]	Model based on Modified DenseNet201	Chest X-rays images (9403 images)	95.34%
7	**Our proposed model**	**Hybrid DenseNet201 with SGD optimizer**	Data Science Bowl 2017 dataset (3914 images)	**99.38%**

4. CONCLUSION:

To identify lung cancer, eight distinct models were tested on the publicly available Data Science Bowl 2017 dataset. Random Forest machine learning technique was used with four types of texture feature methods (GLCM, GLRLM, GLSZM, and NGTDM). The combination of Random Forest with NGTDM features provided the best result among other texture features methods (Accuracy-84.71%, recall-82.99%, precision-89.83%, and F1score- 86.25%). After applying the augmentation to the dataset, deep learning models hybrid VGG-16 and hybrid DenseNet201 were applied with Adam and SGD optimizers (VGG16+Adam, VGG16+SGD, DenseNet201+Adam, and DenseNet201+SGD). The combination of DenseNet201 with the SGD optimizer gave better results using training data than other combinations (Accuracy-99.16%, recall-99.16%, precision-99.16%, and F1score- 99.16%). In the future, these models can be used for other available datasets for detecting cancer and for identifying other diseases like brain tumors, etc.

FUNDING STATEMENT

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that they have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Author Contributions

Shashank Yadav contributed to the conceptualization, methodology, software implementation, experimental analysis, and writing of the original manuscript. Upendra Kumar contributed to supervision, validation of results, and critical review and editing of the manuscript. All authors read and approved the final manuscript.

Data Availability

The datasets used in this study are publicly available from online repositories.

https://academictorrents.com/details/015f31a94c600256868be155358dc114157507fc

Ethics, Consent to Participate, and Consent to Publish: Not applicable.

REFERENCES:

1. Said Y, Alsheikhy AA, Shawly T, Lahza H. Medical Images Segmentation for Lung Cancer Diagnosis Based on Deep Learning Architectures. *Diagnostics*. 2023;13(3):546. DOI: https://doi.org/10.3390/diagnostics13030546
2. Riaz Z, Khan B, Abdullah S, Khan S, Islam MS. Lung Tumor Image Segmentation from Computer Tomography Images Using MobileNetV2 and Transfer Learning. *Bioengineering*. 2023;10(8):981. DOI: https://doi.org/10.3390/bioengineering10080981
3. Abdillah B, Bustamam A, Sarwinda D. Image processing based detection of lung cancer on CT scan images. *J Phys: Conf Ser*. 2017;893:012063. DOI: 10.1088/1742-6596/893/1/012063
4. Kumar KS, Venkatalakshmi K, Karthikeyan K. Lung Cancer Detection Using Image Segmentation by means of Various Evolutionary Algorithms. *Computational and Mathematical Methods in Medicine*. 2019;2019:1-16. DOI: https://doi.org/10.1155/2019/4909846
5. Joon, P, Bajaj SB, Jatain A. Segmentation and Detection of Lung Cancer Using Image Processing and Clustering Techniques. *Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 713. Springer, Singapore*. DOI: https://doi.org/10.1007/978-981-13-1708-8_2
6. Nadkarni NS and Borkar S. Detection of Lung Cancer in CT Images using Image Processing. *Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore* Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8
7. Uzelaltinbulat S, Ugur B. Lung tumor segmentation algorithm. *Procedia Computer Science*. 2017;120:140-7. DOI: https://doi.org/10.1016/j.procs.2017.11.221

8. Primakov SP, Ibrahim A, van Timmeren JE, et al. Automated detection and segmentation of non-small cell lung cancer computed tomography images. *Nat Commun*. 2022; 13:3423. DOI: https://doi.org/10.1038/s41467-022-30841-3
9. Shimazaki A, Ueda D, Choppin A, et al. Deep learning-based algorithm for lung cancer detection on chest radiographs using the segmentation method. *Sci Rep*. 2022; 12:727. DOI: https://doi.org/10.1038/s41598-021-04667-w
10. Miah MBA and Yousuf MA. Detection of Lung Cancer from CT Image Using Image Processing and Neural Network. *2nd Int'l Conf on Electrical Engineering and Information & Communication Technology (ICEEICT) 20 IS Jahangirnagar University, Dhaka-1342, Bangladesh*, 2015; 21-23.
11. Cifci MA. SegChaNet: A Novel Model for Lung Cancer Segmentation in CT Scans. *Applied Bionics and Biomechanics*. 2022;2022:1-16. DOI: https://doi.org/10.1155/2022/1139587
12. Patil JK, Dhole SA, Mandlik VS and Patil VD. Detection of Lung Cancer from CT Scan Images using GLCM Features. *Grenze International Journal of Engineering and Technology, Jan Issue*. 2023.
13. Vironicka AS and Sathiaseelan JGR. Texture Feature Extraction with Medical Image using GLCM And Machine Learning Techniques. *ICTACT Journal on Image and Video Processing*, MAY 2022. VOLUME: 12, ISSUE: 04. DOI: 10.21917/ijivp.2022.0388
14. Muthukumaran N, Archana, Majitha A M, Sophiya TG. Analysis of Brain Tumor using Novel Image Processing Approach. *Proceedings of the Third International Conference on Electronics and Sustainable Communication Systems (ICESC 2022)IEEE Xplore*. 2022; Part Number: CFP22V66-ART; ISBN: 978-1-6654-7971-4
15. Thibault G, Fertil B, Navarro C, Pereira S, Cau P, Levy N, Sequeira J, Mari JA. Texture Indexes and Gray Level Size Zone Matrix Application to Cell Nuclei Classification. *10th International Conference on Pattern Recognition and Information Processing*. 2009.
16. Chen S, Harmon S, Perk T, et al. Using neighborhood gray tone difference matrix texture features on dual time point PET/CT images to differentiate malignant from benign FDG-avid solitary pulmonary nodules. *Cancer Imaging*. 2019;19(1). DOI: https://doi.org/10.1186/s40644-019-0243-3
17. Jiang Z, Liu Y, Shao Z, Huang K. An Improved VGG16 Model for Pneumonia Image Classification. *Applied Sciences*. 2021;11(23):11185. DOI: https://doi.org/10.3390/app112311185
18. Liu Z, Peng J, Guo X, Chen S, Liu L. Breast cancer classification method based on improved VGG16 using mammography images. *Journal of Radiation Research and Applied Sciences*. 2024;17(2):100885. DOI: https://doi.org/10.1016/j.jrras.2024.100885
19. Taher O, Özacar K. MedCapsNet: A modified Densenet201 model integrated with capsule network for heel disease detection and classification. *Heliyon*. 2024;10(14):e34420. DOI: https://doi.org/10.1016/j.heliyon.2024.e34420
20. Benhassine NE, Boukaache A, Boudjehem D. Breast cancer image classification using DenseNet201 and AlexNet based deep transfer learning. *Proceedings of the International Conference on Emerging Intelligent Systems for Sustainable Development (ICEIS 2024)*. 2024. DOI: https://doi.org/10.2991/978-94-6463-496-9_11
21. https://academictorrents.com/details/015f31a94c600256868be155358dc114157507fc
22. Makaju S, Prasad P, Alsadoon A, Singh A, Elchouemi A. Lung Cancer Detection using CT Scan Images. *Procedia Computer Science*. 2018;125:107-14. DOI: 10.1016/j.procs.2017.12.016
23. Tripathi P, Tyagi S, Nath M. A Comparative Analysis of Segmentation Techniques for Lung Cancer Detection. *Pattern Recognit Image Anal*. 2019;29(1):167-73. DOI: 10.1134/S105466181901019X
24. Kornilov AS, Safonov IV. An Overview of Watershed Algorithm Implementations in Open Source Libraries. *J Imaging*. 2018;4(10):123. DOI: 10.3390/jimaging4100123
25. Kurniati FT, Sembiring I, Setiawan A, Setyawan I, Huizen RR. GLCM-Based Feature Combination for Extraction Model Optimization in Object Detection Using Machine Learning. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*. Vol. 9, No. 4, December 2023, pp. 1196-1205 ISSN: 2338-3070, DOI: 10.26555/jiteki.v9i4.27842
26. Haralick RM, Shanmugam K, Dinstein I. Textural Features for Image Classification. *IEEE Trans Syst, Man, Cybern*. 1973; SMC-3(6):610-21. https://doi.org/10.1109/TSMC.1973.4309314
27. Kumar U. A noval approach of classification of COVID-19 from chest CT-Scan images using ensemble classifier in combination with cognition based texture features. *Multimed Tools Appl*. 2024; https://doi.org/10.1007/s11042-024-18670-6
28. Zhang H, Hung C, Min G, Guo J, Liu M, Hu X. GPU-Accelerated GLRLM Algorithm for Feature Extraction of MRI. *Sci Rep*. 2019; 9:10883. DOI: https://doi.org/10.1038/s41598-019-46622-w

29. Mayerhoefer ME, Materka A, Langs G, et al. Introduction to Radiomics. *J Nucl Med*. 2020;61(4):488-95. DOI: 10.2967/jnumed.118.222893

30. Leobreiman. Random Forests. *Machine Learning*. 2001; 45, 5–32.

31. Konukoglu E and Glocker B. Random forests in medical image computing. *Handbook of Medical Image Computing and Computer Assisted Intervention, The Elsevier and MICCAI Society Book Series*. DOI: https://doi.org/10.1016/B978-0-12-816176-0.00024-7

32. Islam T, Hafiz MS, Jim JR, Kabir MM, Mridha M. A systematic review of deep learning data augmentation in medical imaging: Recent advances and future research directions. *Healthcare Analytics*. 2024;5:100340. DOI: https://doi.org/10.1016/j.health.2024.100340

33. Goceri E. Medical image data augmentation: techniques, comparisons and interpretations. *Artif Intell Rev*. 2023;56(11):12561-605. DOI: https://doi.org/10.1007/s10462-023-10453-z

34. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*. 2015.

35. Linkon AHM, Labib MM, Hasan T, Hossain M, Jannat M. Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study. *Informatics in Medicine Unlocked*. 2021;24:100582. DOI: https://doi.org/10.1016/j.imu.2021.100582

36. Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *IJSRP*. 2019;9(10):p9420. DOI: http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420

37. Javaid S, Rizvi S, Ubaid MT, Darboe A, Mayo SM. Interpretation of Expressions through Hand Signs Using Deep Learning Techniques. *IJIST*. 2022;4(2):596-611. DOI: 10.33411/IJIST/2022040225

38. Mehmood F, Ahmad S, Whangbo TK. An Efficient Optimization Technique for Training Deep Neural Networks. *Mathematics*. 2023;11(6):1360. DOI: https://doi.org/10.3390/math11061360

39. Huang G, Liu Z, Maaten LVD, Weinberger KQ. Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*. 2017; pp. 2261-2269, DOI: 10.1109/CVPR.2017.243

40. Adhinata FD, Rakhmadani DP, Wibowo M, Jayadi A. A Deep Learning Using DenseNet201 to Detect Masked or Non-masked Face. *JUITA: Jurnal Informatika*. 2021; e-ISSN: 2579-8901; Vol. 9, No. 1.

41. Kumaran S Y, Jeya JJ, R MT, Khan SB, Alzahrani S, Alojail M. Explainable lung cancer classification with ensemble transfer learning of VGG16, Resnet50 and InceptionV3 using grad-cam. *BMC Med Imaging*. 2024;24(1) DOI: https://doi.org/10.1186/s12880-024-01345-x

42. Chutia U, Tewari AS, Singh JP, Raj VK. Classification of Lung Diseases Using an Attention-Based Modified DenseNet Model. *j Imaging Inform med*. 2024;37(4):1625-41. DOI: https://doi.org/10.1007/s10278-024-01005-0