Archives of Computational Methods in Engineering (2024) 31:1023–1049
https://doi.org/10.1007/s11831-023-10006-1

REVIEW ARTICLE

An Analysis of Deep Transfer Learning-Based Approaches for Prediction and Prognosis of Multiple Respiratory Diseases Using Pulmonary Images

Apeksha Koul[1] · Rajesh K. Bawa[2] · Yogesh Kumar[3]

Received: 8 March 2023 / Accepted: 15 September 2023 / Published online: 31 October 2023

Abstract

Respiratory diseases can lead to lung failure, which happens when the lungs cannot give the body enough oxygen. These diseases can be diagnosed using medical data, lung pathology, functional testing of the lungs, etc. However, challenges in care quality persist, which results in inaccurate diagnoses and restricts patient satisfaction. Deep transfer learning models have emerged as a powerful tool for detecting and classifying respiratory diseases due to their ability for analyzing the large bulk of data from medical images, patient records, etc., to identify patterns and predict the likelihood of disease. Keeping this in view, the paper aims to design a system for the prediction and classification of multiple respiratory diseases. Various deep transfer learning models such as EfficientNetB6, EfficientNetV2B3, DenseNet201, Inception-v3, Xception, EfficientNetV2B1, ResNet50V2, EfficientNetV2S, InceptionResNet-v2, ResNet101V2, and a proposed hybrid model (EfficientNetB6 + ResNet101V2) have been used to analyze 19,488 pulmonary images such as CT scans as well as chest x-ray of lung cancer, pulmonary embolism, COVID, and pneumoconiosis along with the healthy lungs. The images are initially pre-processed with the help of contrast enhancement technique and are represented graphically via histogram equalization to understand the distribution of their pixel intensity. To obtain the region of interest and extract features, contour features as well as thresholding techniques are applied. Later, the models are trained and evaluated which depicts that the proposed hybrid model computes the best accuracy, precision, recall, and F1 score with values of 99.77%, 1.00, 0.99, and 1.00, respectively, and the lowest loss value of 0.001. Based on the research, the proposed hybrid deep transfer learning model will help doctors and experts make better predictions and improve the classification of respiratory diseases.

1 Introduction

Lungs are the most important organ in human body but people neither take good care of them nor give importance to their own respiratory and breathing related issues which later causes various infections and injuries [1]. According to the WHO, airway diseases are the foremost cause of death and disability worldwide hence it is essential to ensure a better diagnostic technique that can provide an accurate diagnosis as well as take appropriate actions to cure it. Numerous factors contribute to the rise of such diseases which includes direct or indirect tobacco smoke exposure, low birth weight, heavy exposure to malnutrition, air pollution, and virus exposure, like the influenza virus or the Coronavirus. In addition to this, there are also certain similarities in the symptoms of these diseases which can cause confusion and lead to misdiagnosis in treatment, therefore it is important to detect and diagnose multiple airway diseases accurately and timely [2].

✉ Apeksha Koul
apekshakoulo9@gmail.com

Rajesh K. Bawa
rajesh.k.bawa@gmail.com

Yogesh Kumar
yogesh.arora10744@gmail.com;
yogesh.kumar@sot.pdpu.ac.in

[1] Department of Computer Science and Engineering, Punjabi University, Patiala, India

[2] Department of Computer Science, Punjabi University, Patiala, India

[3] Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar, India

In the healthcare sector, where massive and complex amounts of data are generated, artificial intelligence (AI) has proven to be an invaluable and irreplaceable asset. AI techniques have done a fantastic job in the image recognition sector of healthcare, such as to evaluate and classify the images of lung cancer, diagnosis of fibrotic lung disease, to interpret the pulmonary function tests as well as to diagnose various restrictive and obstructive lung diseases. In a broader sense, the medical sector is getting more involved with artificial intelligence for assisting doctors to predict and diagnose numerous types of diseases, particularly in the previous years when there was pandemic due to COVID-19 virus and insufficient hospitals were there to provide adequate care to ill people [3]. According to the NHS-commissioned Topol Report, the advanced algorithms of mathematics, cloud computing, etc. have escalated to develop the methods based on artificial intelligence (AI) techniques to analyze, interpret, and forecast healthcare data [4].

One of the researchers created a cough detection-based application, as shown in Fig. 1, that uses sensors to record the symptoms such as cough sound, body temperature, airflow, etc. of patients' or users. The recorded data was later converted and processed by a machine learning based techniques for identifying patterns and classifying the combined symptoms of various respiratory disorders [2].

Another study stated that Google AI scientists developed a neural network where they had reported that their network was better or as accurate as radiologists to detect malignant lung nodules [5]. A similar model was found to detect chronic obstructive pulmonary disease (COPD) in smokers to predict acute respiratory disease cases and mortality [6]. In paper [7], the authors learned that the machine learning algorithm worked well with the radiologists to interpret thoracic using high resolution computed tomography images by 73%. Their study also demonstrated that deep learning algorithms could be valuable in diagnosing interstitial lung disease. Similarly, in paper [8], the authors discovered that deep learning improved the diagnosis of chronic hypersensitivity pneumonitis, nonspecific interstitial pneumonia, cryptogenic organizing pneumonia, and typical interstitial pneumonia patterns. Therefore, it can be said that AI-based techniques have demonstrated superior performance and provide clinicians with a powerful decision support tool. The importance of such technology in improving clinical practice will drive medical community acceptance in the real world [9–12].

In this paper, the image dataset of four respiratory diseases like lung cancer, pulmonary embolism (PE), covid-19, and pneumoconiosis, including the normal lung images, have been considered and applied on various deep learning models such as EfficientNetB6, EfficientNetV2B1, EfficientNetV2B3, DenseNet201, Xception, ResNet50V2, Inception-v3, EfficientNetV2S, InceptionResNet-v2, ResNet101V2, and proposed hybrid model (EfficientNetB6 and ResNet101V2). The models are evaluated using several factors which include loss, accuracy, F1 score, MCC, precision, and recall. The research found that the proposed hybrid model obtained the highest training and testing accuracy of 99.84% and 99.77%, respectively.

1.1 Contribution

The contributions that have been made to develop the prediction system of respiratory diseases are shown as under:

1. The dataset of 19,488 images was initially taken from the four diseases, including the normal lungs, and later pre-processed by applying the CLAHE technique to enhance the contrast and remove noisy signals from images.
2. Using histogram equalization, the images have been visualized graphically to study the patterns of the pixel as well as detect anomalies if present.
3. For extracting features and to obtain ROI various techniques have been used such as contour features, Otsu thresholding, and Adaptive thresholding. It results in

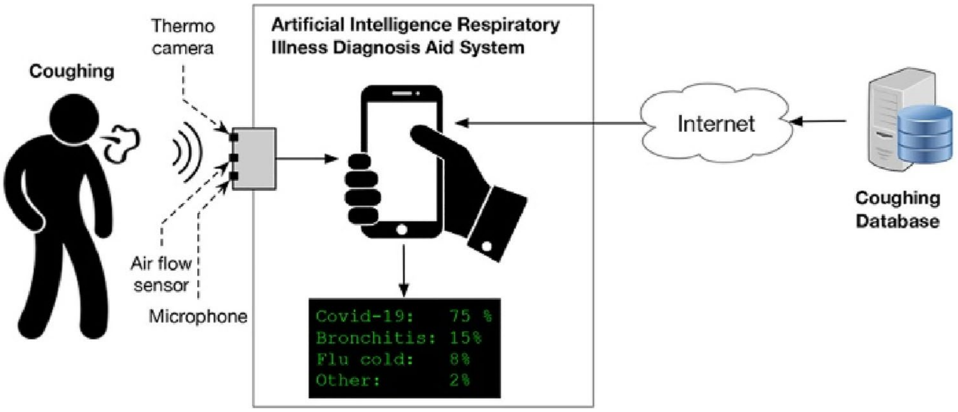


Fig. 1 AI based cough detection application [2]

38,876 image features and is later split into training and testing sets on a scale of 70:30.

4. Subsequently, ten deep transfer learning models along with the proposed hybridized model are applied and trained using training and testing dataset. These models are further examined through various parameters such as loss, accuracy, recall, F1 score, precision, and MCC values. Besides this, the values of confusion matrix to showcase the best model for identifying and classifying the respiratory diseases have been also generated.

1.2 Road Map of the Paper

The first section of the research paper is referred to as the Introduction. It provides concise information on respiratory diseases, their impact, and AI-based strategies for combating them. The context for discussing the researcher's work in the field of detecting respiratory disease is presented in Sect. 2. In addition, Sect. 3 describes the dataset, techniques, and parameters used to develop the respiratory disease detection system, while Sect. 4 describes the system's outcomes. In Sects. 5 and 6, the proposed work is contrasted with the existing one in the discussion, and the paper is summarized and concluded respectively.

2 Background

In this section, the work of various researchers in detecting lung cancer, covid 19, PE, and pneumoconiosis with the help of machine and deep learning techniques have been showcased [13, 14]. A tabular representation in Table 1 has also been provided to make it more informative, where the dataset, methods used by the researchers, and their outcomes and limitations have been shown. In the case of lung cancer detection, researchers like Dunke et al. [15] and Sori et al. [16] classified the lung nodules and detected their malignancy level using a 3D multi-path VGG network, U-Net architecture, and multi-phase CNN, respectively. Likewise, Chen et al. [17] researched lung cancer treatment to reflect better diagnosis by providing higher interpretability of the output. During the research, they analyzed the model's performance for a small imbalance dataset, and to overcome it, they proposed a new bag simulation method for multiple instance learning. Likewise, Said et al. [18] had discussed the use of deep learning techniques for accurate diagnosis of lung cancer through medical image segmentation. The study used a dataset of CT scans of lung cancer patients and compared the performance of different deep learning architectures for image segmentation. The results showed that the use of deep learning techniques significantly improved the accuracy of lung cancer diagnosis through medical image segmentation. The study had important implications for the future of lung cancer diagnosis and treatment.

In the case of pneumoconiosis, Sun et al. [19] proposed a fully deep learning technique that comprised segmentation and a staging procedure. The researchers initially segmented and extracted lung regions in the CXR images and later classified them into four stages using focal staging loss and deep log-normal label distribution learning. Similarly, Yang et al. [20] also developed the automatic pneumoconiosis screening system, which used pre-processing pipeline along with the ResNet classification model. As per the authors, a large set of data was used. In their paper, Zhang et al. [21] mentioned an AI-based model that assisted radiologists in screening and stage pneumoconiosis based on their CXR images. The model initially segmented the lung region into six subregions, and later CNN based network was applied to classify and predict the opacity level of each sub-region. Their research ended by diagnosing each area and classifying them under the classes such as normal, stage 1, stage II, or III based on their prediction results. Peng et al. [22] investigated the use of convolutional neural networks in medical images to enhance pneumoconiosis diagnosis. The research gathered 8361 chest X-ray films for the first round of model testing and 24,887 chest X-ray films for the third round of model testing. Three distinct models were designated with test sets, and the diagnostic efficacy of each was computed.

In the case of PE [23], the authors applied a novel approach for analyzing the incomplete and partial datasets based on *Q*-analysis and ML algorithms. The authors' main aim was to introduce the hybridization of the theory of hyper networks and supervised artificial neural networks. Using this strategy, they developed new computer-aided design software to detect PE for reducing the number of CT-angiography analyses and ensure a great efficiency of the diagnosis. Similarly, in [24], the researchers worked on CT-angiography images of PE by training weakly labelled data on a deep neural network. The author took a small dataset and proved that the results obtained were much better, and it demonstrated that small research groups could use DL models for limited resources. In [25], the authors stated that a CT exam is necessary for achieving fast detection and diagnosis of PE. So based on it, they proposed a pipeline-based technique that used U-Net (Fig. 2) to detect embolisms from CT images and classified them between true positives and false positives using machine learning algorithms.

Grenier et al. [26] developed a CNN model as well as the hybrid 3D/2D UNet topology to detect and suspect PEs on computed tomography angiograms (CTAs). They used the dataset of 387 anonymized real-world chest CTAs which had been acquired on 41 different scanner models. The results showed that their algorithm correctly identified 170 out of 186 positive PE cases with 91.4% sensitivity and 184 out of 201 negative PE cases with 91.5% specificity.

Table 1 Analysing the work of the researchers

References	Disease	Dataset	Technique	Outcomes	Limitation
[15]	Lung cancer	Lung image database	VGG, UNet	Accuracy = 94%	The data had been trained with the limited dataset
[19]	Pneumoconiosis	Chest radiograph images	Asymmetric Encoder-Decoder Network	Accuracy = 90.4%	Modelling errors
[25]	Pulmonary embolism	CAD-PE dataset	UNet	Dice score = 0.81 IoU = 0.79	In order to distinguish between small noise and other thrombus artifacts, the network was not improved
[27]	Covid 19	Data collected from Qatar University	Fuzzy color, stacking technique, deep learning model	Accuracy = 99.27%	The system did not function with the input images' low resolution and varied sizes
[17]	Lung cancer	Lung Image Database Consortium	Multiple instance learning	Accuracy = 80.7%	The model failed to detect the small nodules
[23]	Pulmonary embolism	Features of 1427 people	Neural Hyper network	Accuracy = 94%	The study had bias data due to which the results get affected
[20]	Pneumoconiosis	Chest X-ray images	ResNet34, UNet	Accuracy = 92.46%	The system did not work properly for all the images of dataset
[28]	Covid-19	Samples gathered at the Brazilian hospital Albert Einstein Israelite in Sao Paulo	Long Short Term Memory	Accuracy = 86.67%	The study's biggest drawback was the volume of data it contained
[16]	Lung cancer	Kaggle Data Science Bowl (KDSB)	DFD-Net, Convolution Neural Network denoising model	Specificity = 89.1% Accuracy = 87.8%	Class Imbalance
[24]	Pulmonary embolism	800 computed tomography pulmonary angiograms	InceptionResNet-v2	Specificity: 93.5% Sensitivity: 86.6%	Limited dataset
[21]	Pneumoconiosis	405 pneumoconiosis patients	AI techniques	Accuracy = 97.3%	Small dataset
[29]	Covid 19	Data gathered from the WHO (Jan 16–20, 2020)	Gated Recurrent Unit, Long Short Term Memory	Accuracy: 87%	The spatiotemporal elements of the LSTM network were not accurately modelled by the model
[18]	Lung Cancer	Decathlon dataset	UNETR	Accuracy: 97.83% Specificity:97.1% Sensitivity: 96.9%	Required high GPU performance
[22]	Pneumoconiosis	Lung X-ray image	ResNet Model	AUC: 0.987 Sensitivity: 96.6% Specificity: 98.6% Accuracy: 95.2%	-
[26]	Pulmonary Embolism	Real world dataset of CTA images	CNN	Sensitivity: 91.4% Specificity: 91.5% Accuracy: 91.5%	Lack of generalization
[30]	Covid-19	CXR images	EfficientNetB3	Accuracy: 96.64% Precision:87.5% Recall: 70% F1 Score:77.8%	Limited training dataset
		LCT images		Accuracy:99.66% Precision: 99.7% Recall:99.4% F1 Score: 99.5%	

To detect covid-19, the authors in [27] used the Fuzzy technique, MobileNetV2, Squeeze Net, and support vector machine. The data classes were restructured during the pre-processing phase and stacked with the original images. The MobileNetV2 and Squeeze Net models were then used to train the stacked dataset and support vector machine was used to combine and classify efficient features. Similarly, in [28], the researchers used deep learning and laboratory data to predict and estimate the covid-19 disease patients. The model was validated using tenfold cross-validation after testing 18 laboratory outcomes from 600 patients. In [29], the authors predicted that during COVID-19, children would experience stress, depression, and anxiety. To understand the children's stress, depression, and anxiety levels, a Deep Learning Neural Network (DLNN)-based method was used. Using cutting-edge Machine Learning techniques, Duong et al. [30] presented a practicable method to detect Covid-19 in chest X-ray (CXR) and lung computed tomography (LCT) images. The primary classification engine used the EfficientNet and MixNet technique on four real-world datasets i.e. two CXR datasets of 17,905 and 15,000 images, and two LCT datasets having 411,500 and 2,482 mages, respectively. The approach was evaluated using a five-fold cross-validation method, in which the dataset was divided into five parts where accuracy consistently exceeded 95.0% across all configurations, indicating a promising prediction performance across all datasets.

3 Methodology

This section addresses the many phases of the research, such as Sect. 3.1, which provides details about the dataset. Section 3.2 describe the procedure for pre-processing the data. Section 3.4 depict picture visualization graphically. Section 3.4 provide the methods for extracting the features, and Sect. 3.5 describe the models briefly. Finally, Sect. 3.6 give an overview of the parameters used to evaluate the model's performance. The flow of all these phases is shown in Fig. 3.

3.1 Dataset

The initial step in developing an automatic identification system for predicting and classifying airway disorders such as lung cancer, PE, pneumoconiosis, and covid-19 is to gather data from authorized sources. To fit the model, the lung cancer images are gathered from a dataset of chest CT scan images in.jpg or.png format [31]. Pneumoconiosis illness images are obtained from the Chongqing CDC through chest x-rays. It is divided into two subfolders: training and validation, which contain 568 and 140 images of pneumoconiosis-affected lungs and normal lungs, respectively [32]. Covid -19 pictures are obtained from the SARS-COV-2 Ct-Scan Dataset. This dataset includes 1230 covid negative and 1252 covid positive CT scans for a total of 2482 CT scans [33]. The images for PE were acquired from CT imaging of PE. The dataset consists of 35 different patients' computed tomography angiography (CTA) pictures for PE [34]. Finally, the images for normal lungs were extracted from all of the datasets mentioned above and merged to generate a single dataset. Figure 4 depicts the original images of numerous airway illnesses including normal lungs used in the research.

3.2 Pre-processing

Following the collection of images of size $224 \times 224 \times 1$ from various disease datasets, pre-processing has been performed to improve the characteristics and remove the noisy signals so that they may be readily evaluated better. The CLAHE approach, which stands for contrast limited adaptive histogram equalization, is useful for medical images as it improves the contrast of an original image by dividing it into small regions called tiles and equalizing the histogram of each tile separately. The approach is adaptive because it adjusts the contrast enhancement locally to the characteristics of each tile, which can vary within an image. This allows it to handle images with large variations in illumination and contrast. Two important parameters of the CLAHE approach are the clip limit and the tile grid size. The clip limit, specified by the cliplimit() function, sets a threshold on the amount of contrast enhancement applied to each tile. This limit prevents over-amplification of the contrast, which can lead to the loss of image details and the introduction of artifacts. In this study, the clip limit has been set to 10, which is lower than the default value 40. The tile grid size, compute by the tileGridSize() function, determines the number of tiles used to divide the image for histogram equalization. The size of the tiles is important because it affects the trade-off between local and global contrast enhancement. A larger tile size leads to more global contrast enhancement, while a smaller tile_size enhances more local contrast. In this approach, the image is initially divided into non-overlapping tiles of equal size, and each tile is processed independently. After CLAHE technique is applied to each tile, the generated tiles are merged using bilinear interpolation to provide a more contrasted and visible output image. Figure 5 shows the output images obtained using the CLAHE approach with the specified parameters.

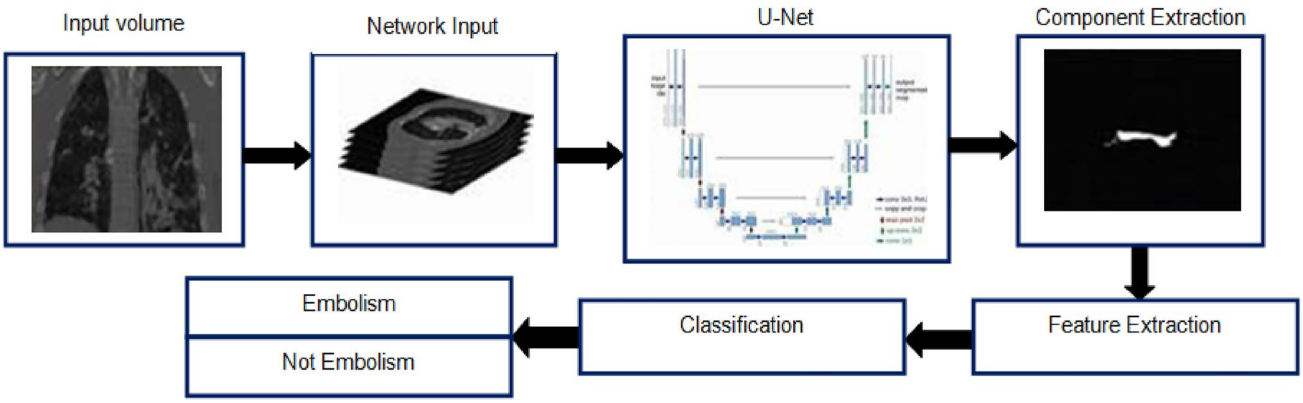


Fig. 2 UNet architecture to detect pulmonary embolism

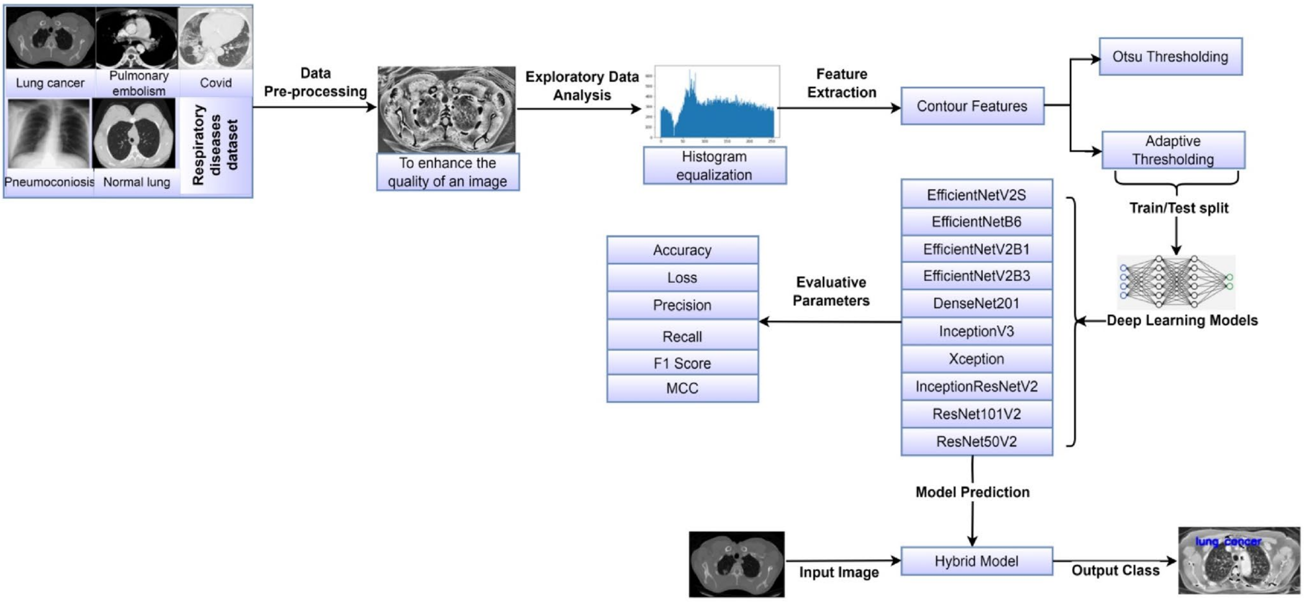


Fig. 3 Proposed system design for respiratory diseases detection and classification

3.3 Exploratory Data Analysis

During pre and post-processing of original images, the histogram using hist() has been generated to find out the pattern of an image. Figure 6a represents the histogram of original images which indicates that the pixel intensity distribution of images is not uniform and have noisy signals in it. On the contrary, after applying the contrast enhancement technique to the images, it can be seen in Fig. 6b that the technique improves the visualization of certain features in the image as well as reduces the noise.

3.4 Feature Extraction

In this section, after obtaining the histogram equalized images, the features have been extracted using various image augmentation techniques. Initially contour feature is used for finding out the extreme points to crop the images for obtaining the desired region using threshold techniques. During this phase, the properties of the images have been generated by calculating the parameters such as area, epsilon, perimeter, height, width, extent, equivalent diameter, minimum value, aspect ratio, maximum value, min value location, max value location, extreme leftmost point, extreme rightmost point, mean color, extreme topmost point, and extreme

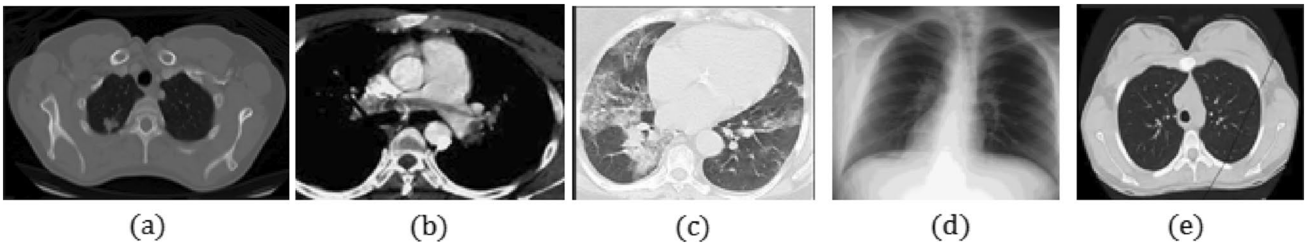


Fig. 4 Original images of **a** lung cancer, **b** PE, **c** Covid-19, **d** pneumoconiosis, **e** normal lungs taken from respective dataset

bottommost point using Eqs. (1) to (16). All the computed values are displayed in Table 2.

Initially, we calculated area, which is the product of height and width and based on it, the aspect ratio has also been calculated. The equations to compute them are (1) and (2)

$$\text{area} = \text{height} \times \text{width} \tag{1}$$

$$\text{Aspect ratio} = \frac{\text{width}}{\text{height}} \tag{2}$$

Further, other parameters such as height and width are computed, as shown in Eqs. (3, 5), depending on the contour feature points which are being passed to the bounding rectangle function through the OpenCV library.

$$\text{height} = cv2.boundingRect(cnt) \tag{3}$$

$$\text{width} = cv2.boundingRect(cnt) \tag{4}$$

Moreover, perimeter, equivalent diameter, extent, and epsilon are also calculated using Eqs. (5–8). Perimeter is computed through arclength, and the extent is the ratio of an object's area to the bounding rectangle area. Diameter is similar to the image's contour area, and in the end, epsilon is used for calculating the distance between the two points of the same classes.

$$\text{epsilon} = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

$$\text{Perimeter} = 0.1 \times cv2 \times arclength(cnt, True) \tag{6}$$

$$\text{Extent} = \frac{\text{object area}}{\text{bounding rectangle area}} \tag{7}$$

$$\text{Equivalent diameter} = \sqrt{\frac{4 \times \text{contour area}}{\pi}} \tag{8}$$

In addition to this, max and min value location as well as max and min values of the feature is calculated along with the value of color intensity as shown in Eqs. (9–12)

$$\text{Minimum value location} = cv2.\text{minMaxLo()} \tag{9}$$

$$\text{Maximum value location} = cv2.\text{minMaxLo()} \tag{10}$$

$$\text{Minimum value} = cv2.\text{min()} \tag{11}$$

$$\text{Maximum value} = cv2.\text{max()} \tag{12}$$

$$\text{Mean color} = \text{cv2.mean()} \tag{13}$$

In the end, extreme leftmost, rightmost, bottommost, and topmost values are also calculated, in which 0 stands for the extreme left and rightmost point which means the calculation of values takes place in the horizontal direction. At the same time, 1 refers to calculating values in the vertical direction for extreme bottommost and topmost points.

$$\text{Extreme leftmost point} = tuple(cnt(cnt[:, :, 0].argmin()[0]) \tag{14}$$

$$\text{Extreme rightmost point} = tuple(cnt(cnt[:, :, 0].argmin()[0]) \tag{15}$$

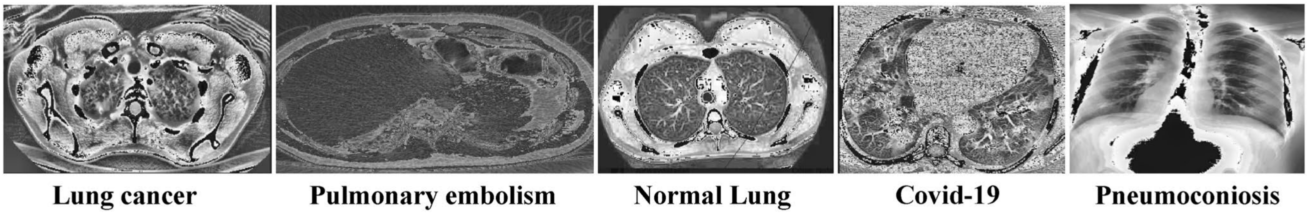
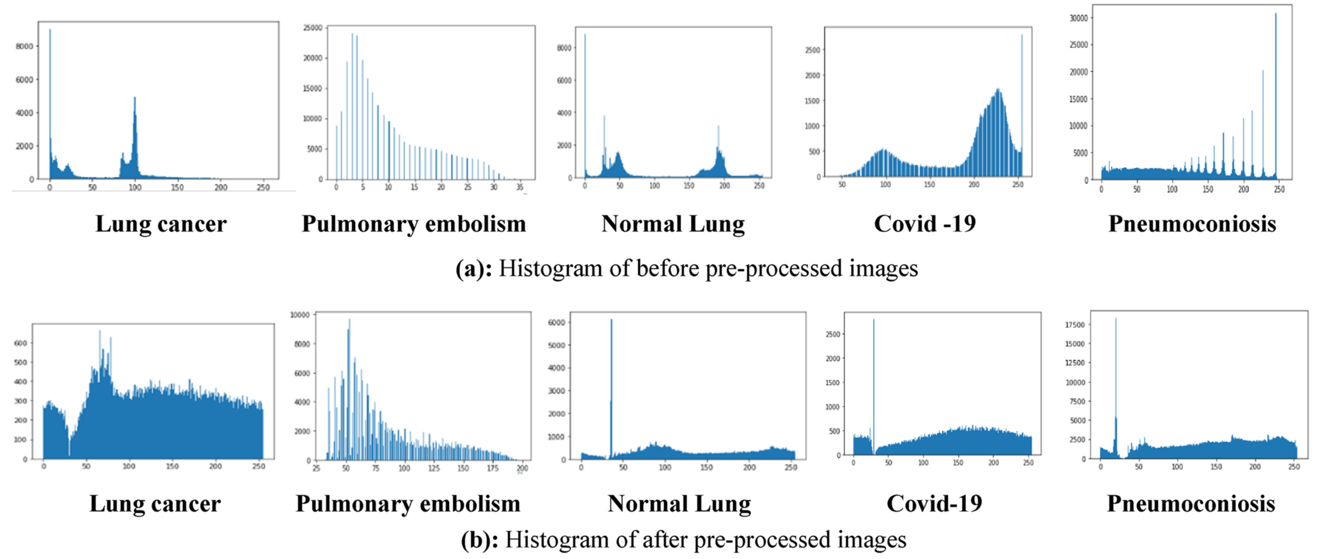

Lung cancer
Pulmonary embolism
Normal Lung
Covid-19
Pneumoconiosis

Fig. 5 Pre processed images of various respiratory diseases


Lung cancer Pulmonary embolism Normal Lung Covid -19 Pneumoconiosis

(a): Histogram of before pre-processed images

Lung cancer Pulmonary embolism Normal Lung Covid-19 Pneumoconiosis

(b): Histogram of after pre-processed images

Fig. 6 **a** Histogram of before pre-processed images. **b** Histogram of after pre-processed images

Table 2 Characteristics of different images of respiratory diseases

Parameters	Lung cancer	Covid-19	Pneumoconiosis	Pulmonary embolism	Normal lungs
Area	4.0	5.5	2.0	7.0	2.0
Perimeter	7.6	9.1	5.6	16.0	5.6
Epsilon	0.7	0.9	0.5	1.6	0.5
Width	3	4	3	8	3
Height	4	4	3	2	3
Aspect ration	0.75	1.0	1.0	4.0	1.0
Extend	0.3	0.3	0.2	0.4	0.2
Equivalent diameter	2.2	2.7	1.5	2.9	1.59
Minimum value	126.0	120	127.0	128.0	127.0
Maximum value	142.0	135	167.0	128.0	255.0
Min value location	39,161	159;219	298,442	240,427	419,338
Max value location	39,163	158;221	298,443	240,427	419,339
Mean color	133.0	129	138	128.0	155.0
Extreme leftmost point	38,161	157,219	297,442	240,427	418,338
Extreme rightmost point	40,161	160;219	297,442	247,428	420,338
Extreme topmost point	39,160	158;218	298,441	240,427	419,337
Extreme bottommost point	39,163	158;221	298,443	240,428	419,339

$$\text{Extreme topmost point} = tuple(cnt(cnt[:, :, 1].argmin()[0]) \tag{16}$$

$$\text{Extreme bottommost point} = tuple(cnt(cnt[:, :, 1].argmin()[0]) \tag{17}$$

Using cv2.contour (), the morphological values of the various contour features are used to determine the largest contour (). The contours are the curves that connect all continuous points (along the boundary) that have the same hue or intensity. They are helpful for analyzing the shape as well as item identification and recognition. They are utilized in this research to generate the extreme points for cropping the image so that the characteristics can be extracted and extraneous information or details can be avoided to save space and time. The colors red, green, blue, and teal are used to define the extreme points for x–y coordinates, and they are determined using argmax() and argmin() as shown in Fig. 7.

Further, the cropped images are segmented to obtain the region of interest by generating the bounding box using Otsu and adaptive thresholding techniques as shown in

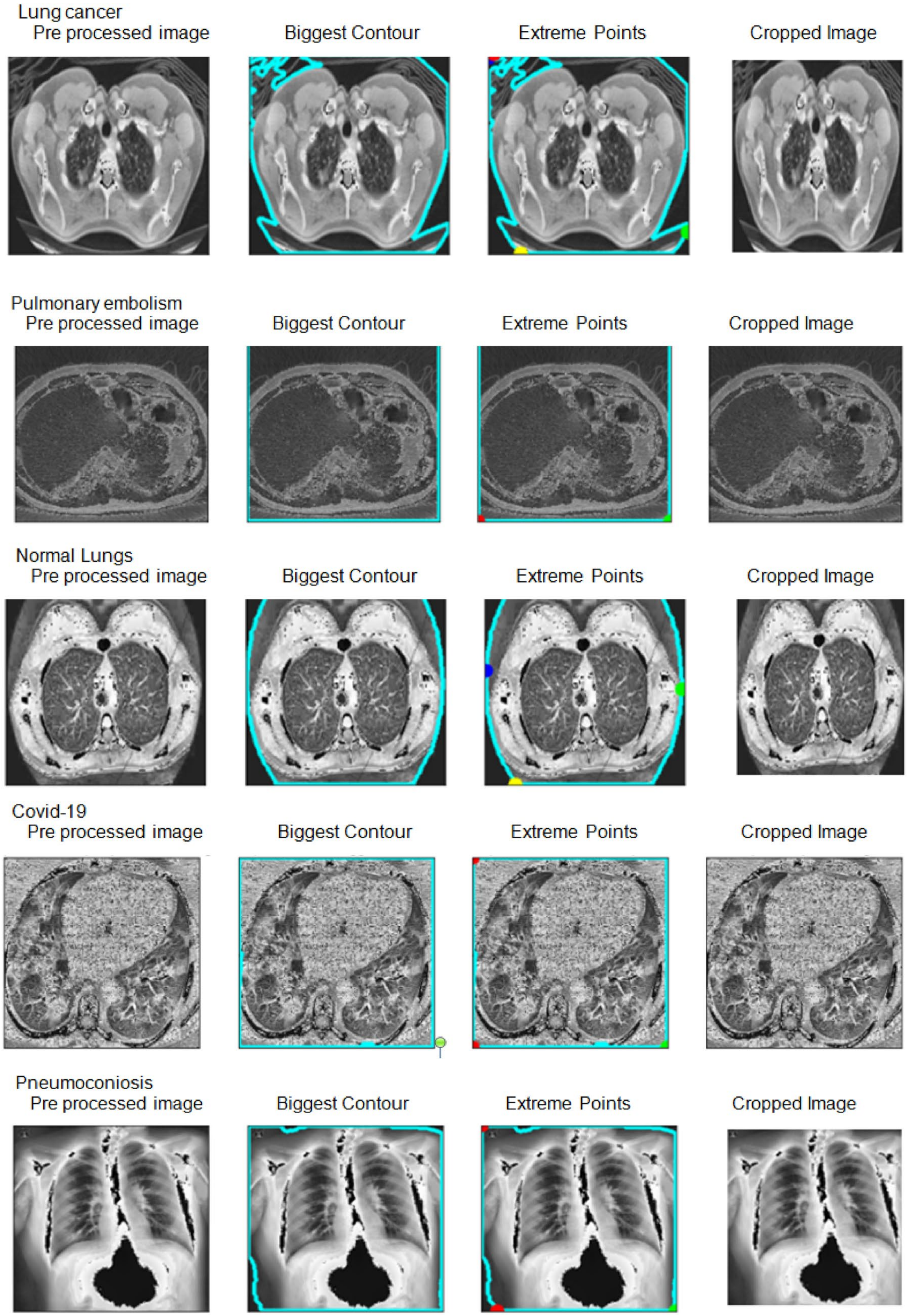


Fig. 7 Applying feature extraction on multiple respiratory diseases. (Color figure online)

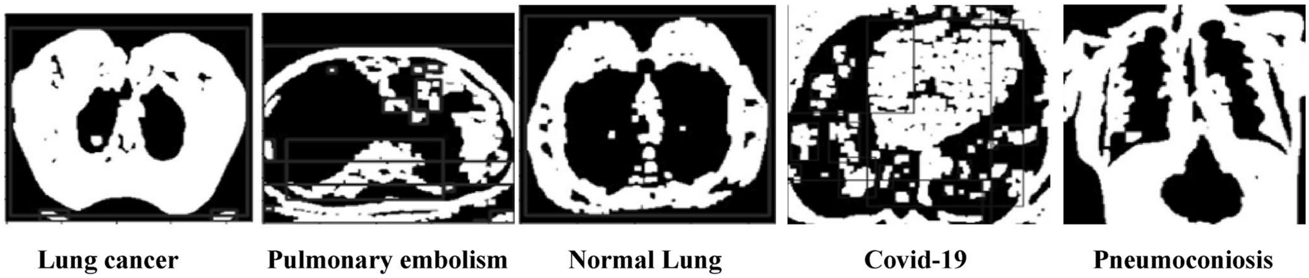


Fig. 8 Images after applying Otsu thresholding

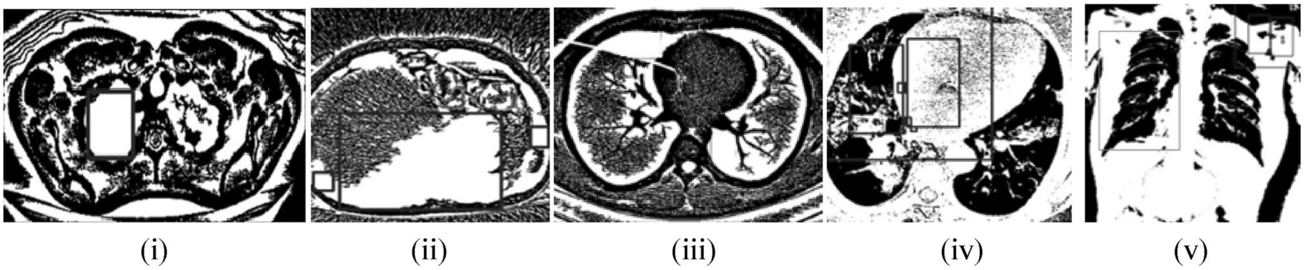


Fig. 9 Images after applying adaptive thresholding **i** lung cancer, **ii** PE, **iii** normal lung, **iv** Covid-19, **v** pneumoconiosis

Figs. 8 and 9 respectively which results in to 38,976 image features. The Otsu method, also known as the binarization algorithm, is an uncomplicated and efficient automatic thresholding technique. The results of thresholding technique are generated using cv2.Otsu(). An image is consisted of two classes i.e. background and foreground. Otsu technique helps to compute an optimized threshold value which minimize and maximize the intra-class variance (σ_{wc}) and the inter-class variance (σ_{bc}) of these five classes respectively. Two variances i.e. σ_{wc} and σ_{bc}, are calculated using eq (xviii) and (xix) respectively for all possible thresholds (thresh = 0 to I, i.e., maximum intensity level). In the end, if the value of the pixel luminance is less or equal to the threshold, it is replaced by 0 (black), and if greater than the threshold, it is replaced by 1 (white) in order to obtain the binary or black/white image.

$$\sigma^2_{wc}t = \omega_1(t)\sigma^2_1 t + \omega_2(t)\sigma^2_2 t \tag{18}$$

$$\sigma^2_{bc}t = \sigma^2 - \sigma^2_w t \tag{19}$$

where, weights $\boldsymbol{\omega}_1(\boldsymbol{t})$ refers to the probabilities which are separated by a threshold t of the two classes. σ_1 and σ_2 are the variances of these two classes [35].

An adaptive threshold is also being chosen on the basis of the statistical properties of the pre-processed images which are cropped after their extreme points have been generated. The function cv2.adaptiveThresholding() used for weight updating unit in order to find an acceptable threshold value for the images that are bimodal in nature. Consider a size [W × H] image and assign two weights, μ_1 and μ_2 and later compare them to each and every pixel value in the [W × H] image. Later, the weight which is closest to the pixel value is selected for updating the weight of each input pixel. Further, the variation between the closest and input weight is multiplied by the learning rate β and is added to the closest weight. If μ_1 is close to that value of the pixel, μ_1 is updated, and if μ_2 is close to that pixel, μ_2 is updated by applying the following Eqs. (20).

$$\mu_{new} = \mu_{old} + \beta \times (pixel - \mu_{old}) \tag{20}$$

The updated weights are applied to each image pixel as well as the average of these two weights is used as the value of the threshold; Eq. (20) describes it. This threshold setting can be used to convert a picture to binary form [36].

$$a_{th} = \frac{\mu_1 + \mu_2}{2} \tag{21}$$

The pixel that ranges above a_{th} value is considered as object and the value that ranges below a_{th} value are consider as background.

Table 3 Hyper-parameters of applied deep learning models

Parameters	Value
Size of batch	128
Learning rate	0.0001
Epochs	15
Optimizer	Adam
Shear range	0.1
Classes	5
Width shift range	0.10
Zoom range	0.1
Horizontal flip	True
Steps per epoch	107
Height shift range	0.10

3.5 Classifiers

This section briefly describes about all deep learning models that have been applied to the dataset (Sect. 3.1) for predicting and classifying airway diseases. In addition to this, their hyper-parameter values are also shown in Table 3 that has been kept fixed throughout the research.

EfficientNet It is a convolutional neural network based scaling and design method that uses a compound coefficient to scale all depth/width/resolution dimensions consistently (Fig. 10). EfficientNet is constructed upon the foundational network derived from the neural architecture search conducted by the AutoML MNAS framework. The architectural design incorporates a mobile inverted bottleneck convolution technique, which bears resemblance to the MobileNet V2 model. However, it should be noted that this architecture exhibits an increase in size, primarily attributed to the corresponding rise in floating-point operations per second (FLOPS) [37]. In this paper, four new efficientNet series were used: EfficientNetB6 (total parameters 40,970,656, Traianable parameters 40,746,221, and Non-trainable params: 224,435), EfficientNetV2B3 (Total params: 12,937,587, Trainable params: 12,828,371, Non-trainable params: 109,216), EfficientNetV2B1 (Total params 6,934,391, Trainable params: 6,863,319, Non-trainable params: 71,072), and EfficientNetV2S (Total params 20,337,333, Trainable params: 20,183,461, Non-trainable params: 153,872).

DenseNet201 DenseNet201, in Fig. 11, has the property of reusing features with the help of its multiple layers, which increases variation in the subsequent layer input and enhances performance. This model has a more complex and denser network where all the layers are linked together with shorter connections in order to efficiently train and generate results [38]. The total number of parameters generated by the DenseNet201 model in this study is 18,321,475, of which 18,092,419 are trained and 229,056 are untrained.

Inception-v3 As shown in Fig. 12, the model contains 42 layers and has a lower error rate than Inception v1 and Inception V2. The core building block of Inception v3 is the inception module. Each inception module comprises parallel branches of different filter sizes, including 1 × 1, 3 × 3, and 5 × 5 convolutions. These branches are designed to capture features at various spatial scales. In addition, 1 × 1 convolutions are used within the inception module to reduce the number of channels and control computational complexity. The outputs of all branches are concatenated along the channel dimension, providing a rich set of multi-scale features [39].

In this study, the Inception-v3 model generated a total of 21,808,355 parameters, of which 21,773,923 are trained and 34,432 are not.

Xception The Xception model, as shown in Fig. 13, comprises of multiple modules called Xception blocks. Each Xception block consists of a sequence of depthwise separable convolutions, batch normalization, and nonlinear activation functions. The residual connections from the Inception-ResNet architecture are also incorporated into Xception blocks to facilitate gradient flow and ease optimization. The

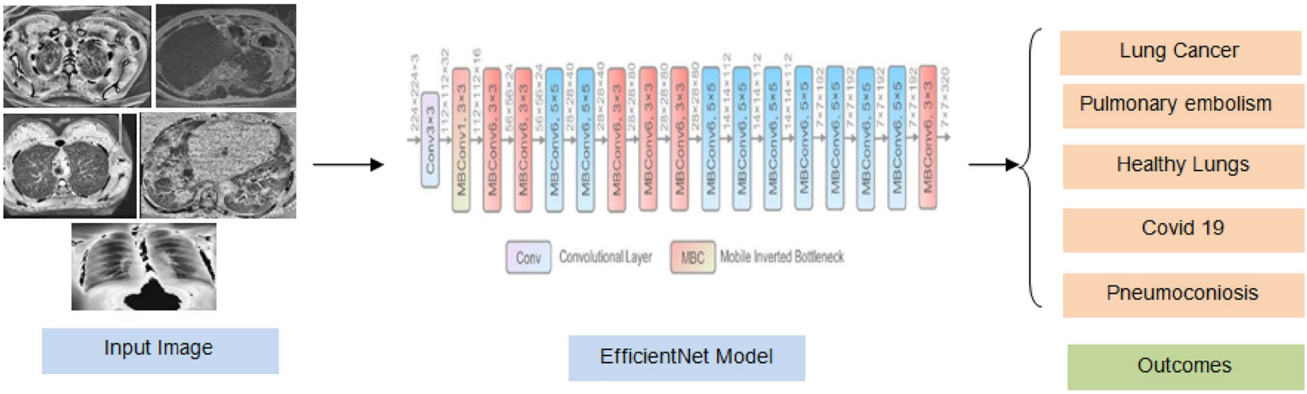


Fig. 10 Architecture of EfficientNet model

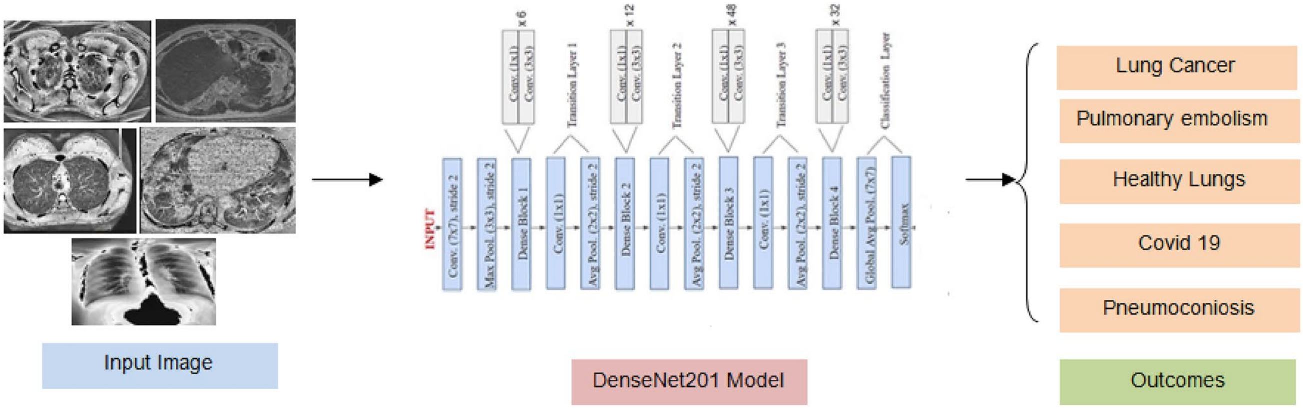


Fig. 11 Architecture of DenseNet201 model

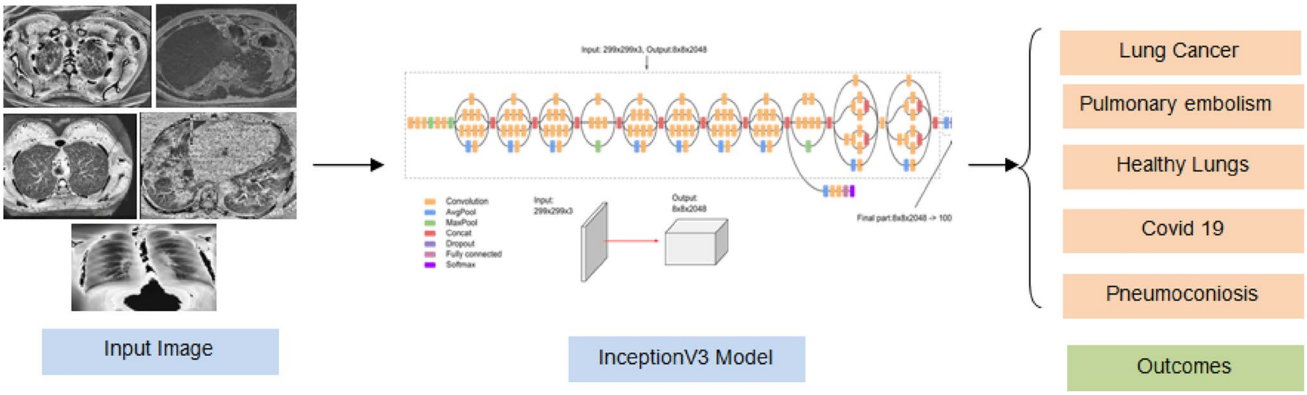


Fig. 12 Architecture of Inception-v3 model

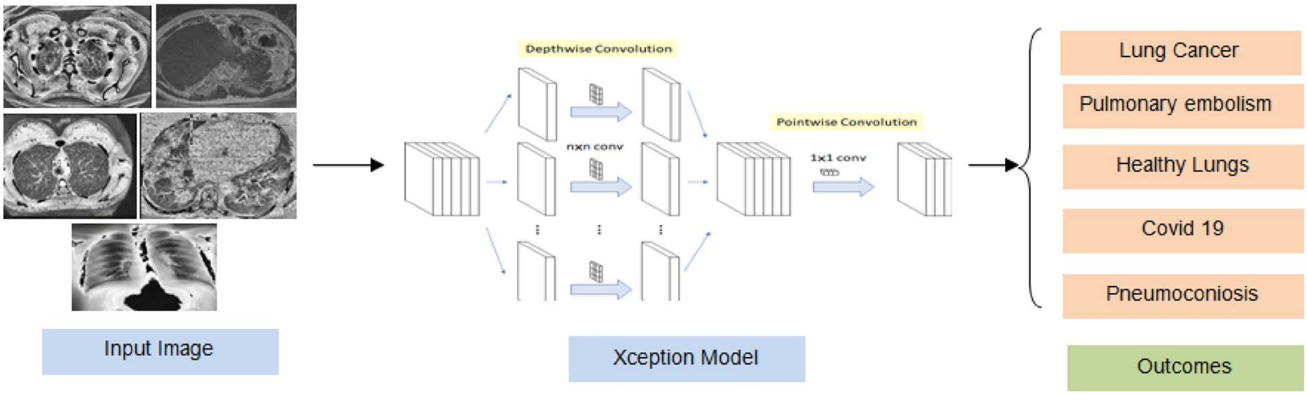


Fig. 13 Architecture of Xception model

Xception model typically concludes with global average pooling and a fully connected layer with softmax activation for classification. The global average pooling reduces the spatial dimensions to a vector representation, and the fully connected layer generates class probabilities [40]. In this study, the Xception model produced an entire set of 20,867,051 parameters, of which 20,812,523 are trained, and 54,528 are not.

ResNet50V2 The modified version of ResNet50 is ResNet50V2 (Fig. 14), and it performs better on the ImageNet dataset than ResNet50 and ResNet101. ResNet50V2 is organized into multiple stages, including Stage 01, Stage

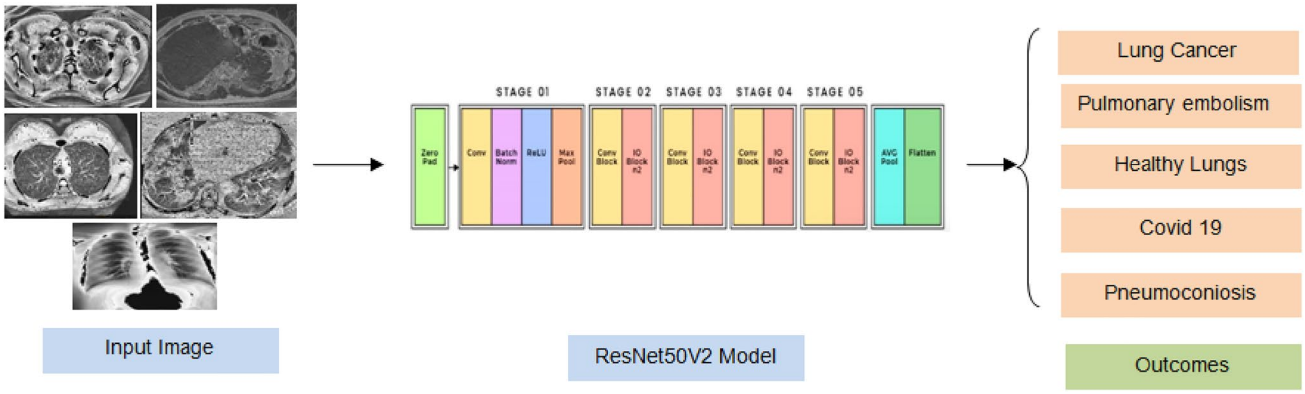


Fig. 14 Architecture of ResNet50V2 model

02, Stage 03, Stage 04, and Stage 05, each containing several residual blocks. The feature map sizes decrease as the network goes deeper, capturing features at different scales. The architecture also uses a bottleneck design within each residual block, consisting of 1 × 1 convolutions to reduce dimensionality, 3 × 3 convolutions for feature extraction, and another 1 × 1 convolution for dimension restoration. This bottleneck architecture reduces computational complexity and allows for more efficient feature learning [41].

The total number of parameters generated by the ResNet50V2 model in this study is 23,564,675, of which 23,519,235 are trained, and 45,440 are untrained.

InceptionResNet-v2 The core building blocks of Inception-ResNetV2 are the Inception blocks. These blocks capture multi-scale features crucial for understanding complex visual patterns. Each Inception block contains parallel branches with different filter sizes and pooling operations. By operating in parallel, the network can capture and efficiently combine features at various scales. One notable feature of Inception-ResNetV2 is the incorporation of residual connections. Residual connections allow for the direct propagation of information from earlier layers to later layers. This enables smoother gradient flow during training and helps alleviate the vanishing gradient problem, which can hinder the training of very deep networks. The residual connections also contribute to the network's ability to learn shallow and deep features effectively. Inception-ResNetV2 architecture also includes auxiliary classifiers. The auxiliary classifiers typically combine convolutional layers, pooling layers, and fully connected layers. These classifiers are inserted at intermediate stages of the network and help with gradient propagation during training. They encourage the network to learn more meaningful representations and prevent overfitting [42].

Figure 15 depicts the InceptionResNet-v2 basic block diagram. In this study, the overall number of parameters generated by the InceptionResV2 model is 54,343,845, 54,283,301 are trained, and 60,544 are untrained.

ResNet101V2 The ResNet101V2 architecture consists of 101 layers and is widely used in computer vision tasks like image classification and object detection. The network starts with an input layer that takes an image as an input and is sent to the convolutional layers that extract low-level features from that input image. The key innovation of ResNet101V2 lies in its residual

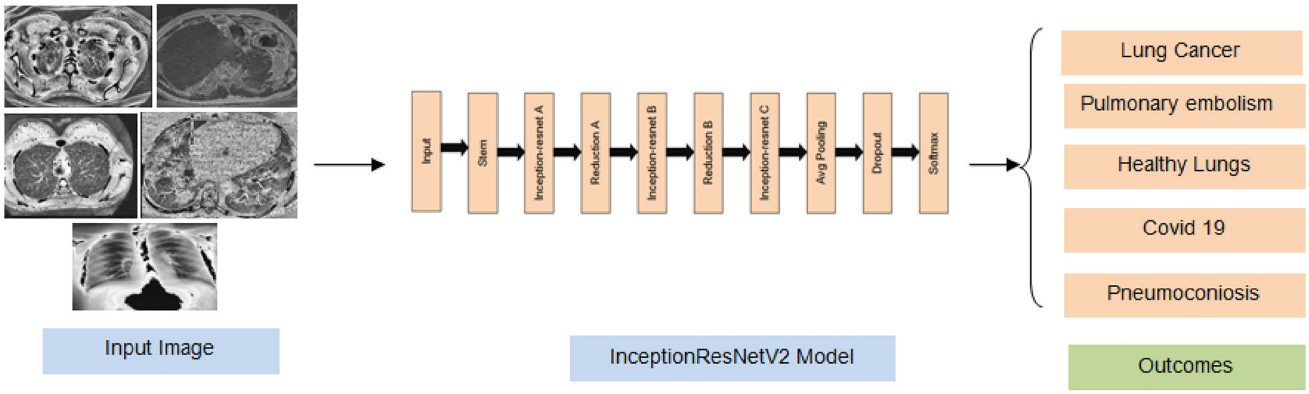


Fig. 15 Architecture of InceptionResNet-v2

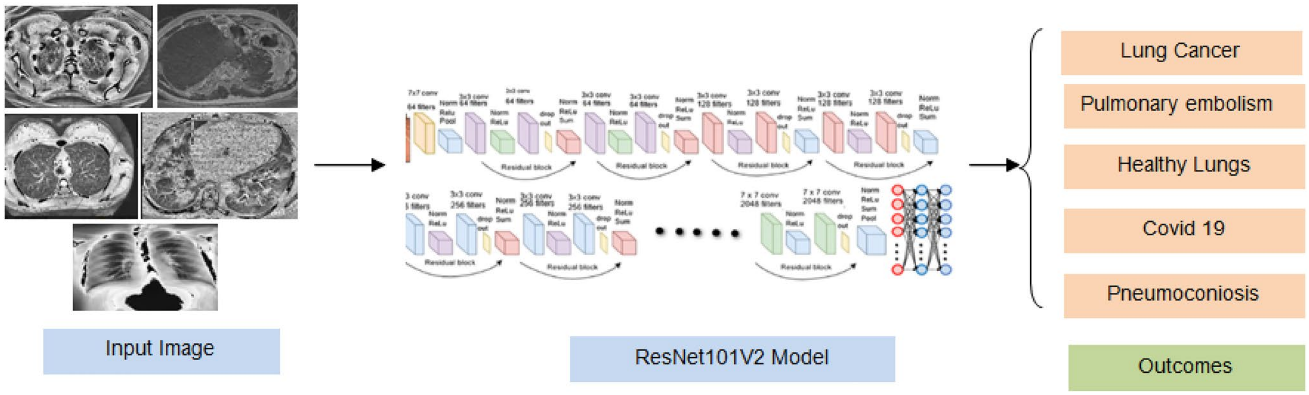


Fig. 16 Architecture of ResNet101V2

blocks, which include skip or shortcut connections. These connections enable the network to learn residual mappings by preserving the input and combining it with the output of convolutional layers. The residual blocks also employ a bottleneck structure, which reduces the dimensionality of feature maps to improve efficiency without degrading the performance. Further, global average pooling reduces spatial dimensions, followed by fully connected layers for final classification or regression. Activation functions, such as ReLU, introduce non-linearity and shortcut connections to ensure the flow of gradients during training [43].

Overall, ResNet101V2 is a powerful architecture that leverages skip connections and bottleneck structures to train deep networks effectively and extract meaningful features for visual tasks. In this research, the ResNet101V2 model (Fig. 16) produced a total of 42,630,533 parameters, of which 42,532,869 are trained and 97,664 are untrained.

Proposed hybrid transfer learning model This proposed hybrid model is composed of two pre trained models such as *EfficientNetB6* and *ResNet101V2* which are being trained with an input size of 224×224 and has generated 83,601,949 parameters, out of which 83,279,085 are trainable parameters, and 322,099 are non-trainable parameters as shown in Fig. 17.

The layered structure of EfficientNetB6 consists of *one* input layer, *one* rescaling layer, *one* normalization layer, *two* 2D convolution layers, *two* batch normalization layers, *two* activation layers. The architecture also contains seven blocks as well as sub blocks which are connected sequentially. In blocks 1 and 7, there are three sub blocks that consist of *one* Global AveragePooling 2D layer, *one* reshape layer, *three* 2D convolution layers, *one* multiply layer, *two* batch normalization layers, *one* drop out layer, *one* activation layer, and *one* add layer each. Likewise, from block 2 to block 6, the eight sub blocks consist of *four* 2D convolution layers, *two* batch normalization layers, *one* activation layer, *one* zero padding 2D layer, *one* depth wise 2D convolution layer, *one* global average pooling 2D layer, and *one* reshape, add as well as multiply layer each.

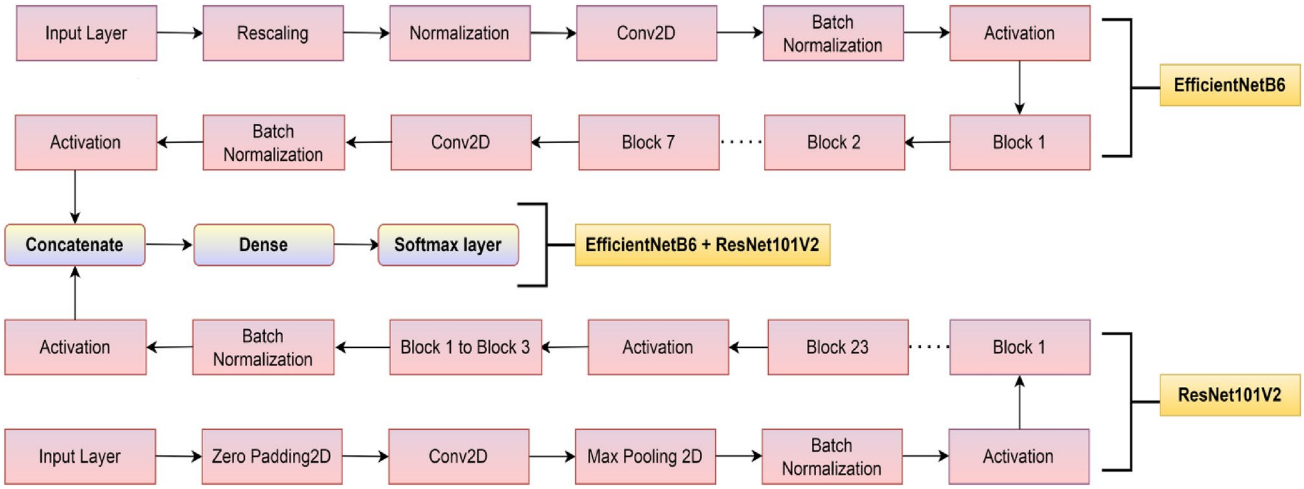


Fig. 17 Layered structure of proposed hybrid model

On the other side, the layered architecture of ResNet101V2 consists of *one* input layer, *one* zero padding 2D layer, *two* 2D convolution layers, *one* max pooling 2D layer, *two* batch normalization layers, and *three* activation layers. The architecture too contains three blocks which are initially followed by twenty three blocks via one activation layer. Each block is having sub blocks which consist of *three* 2D convolution layers, *two* batch normalization layers, *two* activation layers, *one* zero padding 2D layer, and *one* add layer.

Later the output activation layer of each model is concatenated at the concatenate layer which is further connected to dense and softmax layer from where the possibilities of predicting the class of airway diseases are obtained.

Besides it, the architecture has also been shown in Table 4 where the layers have been taken randomly but sequentially to provide the gist and parameters of EfficientNetB6 + ResNet101V2.

The columns that have been depicted in the table contains **Layer** which lists the name or type of each layer in the model, **Output Shape** to indicate the shape of the output tensor or feature map produced by each layer. The shape is represented as (batch_size, height, width, channels), and **Param#** to show the number of parameters (weights and biases) associated with each layer. The description about each layer is as followed:

input_5 (InputLayer) This is the input layer of the model, expecting input tensors with a shape of (None, 224, 224, 1). "None" represents a variable batch size, 224 × 224 is the input image size, and 1 is the number of channels (grayscale).

Table 4 Architecture of proposed hybrid model (EfficientNetB6 + ResNet101V2)

Layer	Output shapre	Param#
input$_5$(InputLayer)	None, 224, 224, 16	0
rescaling_2(Rescaling)	None, 224,224,1	0
normalization_2 (Normalization)	None, 224, 224, 1	3
stem_conv_pad (ZeroPadding2D)	None, 225, 225, 1	0
stem_conv (Conv2D)	None, 112, 112, 56	504
stem_bn (BatchNormalization)	None, 112, 112, 56	224
stem_activation (Activation)	None, 112, 112, 56	0
block1a_dwconv (DepthwiseConv2	None, 112, 112, 56	504
$\text{conv2}_{\text{block1}_{1_{\text{conv}\,(\text{Conv2D})}}}$	None, 56, 56, 64	4096
block7c_project_conv (Conv2D)	None, 7, 7, 576	1,990,656
$\text{conv5}_{\text{block2}_{2_{\text{conv}\,(\text{Conv2D})}}}$	None, 7, 7, 512	2,359,296
conv5_block3_3_conv (Conv2D)	None, 7, 7, 2048	1,050,624
top_conv (Conv2D)	None, 7, 7, 2304	1,327,104
conv5_block3_out (Add)	None, 7, 7, 2048	0
top_bn (BatchNormalization	None, 7, 7, 2304	9216
post_bn (BatchNormalization)	None, 7, 7, 2048	8192
Activation	None, 7, 7, 2048	0
concatenate_2	None, 7, 7, 4352	0
dense_2 (Dense)	None, 7, 7, 5	21,765

rescaling_2 (Rescaling) This layer rescales the input data, so the values fall within a specific range.

normalization_2 (Normalization) This layer normalizes the input data, making it have zero mean and unit variance.

stem_conv_pad (ZeroPadding2D) This layer adds zero-padding to the input tensor.

stem_conv (Conv2D) It applies convolutional operations to the input and produces an output tensor with a shape of (None, 112, 112, 56).

stem_bn (BatchNormalization) This layer performs batch normalization on the previous output tensor.

stem_activation (Activation) It applies an activation function to introduce non-linearity to the tensor.

block1a_dwconv (DepthwiseConv2D) This layer performs depthwise convolution, which applies separate convolutions to each input channel.

conv2_block1_1_conv (Conv2D) This convolutional layer produces an output tensor with a shape of (None, 56, 56, 64).

block7c_project_conv (Conv2D) This layer applies convolution to the input tensor, resulting in an output tensor with a shape of (None, 7, 7, 576).

conv5_block2_2_conv (Conv2D) It performs convolution on the input tensor, generating an output tensor of shape (None, 7, 7, 512).

conv5_block3_3_conv (Conv2D) This convolutional layer produces an output tensor with a shape of (None, 7, 7, 2048).

top_conv (Conv2D) It applies convolution to the input tensor, resulting in an output tensor with a shape of (None, 7, 7, 2304).

conv5_block3_out (Add) This layer performs element-wise addition between two input tensors.

top_bn (BatchNormalization) It performs batch normalization on the previous output tensor.

post_bn (BatchNormalization) This layer applies batch normalization to the input tensor.

Activation It applies an activation function to the tensor.

concatenate_2 This layer concatenates multiple input tensors along the channel axis.

dense_2 (Dense) It is a fully connected (dense) layer that produces an output tensor with a shape of (None, 7, 7, 5).

In a nutshell, the table provides a summary of the architecture, input/output shapes, and parameter counts for each layer in the model.

3.6 Evaluation parameters

The aforementioned applied models have to be now evaluated to test their performances and for that certain parameters are required which are described in this section.

Accuracy The parameter measures the efficiency of the model to correctly classify the image of any respiratory disease [44]. It is calculated by Eq. (22)

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \tag{22}$$

Loss This parameter is used for predicting the discrepancy between the actual and the predicted values. If the loss value generated is nearer to zero, it implies that the model works best else it should be re-trained [45]. It is calculated by Eq. (23)

$$\text{Loss} = \frac{(\text{Actual} - \text{Predicted})^2}{\text{Total number of observations}} \tag{23}$$

Precision and Recall These parameters are used to examine the model in terms of the positive prediction [46]. These both metrics are calculated by Eqs. (24) and (25) respectively.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{24}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{25}$$

F1 score It is the parameter that evaluates the performance of the classifier especially in those scenarios where both precision and recall are important and are needed to be balanced efficiently [47]. It is represented by Eq. (26)

$$F1\ \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{26}$$

Matthew's Correlation coefficient The Matthews correlation coefficient (MCC) is a parameter which depends on the values of confusion matrix as well as defines the quality of the prediction of classifier [48]. It is computed using Eq. (17)

$$\text{MCC} = \frac{\text{True Positive} \times \text{True negative} - \text{False negative} \times \text{False positive}}{\sqrt{(\text{True positive} + \text{False positive})(\text{True positive} + \text{False negative})(\text{True negative} + \text{False positive})(\text{True negative} + \text{False negative})}} \tag{27}$$

4 Experimental Results

Table 5 Evaluation of models for various respiratory diseases

Algorithms	Training Phase		Testing Phase	
	Accuracy	Loss	Accuracy	Loss
EfficientNetB6#	***98.99***	***0.02***	***99.26***	***0.01***
EfficientNetV2B3	99.06	0.02	93.96	0.14
DenseNet201	99.78	0.06	98.71	0.04
Xception#	***99.75***	***0.01***	98.85	0.04
EfficienetNetV2B1	99.27	0.07	69.48	0.84
ResNetV2_50	99.65	0.08	98.54	0.05
Inception_V3	99.23	0.01	98.88	0.02
EfficientNetV2S	99.05	0.02	98.54	0.03
InceptionResNet-v2	99.69	0.07	99.03	0.02
ResNet101V2[a]	99.62	0.09	***99.13***	***0.01***
Hybrid(EfficientNetB 6+ResNet101V2)	**99.84**	**0.03**	**99.77**	**0.001**

[a] The top two models have been bold and *italicized* based on their best training as well as testing accuracy and loss values

This section reflects that the models such as EfficientNetB6, EfficientNetV2B3, DenseNet201, Inception-v3, Xception, EfficientNetV2B1, ResNet50V2, EfficientNetV2S, InceptionResNet-v2, ResNet101V2, and Hybrid (EfficientNetB6+ResNet101V2) have been evaluated using the parameters as mentioned in Sect. 3.6 to test their performances for different diseases dataset.

Initially, the models are evaluated for their training and testing accuracy as well as loss. Later the confusion matrix of 5×5 for classes C0 as lung cancer, C1 as PE, C2 as covid 19, C3 as normal lungs, and C4 as pneumoconiosis has been generated to evaluate the performance of the classification models for the said classes in Fig. 16 in order to compare their actual target values with the predicted ones.

In Table 5, during the training phase, the EfficeintNetB6 and Xception generated the best accuracy of 98.99% and 99.75%, respectively, whereas loss values of 0.02 and 0.01, respectively. On the other hand, during the testing phase, EfficientNetB6 and ResNet101V2 computed the best accuracy by 99.26% and 99.13%, and loss by 0.01 each, respectively.

Based on their best accuracy and loss for a testing dataset, the aforementioned models have been clubbed together to form the hybrid model and tested on the same dataset. The overall scenario shows that the values in bold are generated by the hybrid model, which indicates its highest accuracy for both training and testing datasets with 99.84% and 99.77%, respectively, with a minimum testing loss of 0.001.

Moreover, the curves generated by the models while iterating over the training and testing datasets for 15 epochs are also studied in Fig. 18. After analyzing the curves, it was discovered that the models at certain epochs reveal that the plot of training loss drops to the point of stability, as does the plot of testing loss, and has a little gap with the training loss.

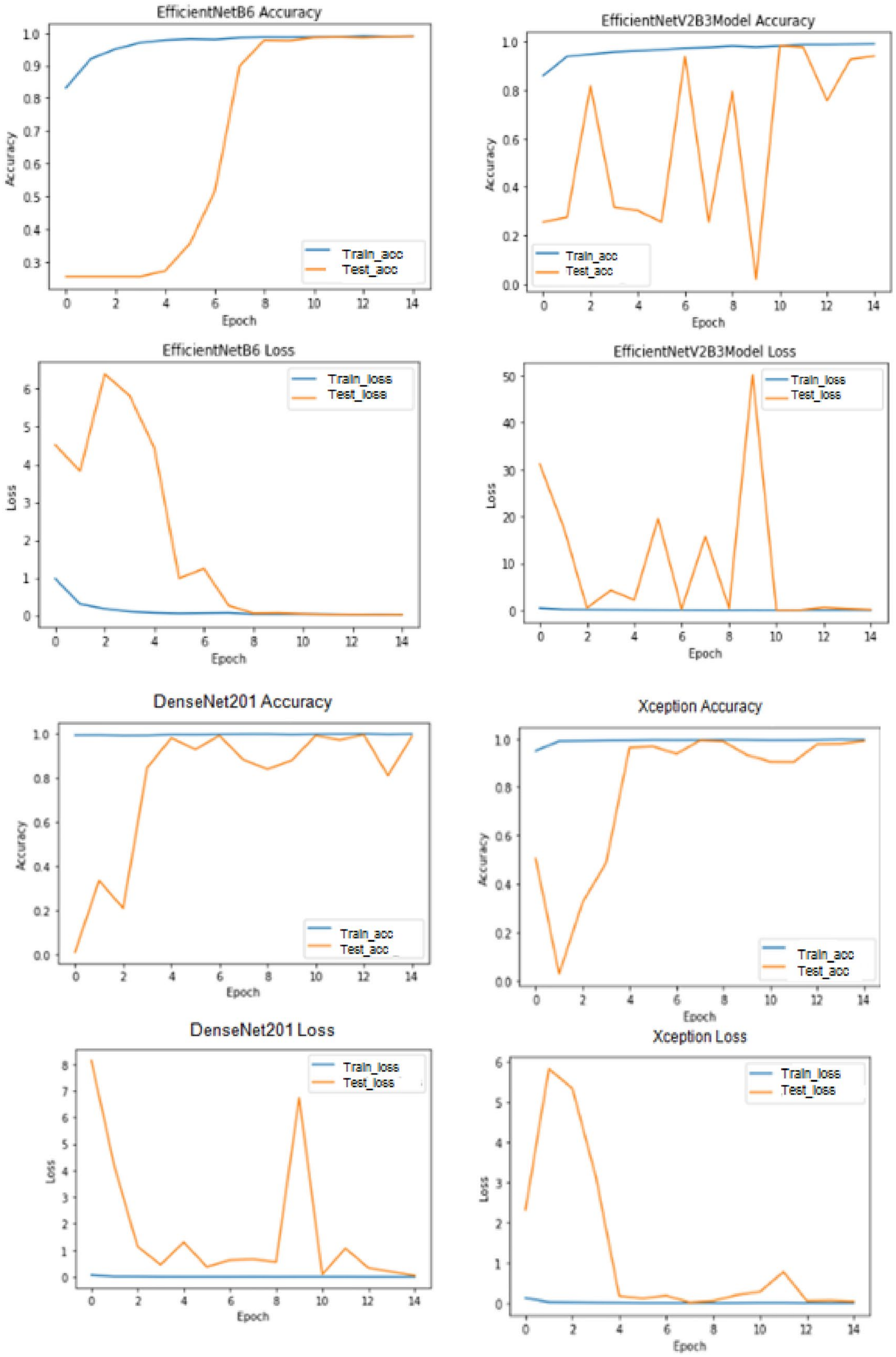


Fig. 18 Analyzing the curves of models during training and testing phase

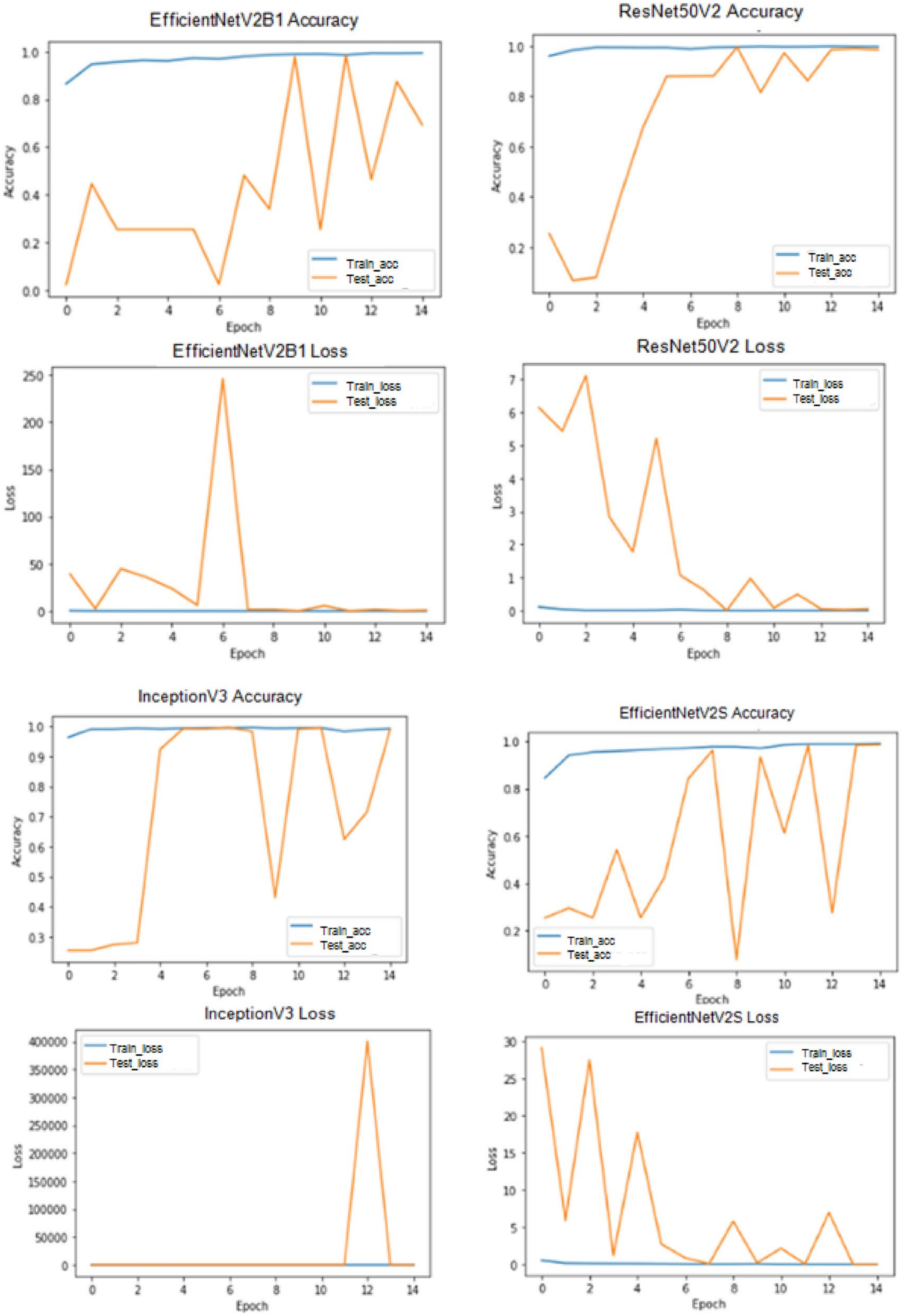


Fig. 18 (continued)

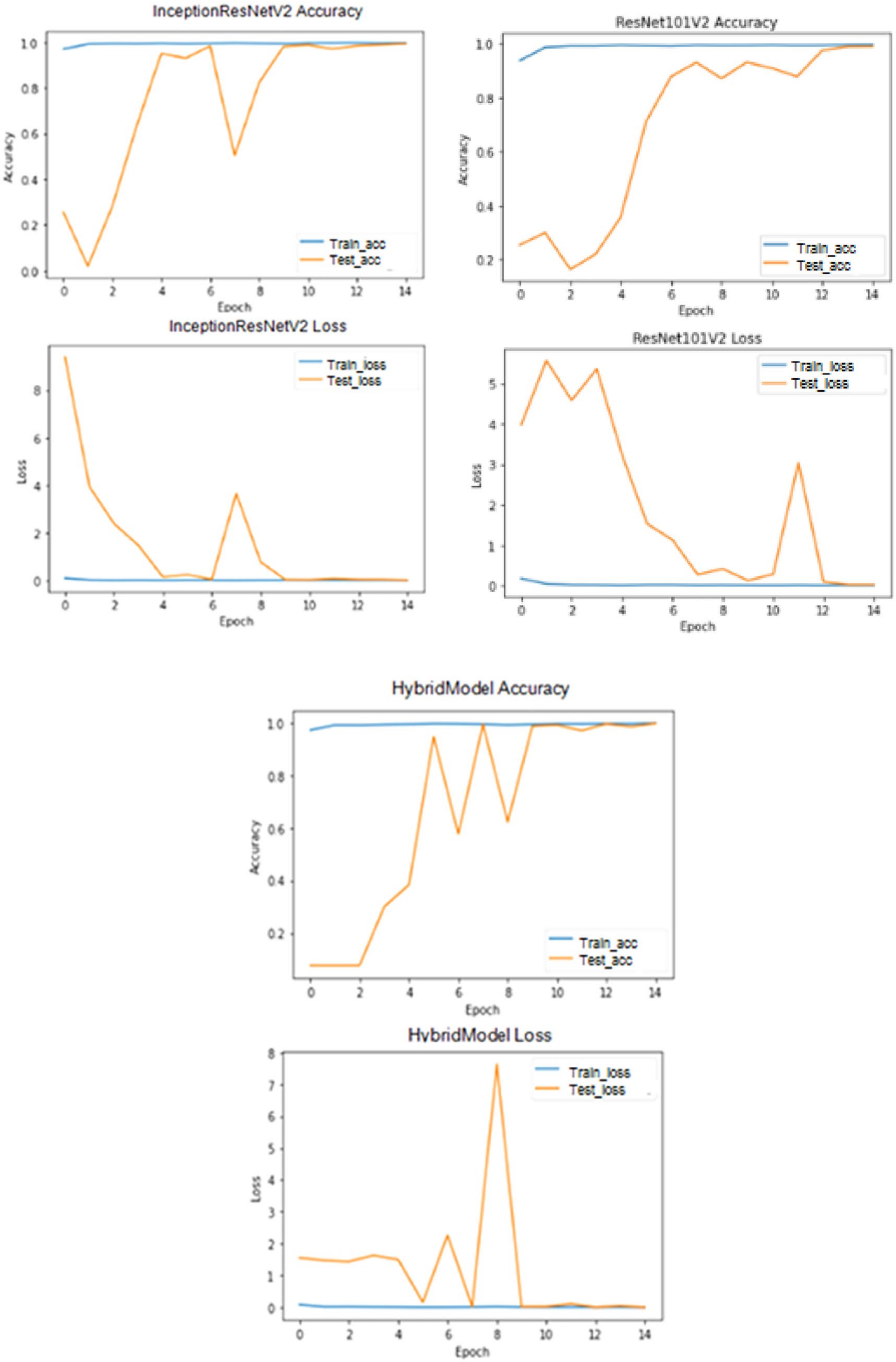


Fig. 18 (continued)

Table 6 Evaluating applied models for multi-disease detection

Algorithms	Precision	F1 score	Recall
EfficientNetB6	0.99	0.99	0.99
EfficientNetV2B3	0.88	0.89	0.91
DenseNet201	0.95	0.97	0.99
Xception	0.98	0.98	0.99
EfficienetNetV2B1	0.63	0.60	0.66
ResNetV250	0.97	0.95	0.94
Inception_V3	0.98	0.99	0.99
EfficientNetV2S	0.97	0.98	0.98
InceptionResNet-v2	0.98	0.98	0.99
ResNet101V2	0.97	0.98	0.98
Hybrid(EfficientNetB6 + ResNet101V2)	**1.00**	**1.00**	**0.99**

Similarly, it can be seen that the testing accuracy plot rises to the point of stability and has a little gap with the training accuracy. This shows that the models have good-fitting learning curves. On the other hand, at the rest of the epochs, there is a big gap between the accuracy and loss curves, which indicates that the training dataset does not provide the enough information in order to understand the problem as compared to the testing dataset during evaluation. Compared to the other remaining models, the accuracy and loss curves of the hybrid model and EffiicientNetB6 are superior.

The models are also evaluated on their performances for the parameters such as F1 score, recall, and Precision, as shown in Table 6. The proposed hybrid model (EfficientNet B6 and ResNet101V2) has generated the highest precision value with 1.00, recall with 0.99, and F1 score value. The least value was obtained by EfficientNetV2B1 by 0.63, 0.66, and 0.60, respectively which mean that EfficientNetV2B1 generated the highest number of false positives compared to true positives (Table 8).

Figure 19 shows the execution time that had been taken by the models to generate the testing accuracy. It can be found that the lowest execution time has been taken by EfficientNetB6 and Inception_V3 with 3280 s, while the highest has been taken by ResNet101V2 with 3779 s. As far as proposed hybrid model is considered, it has taken 3291 s to generate the output of testing accuracy. Besides this, if training time is considered, then all transfer models took the average time of 4 to 5 h to get trained but the proposed hybrid model took 10 h to generate the training accuracy and loss.

After training and testing the model with the airway diseases dataset, the confusion matrix, as shown in Fig. 20, has been generated for five target classes to compute their true positive as well as false positive, false negative as well as true negative values by using the formula as shown in Table 7.

Here the value of i and j is same as the label of the class. For example if it is class 0, then it's true positive will be the value at C_{00} and so on. In a nutshell all the diagonal values of the confusion matrix are the true positive of their corresponding class.

Table 8 shows that using EfficientNetB6 for class 0 (lung cancer), true positive value 430 indicates that 430 data points of positive class are successfully classified. False negative 31 indicates that 31 positive class data points are

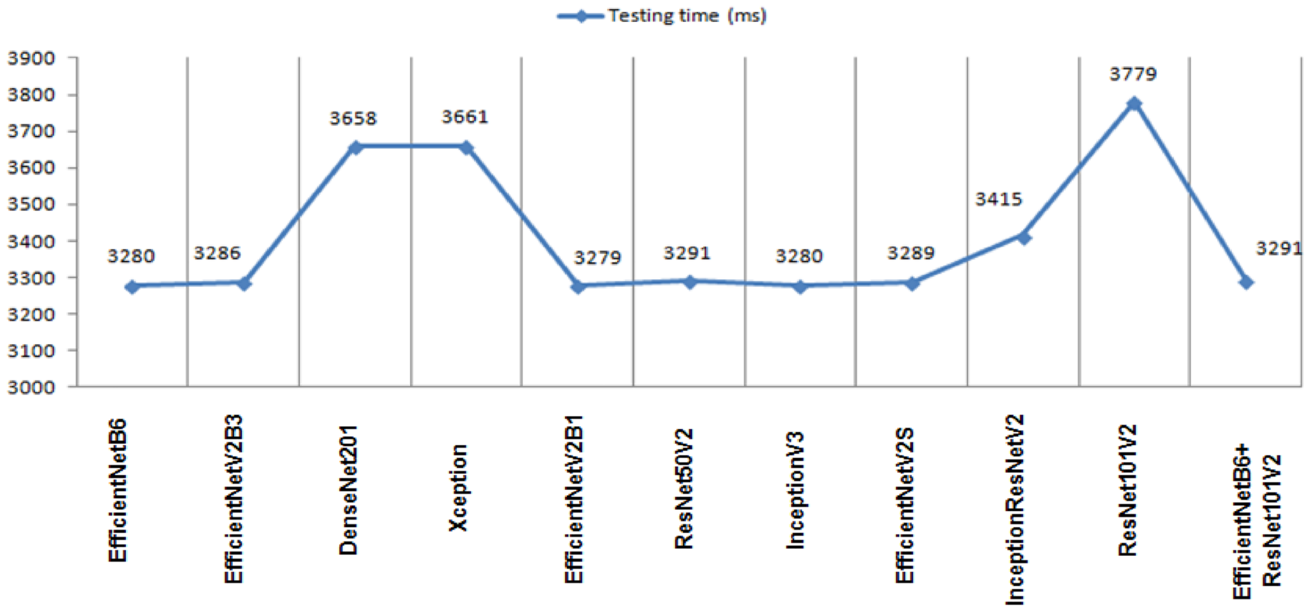
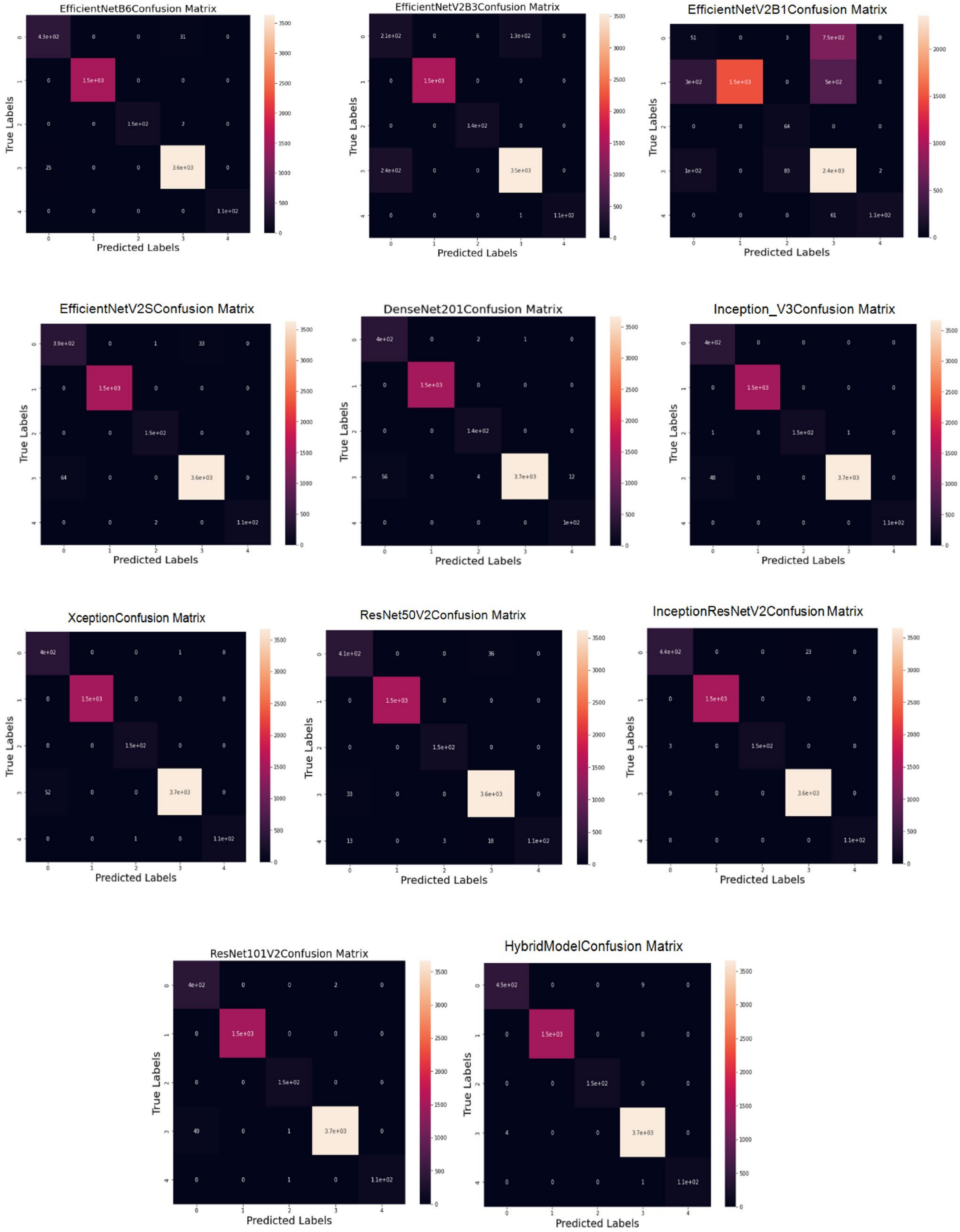


Fig. 19 Execution time of models



EfficientNetB6Confusion Matrix

	0	1	2	3	4
0	4.3e+02	0	0	31	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	2	0
3	25	0	0	3.6e+03	0
4	0	0	0	0	1.1e+02

EfficientNetV2B3Confusion Matrix

	0	1	2	3	4
0	2.1e+02	0	6	1.3e+02	0
1	0	1.5e+03	0	0	0
2	0	0	1.4e+02	0	0
3	2.4e+02	0	0	3.5e+03	0
4	0	0	0	1	1.1e+02

EfficientNetV2B1Confusion Matrix

	0	1	2	3	4
0	51	0	3	7.5e+02	0
1	3e+02	1.5e+03	0	5e+02	0
2	0	0	64	0	0
3	1e+02	0	83	2.4e+03	2
4	0	0	0	61	1.1e+02

EfficientNetV2SConfusion Matrix

	0	1	2	3	4
0	3.9e+02	0	1	33	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	0	0
3	64	0	0	3.6e+03	0
4	0	0	2	0	1.1e+02

DenseNet201Confusion Matrix

	0	1	2	3	4
0	4e+02	0	2	1	0
1	0	1.5e+03	0	0	0
2	0	0	1.4e+02	0	0
3	56	0	4	3.7e+03	12
4	0	0	0	0	1e+02

Inception_V3Confusion Matrix

	0	1	2	3	4
0	4e+02	0	0	0	0
1	0	1.5e+03	0	0	0
2	1	0	1.5e+02	1	0
3	48	0	0	3.7e+03	0
4	0	0	0	0	1.1e+02

XceptionConfusion Matrix

	0	1	2	3	4
0	4e+02	0	0	1	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	0	0
3	52	0	0	3.7e+03	0
4	0	0	1	0	1.1e+02

ResNet50V2Confusion Matrix

	0	1	2	3	4
0	4.1e+02	0	0	36	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	0	0
3	33	0	0	3.6e+03	0
4	13	0	3	18	1.1e+02

InceptionResNetV2Confusion Matrix

	0	1	2	3	4
0	4.4e+02	0	0	23	0
1	0	1.5e+03	0	0	0
2	3	0	1.5e+02	0	0
3	9	0	0	3.6e+03	0
4	0	0	0	0	1.1e+02

ResNet101V2Confusion Matrix

	0	1	2	3	4
0	4e+02	0	0	2	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	0	0
3	49	0	1	3.7e+03	0
4	0	0	1	0	1.1e+02

HybridModelConfusion Matrix

	0	1	2	3	4
0	4.5e+02	0	0	9	0
1	0	1.5e+03	0	0	0
2	0	0	1.5e+02	0	0
3	4	0	0	3.7e+03	0
4	0	0	0	1	1.1e+02

Fig. 20 Confusion matrix of models

Table 7 Formulae to compute values of confusion matrix

Class	True Positve(TP)	False Positve (FP)	False Negative (FN)	True Negative (TN)
0	C_{ij}	$\sum_{i=0}^{4} C_{i0} - TP$	$\sum_{j=0}^{4} C_{0j} - TP$	$\sum_{i,j=0}^{4} C_{ij} - (TP + FP + FN)$
1	C_{ij}	$\sum_{i=0}^{4} C_{i1} - TP$	$\sum_{j=0}^{4} C_{i1} - TP$	$\sum_{i,j=0}^{4} C_{ij} - (TP + FP + FN)$
2	C_{ij}	$\sum_{i=0}^{4} C_{i2} - TP$	$\sum_{j=0}^{4} C_{2j} - TP$	$\sum_{i,j=0}^{4} C_{ij} - (TP + FP + FN)$
3	C_{ij}	$\sum_{i=0}^{4} C_{i3} - TP$	$\sum_{j=0}^{4} C_{3j} - TP$	$\sum_{i,j=0}^{4} C_{ij} - (TP + FP + FN)$
4	C_{ij}	$\sum_{i=0}^{4} C_{i3} - TP$	$\sum_{j=0}^{4} C_{3j} - TP$	$\sum_{i,j=0}^{4} C_{ij} - (TP + FP + FN)$

Table 8 Values of TP, TN, FP, and FN for different classes of respiratory diseases

Models	Classes	True positive	False positive	False negative	True negative
EfficientNetB6	Lung cancer	430	25	31	5362
	Pul_embolism	1500	0	0	4348
	Covid-19	150	0	2	2190
	Normal	3600	33	25	2190
	Pneumoconiosis	110	0	0	5738
EfficientNetV2B3	Lung cancer	210	240	130	5250
	Pul_embolism	1500	0	0	4336
	Covid-19	140	6	0	5697
	Normal	3500	131	240	1967
	Pneumoconiosis	110	0	1	5727
EfficientNetV2B1	Lung cancer	51	400	753	5473
	Pul_embolism	1500	0	800	3624
	Covid-19	64	86	0	5774
	Normal	2400	1311	100	2028
	Pneumoconiosis	110	2	61	5751
EfficientNetV2S	Lung cancer	390	64	34	5362
	Pul_embolism	1500	0	0	4350
	Covid-19	150	1	0	5697
	Normal	3600	33	64	2003
	Pneumoconiosis	110	0	2	5738
DenseNet201	Lung cancer	400	56	3	5456
	Pul_embolism	1500	0	0	4415
	Covid-19	140	6	0	5769
	Normal	3700	1	68	2142
	Pneumoconiosis	100	12	0	5803
Inception-v3	Lung cancer	400	49	0	5461
	Pul_embolism	1500	0	0	4410
	Covid-19	150	0	2	5758
	Normal	3700	1	48	2161
	Pneumoconiosis	110	0	0	5800
Xception	Lung cancer	400	52	1	5461
	Pul_embolism	1500	0	0	4414
	Covid-19	150	1	0	5763
	Normal	3700	1	52	2162
	Pneumoconiosis	110	0	1	5803
ResNet50V2	Lung cancer	410	46	36	5417
	Pul_embolism	1500	0	0	4373
	Covid-19	150	3	0	5720
	Normal	3600	54	33	2186
	Pneumoconiosis	110	0	34	5729

Table 8 (continued)

Models	Classes	True positive	False positive	False negative	True negative
InceptionResNet-v2	Lung cancer	440	12	23	5360
	Pul_embolism	1500	0	0	4335
	Covid-19	150	0	3	5682
	Normal	3600	23	9	2203
	Pneumoconiosis	110	0	0	5725
ResNet101V2	Lung cancer	400	49	2	5462
	Pul_embolism	1500	0	0	4413
	Covid-19	150	1	0	5761
	Normal	3700	2	50	2161
	Pneumoconiosis	110	0	1	5802
Hybrid(EfficientNetB6 + ResNet101V2)	Lung cancer	450	4	9	5461
	Pul_embolism	1500	0	0	4424
	Covid-19	150	0	0	5774
	Normal	3700	10	4	2211
	Pneumoconiosis	110	0	1	5813

classified incorrectly as negative class data points, False positive 25 indicates that 25 data points of negative class are classified incorrectly as positive class data points, and true negative 5362 indicates that 5362 negative class data points are correctly classified. Similarly, we can explain the significance of TP, FN, FP, and TN values for the remaining classes and classifiers. After assaying the table completely, it has been found that the classifiers have been quite good for our dataset by obtaining greater true negative and true positive values except for EfficientNetV2B1. In the end, the system performance of the proposed hybrid (EfficientNetB6 and ResNet101V2) model has also been validated using the images taken from the dataset to predict the class of each disease and the results are shown in Fig. 21.

5 Discussion

Artificial intelligence technologies have been used to forecast the mortality rate in patients who are having airway illnesses because these diseases are one of the most common causes of mortality worldwide. In this paper, various techniques have been used to develop a system for identifying and classifying airway diseases like PE, covid-19, lung cancer, and pneumoconiosis along with normal lung images. Initially, the CLAHE technique was used for enhancing the quality and contrast of an image, followed by extracting contour features (Sects. 3.2, 3.4). These contour features were used to crop the image, which was later sent for segmentation to get a region of interest. Two thresholding techniques, Otsu /binarization and Adaptive, were applied to the image dataset to obtain the ROI efficiently (Sect. 3.4). Later ten deep pre-trained models were used such as EfficientNetB6, EfficientNetV2B3, DenseNet201, Inception-v3, Xception, EfficientNetV2B1, ResNet50V2, EfficientNetV2S, InceptionResNet-v2, and ResNet101V2 from which the two best

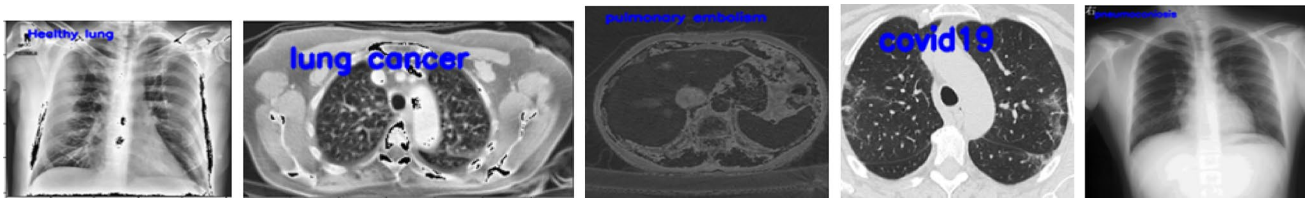


Fig. 21 Prediction of respiratory diseases using proposed hybrid model

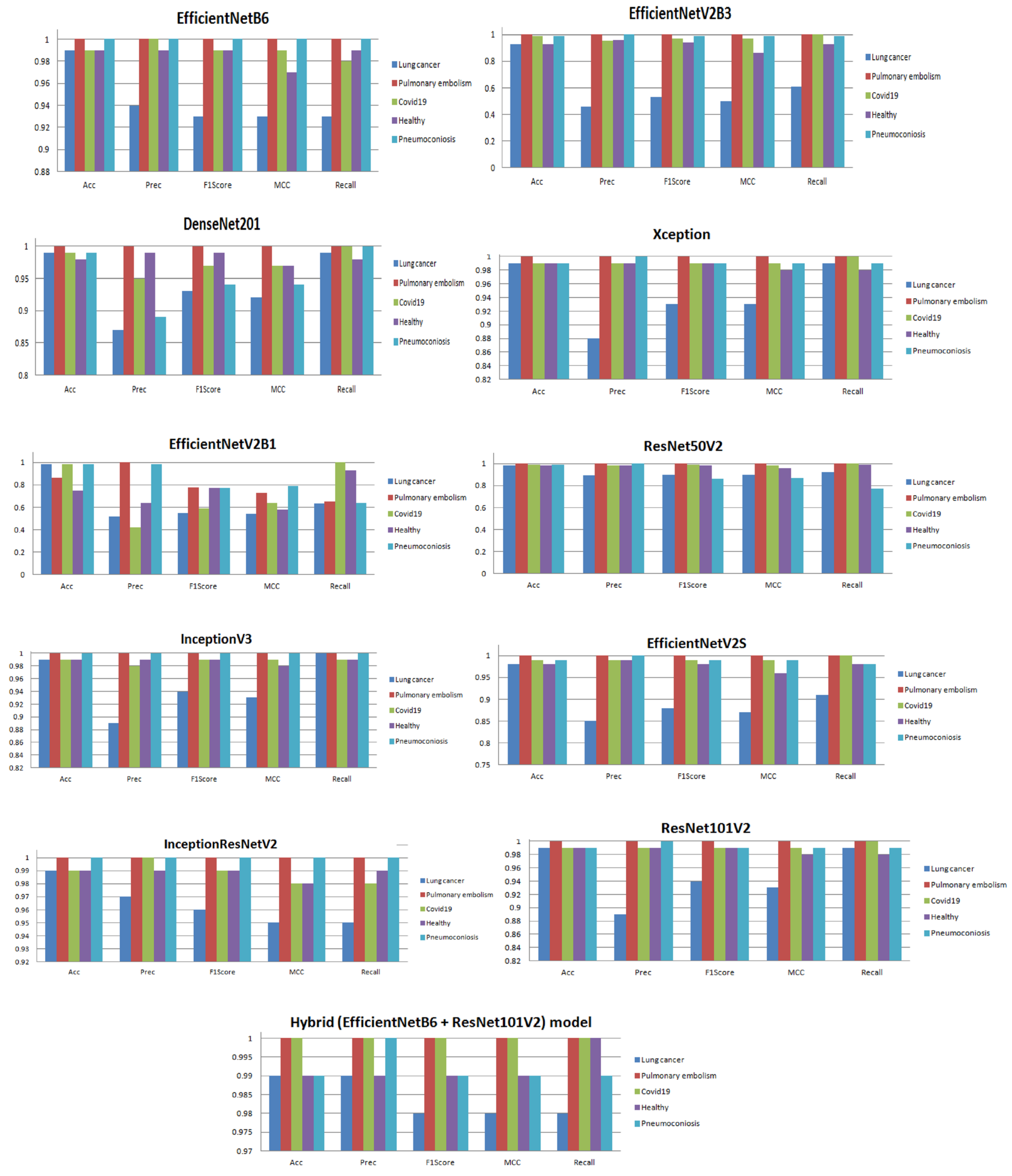


Fig. 22 Analysis of models for different performance metrics

Table 9 Comparing the existing and the current technique

References	Dataset	Techniques	Accuracy (%)
[21]	405 pneumoconiosis patients	AI techniques	97.3
[17]	Lung Image Database Consortium	Multiple Instance Learning	80.7
[20]	Chest X-ray images	ResNet34, UNet	92.46
[18]	Decathlon dataset	UNETR	97.83
[22]	Lung X-ray image	ResNet Model	95.2
Our study	19,488 images of lung cancer, pulmonary embolism, covid 19, pneumoconiosis, and healthy lungs	Proposed hybrid model (EfficientNetV2B1 + ResNet101V2)	99.77

models were hybridized and re-trained by using the same dataset. During testing, it was determined that the proposed hybrid model had the highest recall, accuracy, precision, and F1 score compared to the other models. Figure 22 depicts more assessments of the models using Recall, Precision, F1 score, and Matthew's correlation coefficient for distinct classes of respiratory diseases.

EfficientNetB6, Inception-v3, and InceptionResNet-v2 obtained the highest precision, accuracy, F1 score, recall, and MCC of 1.00 for PE and pneumoconiosis while as EfficientNetV2B3, DenseNet201, Xception ResNet50V2, EfficientNetV2S, and ResNet101V2 obtained the same values only for PE. On the other hand, EfficientNetV2B1 computed the highest accuracy of 0.98 for lung cancer, the precision of 1.00 and F1 score of 0.78 for PE, recall of 0.78 for covid 19, and MCC of 0.79 for pneumoconiosis. The proposed hybrid model (EfficientNetB6 and ResNet101V2) obtained 1.00 accuracy, recall, precision, F1 score, and MCC for PE and covid.

After obtaining all the results, the comparison has been done between the proposed hybridized method and the techniques that have been used by the researchers to predict multiple airway diseases on the basis of their accuracy metric, as mentioned in Table 9.

6 Conclusion

In this paper, ten deep transfer learning models such as EfficientNetB6, EfficientNetV2B3, DenseNet201, Inception-v3, Xception, EfficientNetV2B1, ResNet50V2, EfficientNetV2S, InceptionResNet-v2, ResNet101V2, along with the proposed hybrid model (EfficientNetB6 + ResNet101V2) had been trained using the dataset of four different respiratory diseases. It has been found that hybridizing the two models, i.e., EfficientNetB6 and ResNet101V2, obtained the highest testing accuracy of 99.77%. On the other hand, the lowest values were obtained by EfficientNetV2B1 with 69.48% accuracy and the highest loss value of 0.84. The research also has limitations, such as much computational time has been taken to pre-process the data and obtain its features. Besides this, the Otsu threshold did not work as efficiently as the Adaptive threshold for the dataset images to find the region of interest. The reason behind this is the limited flexibility of the technique, which cannot handle or generate accurate ROI for complex images with multiple regions and vary with intensity values. In addition, EfficientNetV2B1 generated the highest number of false positives, which obtained low precision of 0.63, an F1 score of 0.60, and a recall of 0.66. This shows that the model suffers from the underfitting problem, which should be taken care in the future to enhance its accuracy in prediction. It can be done by adding more layers, increasing the number of neurons in the existing layers, or using more complex model architecture. Another approach is to provide more training data or increase the number of epochs during training to allow the model to learn the underlying patterns in the data more effectively. On a large scale, researchers can also work on developing a unified platform that can detect all airway diseases in an instant in the future to minimize the time of patients and clinicians.

Funding No grant from funding has been received for this work.

Data availability Not applicable.

Declarations

Conflict of interest No conflict of interest has been declared by the authors.

References

1. D'Amato M, Molino A, Calabrese G, Cecchi L, Annesi-Maesano I, D'Amato G (2018) The impact of cold on the respiratory tract and its consequences to respiratory health. Clin Transl Allergy 8(1):1–8. https://doi.org/10.1186/s13601-018-0208-9
2. Belkacem AN, Ouhbi S, Lakas A, Benkhelifa E, Chen C (2021) End-to-end AI-based point-of- care diagnosis system for classifying respiratory illnesses and early detection of COVID-19: a theoretical framework. Front Med 8:585578. https://doi.org/10.3389/fmed.2021.585578

3. Vaishya R, Javaid M, Khan IH, Haleem A (2020) Artificial Intelligence (AI) applications for COVID-19 pandemic. Diabetes Metab Syndr 14(4):337–339. https://doi.org/10.1016/j.dsx.2020.04.012
4. Koul A, Bawa RK, Kumar Y (2022) Artificial intelligence techniques to predict the airway disorders illness: a systematic review. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-022-09818-4
5. Jacobs C, van Ginneken B (2019) Google's lung cancer AI: a promising tool that needs further validation. Nat Rev Clin Oncol 16(9):532–533. https://doi.org/10.1038/s41571-019-0248-7
6. Feng Y, Wang Y, Zeng C, Mao H (2021) Artificial intelligence and machine learning in chronic airway diseases: focus on asthma and chronic obstructive pulmonary disease. Int J Med Sci 18(13):2871. https://doi.org/10.7150/ijms.58191
7. Walsh SL, Hansell DM (2014) High-resolution CT of interstitial lung disease: a continuous evolution. Semin Respir Crit Care med 35(01):129–144. https://doi.org/10.1055/s-0033-1363458
8. Choe J, Hwang HJ, Seo JB, Lee SM, Yun J, Kim MJ et al (2022) Content-based image retrieval by using deep learning for interstitial lung disease diagnosis with chest CT. Radiology 302(1):187–197. https://doi.org/10.1148/radiol.2021204164
9. Nawaz MS, Fournier-Viger P, Shojaee A, Fujita H (2021) Using artificial intelligence techniques for COVID-19 genome analysis. Appl Intell 51(5):3086–3103. https://doi.org/10.1007/s10489-021-02193-w
10. Kumar A, Kumar N, Kuriakose J et al (2023) A review of deep learning-based approaches for detection and diagnosis of diverse classes of drugs. Arch Computat Methods Eng 30:3867–3889. https://doi.org/10.1007/s11831-023-09936-7
11. Bhardwaj P, Kumar S, Kumar Y (2023) A comprehensive analysis of deep learning-based approaches for the prediction of gastrointestinal diseases using multi-class endoscopy images. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-023-09951-8
12. Chaplot N, Pandey D, Kumar Y et al (2023) A comprehensive analysis of artificial intelligence techniques for the prediction and prognosis of genetic disorders using various gene disorders. Arch Comput Methods Eng 30:3301–3323. https://doi.org/10.1007/s11831-023-09904-1
13. Kaur I, Sandhu AK, Kumar Y (2022) Artificial intelligence techniques for predictive modeling of vector-borne diseases and its pathogens: a systematic review. Arch Comput Methods Eng 29:3741–3771. https://doi.org/10.1007/s11831-022-09724-9
14. Modi K, Singh I, Kumar Y (2023) A comprehensive analysis of artificial intelligence techniques for the prediction and prognosis of lifestyle diseases. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-023-09957-2
15. Dunke SR, Tarade SS, Waghule PB, Kolase SR (2022) Lung cancer detection using deep learning. Int J Res Publ Rev 3(5):3100–3104
16. Sori WJ, Feng J, Liu S (2019) Multi-path convolutional neural network for lung cancer detection. Multidimens Syst Signal Process 30(4):1749–1768. https://doi.org/10.1007/s11045-018-0626-9
17. Chen J, Zeng H, Zhang C, Shi Z, Dekker A, Wee L, Bermejo I (2022) Lung cancer diagnosis using deep attention-based multiple instance learning and radiomics. Med Phys 49(5):3134–3143. https://doi.org/10.1002/mp.15539
18. Said Y, Alsheikhy AA, Shawly T, Lahza H (2023) Medical images segmentation for lung cancer diagnosis based on deep learning architectures. Diagnostics 13(3):546. https://doi.org/10.3390/diagnostics13030546
19. Sun W, Wu D, Luo Y, Liu L, Zhang H, Wu S et al (2022) A fully deep learning paradigm for pneumoconiosis staging on chest radiographs. IEEE J Biomed Health Inform. https://doi.org/10.1109/JBHI.2022.3190923
20. Yang F, Tang ZR, Chen J, Tang M, Wang S, Qi W et al (2021) Pneumoconiosis computer aided diagnosis system based on X-rays and deep learning. BMC Med Imaging 21(1):1–7. https://doi.org/10.1186/s12880-021-00723-z
21. Zhang L, Rong R, Li Q, Yang DM, Yao B, Luo D et al (2021) A deep learning-based model for screening and staging pneumoconiosis. Sci Rep 11(1):1–7. https://doi.org/10.1038/s41598-020-77924-z
22. Peng S (2023) Application of medical image detection technology based on deep learning in pneumoconiosis diagnosis. Data Intell. https://doi.org/10.1162/dint_a_00228
23. Rucco M, Sousa-Rodrigues D, Merelli E, Johnson JH, Falsetti L, Nitti C, Salvi A (2015) Neural hyper-network approach for pulmonary embolism diagnosis. BMC Res Notes 8(1):1–11. https://doi.org/10.1186/s13104-015-1554-5
24. Huhtanen H, Nyman M, Mohsen T, Virkki A, Karlsson A, Hirvonen J (2022) Automated detection of pulmonary embolism from CT-angiograms using deep learning. BMC Med Imaging 22(1):1–10. https://doi.org/10.1186/s12880-022-00763-z
25. Olescki G, Clementin de Andrade JM, Escuissato DL, Oliveira LF (2022) A two step workflow for pulmonary embolism detection using deep learning and feature extraction. Comput Methods Biomech Biomed Eng Imaging Vis. https://doi.org/10.1080/21681163.2022.2060866
26. Grenier PA, Ayobi A, Quenet S, Tassy M, Marx M, Chow DS et al (2023) Deep learning-based algorithm for automatic detection of pulmonary embolism in chest CT angiograms. Diagnostics 13(7):1324. https://doi.org/10.3390/diagnostics13071324
27. Toğaçar M, Ergen B, Cömert Z (2020) COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. Comput Biol Med 121:103805. https://doi.org/10.1016/j.compbiomed.2020.103805
28. Alakus TB, Turkoglu I (2020) Comparison of deep learning approaches to predict COVID-19 infection. Chaos Solitons Fractals 140:110120. https://doi.org/10.1016/j.chaos.2020.110120
29. Bandyopadhyay SK, Dutta S, Goyel V (2020) A proposed method using deep learning from unseen to seen anxieties of children during COVID-19. Preprints, pp 2–16. https://doi.org/10.20944/preprints202009.0323.v1
30. Duong LT, Nguyen PT, Iovino L, Flammini M (2023) Automatic detection of Covid-19 from chest X-ray and lung computed tomography images using deep neural networks and transfer learning. Appl Soft Comput 132:109851. https://doi.org/10.1016/j.asoc.2022.109851
31. Aggarwal T, Furqan A, Kalra K (2015, August) Feature extraction and LDA based classification of lung nodules in chest CT scan images. In 2015 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 1189–1193. https://doi.org/10.1109/ICACCI.2015.7275773
32. Hao C, Jin N, Qiu C, Ba K, Wang X, Zhang H et al (2021) Balanced convolutional neural networks for pneumoconiosis detection. Int J Environ Res Public Health 18(17):9091. https://doi.org/10.3390/ijerph18179091
33. Soares E, Angelov P, Biaso S, Froes MH, Abe DK (2020) SARS-CoV-2 CT-scan dataset: a large dataset of real patients CT scans for SARS-CoV-2 identification. MedRxiv. https://doi.org/10.1101/2020.04.24.20078584
34. Masoudi M, Pourreza HR, Saadatmand-Tarzjan M, Eftekhari N, Zargar FS, Rad MP (2018) A new dataset of computed-tomography angiography images for computer-aided detection of pulmonary embolism. Sci Data 5(1):1–9. https://doi.org/10.1038/sdata.2018.180
35. Helen R, Kamaraj N, Selvi K, Raman VR (2011) Segmentation of pulmonary parenchyma in CT lung images based on 2D Otsu optimized by PSO. In: 2011 International conference on emerging trends in electrical and computer technology. IEEE, pp 536–541. https://doi.org/10.1109/ICETECT.2011.5760176

36. Patil MP, Ratnaparkhe VR, Kakarwal SN (2015) Adaptive thresholding for image enhancement: hardware approach. Int J Eng Res Technol 3(1):141–150
37. Agarwal V (2020) Complete architectural details of all efficientnet models. https://www.towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142
38. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M (2021) Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. J Biomol Struct Dyn 39(15):5682–5689. https://doi.org/10.1080/07391102.2020.1788642
39. Kumar Y, Gupta S (2023) Deep transfer learning approaches to predict glaucoma, cataract, choroidal neovascularization, diabetic macular edema, Drusen and healthy eyes: an experimental review. Arch Comput Methods Eng 30(1):521–541. https://doi.org/10.1007/s11831-022-09807-7
40. Tsang SH (2018) Review: Xception-with depthwise separable convolution, better than inception-v3 (image classification). Towards Data Sci. https://www.towardsdatascience.com/review-xception-with-depthwiseseparable-convolution-better-than-inception-v3-image-dc967dd42568
41. Rahimzadeh M, Attar A (2020) A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. Inform Med Unlocked 19:100360. https://doi.org/10.1016/j.imu.2020.100360
42. Sharma CM, Goyal L, Chariar VM, Sharma N (2022) Lung disease classification in CXR images using hybrid Inception-ResNet-v2 model and edge computing. J Healthc Eng. https://doi.org/10.1155/2022/9036457
43. Pandey D, Pandey K (2022, March) An extended deep learning based solution for screening COVID-19 CT-Scans. In: 2022 9th International conference on computing for sustainable global development (INDIACom). IEEE, pp 173–176. https://doi.org/10.23919/INDIACom54597.2022.9763194
44. Kaur S, Kumar Y, Koul A, Kumar Kamboj S (2022) A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: open issues and challenges. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-022-09853-1
45. Sisodia PS, Ameta GK, Kumar Y, Chaplot N (2023) A review of deep transfer learning approaches for class-wise prediction of Alzheimer's disease using MRI images. Arch Comput Methods Eng 30(4):2409–2429. https://doi.org/10.1007/s11831-022-09870-0
46. Devnath L, Luo S, Summons P, Wang D (2021) Automated detection of pneumoconiosis with multilevel deep features learned from chest X-Ray radiographs. Comput Biol Med 129:104125. https://doi.org/10.1016/j.compbiomed.2020.104125
47. Ibrahim DM, Elshennawy NM, Sarhan AM (2021) Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Comput Biol Med 132:104348. https://doi.org/10.1016/j.compbiomed.2021.104348
48. Patil S, Tiple B (2022) Deep learning framework for lung disease prognosis using X-ray image. In: Soft computing for security applications. Springer, Singapore, pp 817–830. https://doi.org/10.1007/978-981-16-5301-8_57

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.