



A Novel Artificial Intelligence System for the Prediction of Interstitial Lung Diseases

Nidhin Raju¹ · D. Peter Augustine¹ · J. Chandra¹

Received: 17 March 2023 / Accepted: 23 November 2023
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2024

Abstract

Interstitial lung disease (ILD) encompasses a spectrum of more than 200 fatal lung disorders affecting the interstitium, contributing to substantial mortality rates. The intricate process of diagnosing ILDs is compounded by their diverse symptomatology and resemblance to other pulmonary conditions. High-resolution computed tomography (HRCT) assumes the role of the primary diagnostic tool for ILD, playing a pivotal role in the medical landscape. In response, this study introduces a computational framework powered by artificial intelligence (AI) to support medical professionals in the identification and classification of ILD from HRCT images. Our dataset comprises 3045 HRCT images sourced from distinct patient cases. The proposed framework presents a novel approach to predicting ILD categories using a two-tier ensemble strategy that integrates outcomes from convolutional neural networks (CNNs), transfer learning, and machine learning (ML) models. This approach outperforms existing methods when evaluated on previously unseen data. Initially, ML models, including Logistic Regression, BayesNet, Stochastic Gradient Descent (SGD), RandomForest, and J48, are deployed to detect ILD based on statistical measures derived from HRCT images. Notably, the J48 model achieves a notable accuracy of 93.08%, with the diagnostic significance of diagonal-wise standard deviation emphasized through feature analysis. Further refinement is achieved through the application of Marker-controlled Watershed Transformation Segmentation and Morphological Masking techniques to HRCT images, elevating accuracy to 95.73% with the J48 model. The computational framework also embraces deep learning techniques, introducing three innovative CNN models that achieve test accuracies of 94.08%, 92.04%, and 93.72%. Additionally, we evaluate five full-training and transfer learning models (InceptionV3, VGG16, MobileNetV2, VGG19, and ResNet50), with the InceptionV3 model achieving peak accuracy at 78.41% for full training and 92.48% for transfer learning. In the concluding phase, a soft-voting ensemble mechanism amplifies training outcomes, yielding ensemble test accuracies of 76.56% for full-training models and 92.81% for transfer learning models. Notably, the ensemble comprising the three newly introduced CNN models attains the pinnacle of test accuracy at 97.42%. This research is poised to drive advancements in ILD diagnosis, presenting a resilient computational framework that enhances accuracy and ultimately better patient outcomes within the medical domain.

Keywords ILD · Deep learning · Transfer learning · Multi-label classification · HRCT

Introduction

A diverse group of acute and chronic bilateral parenchymal pulmonary disorders known as ILD contain some clinical characteristics, but their severity and prognosis vary significantly [1]. Idiopathic pulmonary fibrosis is one of the most common and serious interstitial lung disorders. It is

characterized by an increase in fibrosis, decreased lung function, and eventually death [2]. Patients with idiopathic pulmonary fibrosis typically die 5 years after being diagnosed [3]. Due to the need to rule out a variety of ILD, connective tissue diseases, and workplace and environmental exposures, diagnosing idiopathic pulmonary fibrosis can be challenging. Patients who are suspected of having idiopathic pulmonary fibrosis frequently undergo high-resolution CT scans, but only when the typical pattern of ILD is clearly visible can the condition be reliably identified. Between the onset of symptoms and a diagnosis of idiopathic pulmonary fibrosis, a period of 1–2 years typically passes [4]. However, in

✉ Nidhin Raju
nidhin.raju@res.christuniversity.in

¹ Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, India

order to clarify the histopathological characteristics of ILD, many patients require an invasive surgical lung biopsy. A conclusive diagnosis may still be difficult to make despite histological evaluation due to the fact that pathologists may differ regarding histopathological findings and that a good diagnosis may depend on individual experience. The diagnosis is more accurate when pulmonologists, radiologists, and pathologists work together; sadly, not all patients have access to information of this caliber. Patients are required to visit recognized competence-based regional centers for such in-depth assessments, which take time [5].

Performance on categorization tasks is crucial to the quality of medical diagnostics. Because diffuse lung disease (DLD) patterns can be seen in the lung at any cross-section, high-resolution computed tomography (HRCT) images are thought to be useful for diagnosing ILD-associated DLD patterns. Unfortunately, DLDs on HRCT images display a wide range of texture pattern meanings, making it difficult to diagnose the location of ILDs. For the proper treatment of IIPs, objective diagnosis and its quality improvement are sought, because the physician's ability to diagnose has an impact on the quality of the diagnosis. A computer-aided diagnostic (CAD) system for objective diagnosis is being developed in these decades to alleviate clinician strain. Using machine learning techniques, CAD systems are designed to provide a categorization function for a second opinion [6, 7].

By employing a variety of distinct processing layers, a subset of machine learning known as “deep learning” enables computational models to learn representations of data with multiple levels of abstraction. The most obvious difference between deep learning and traditional machine learning is that deep learning can automatically extract fully automated features and generate models that are suitable for tasks from raw data on its own, whereas traditional machine learning requires feature extraction from humans [8]. In recent years, deep learning techniques have made progress in a number of fields, including information technology and image and speech recognition. However, deep learning applications in the medical field are still in their infancy. The analysis of medical images and the associated patient electronic medical data is a perfect application for deep learning [9, 10].

The significance of ILD detection through deep learning within the industrial context rests upon its capacity to significantly enhance diagnosis efficiency and accuracy. In light of the escalating integration of medical imaging in clinical settings, an escalating demand arises for automated systems that can collaboratively aid radiologists and healthcare professionals in comprehending these intricate images. By harnessing deep learning algorithms, the capability to swiftly and meticulously process vast volumes of medical imaging data emerges, thereby mitigating the burden on healthcare practitioners and augmenting patient outcomes.

Moreover, the utilization of deep learning for ILD detection extends its influence into the realms of drug development and clinical trials. Through early identification of ILD patients, drug developers can orchestrate more streamlined and effective assessments of potential treatments. This expedites the drug development process, ultimately translating into superior treatment options for ILD patients. The AI-driven ILD detection paradigm engenders an amalgamation of advantages: elevating diagnostic precision and efficiency, diminishing the operational load on healthcare professionals, and catalyzing the pace of drug development and clinical trials. This confluence of benefits underscores the compelling potential of AI-based ILD detection to reshape the medical landscape.

Deep learning applied to the patient's medical imaging and data mining of the patient's electronic medical record (big data) should improve patient outcomes. Cloud-based applications allow the deep learning algorithm to train continuously on data sets that are not restricted to a single institution. In order to address unmet clinical requirements, numerous organizations are currently investigating applications based on deep learning. In the field of chest imaging, the creation and implementation of computer-aided detection (CAD) systems for the detection of nodules on chest radiographs and chest computed tomography (CT) has received a lot of attention [11]. Despite the fact that numerous CAD systems are in use in clinical practice, their sub-par performance (frequent false-positive and false-negative cases) has prevented widespread adoption [12]. Deep learning methods have the potential to overcome the limitations of existing CAD systems, and a number of experiments have produced encouraging outcomes. In chest imaging, disease pattern identification, diagnosis, and survival prediction have all been carried out with success using deep learning. There are still some reservations regarding its therapeutic application potential [13, 14].

The remaining sections of the article are arranged as follows. Section “[Literature Review](#)” covers a brief review of similar existing works. Section “[Methods](#)” deals with the methods that have been used in this work. The results and analysis of the experiments are presented in section “[Data-sets](#)”. And a brief discussion and conclusion of this work are demonstrated in sections “[Data Preprocessing](#)” and “[Results and analysis](#)”, respectively.

Literature Review

Classical feature extraction methods like first order grey level statistics, grey level co-occurrence matrices (GLCM), run-length matrices (RLM), and fractal analysis were provided by the earliest CAD systems for ILDs to represent 2D texture [15]. The adaptive multiple feature method

(AMFM) was created when these features were eventually combined [16]. Certain attempts have recently been made to apply deep learning DL techniques, particularly CNNs, following their outstanding performance in large-scale color image categorization [17]. The CNN nodes can learn features simultaneously while training an Artificial Neural Networks (ANN) classifier by minimizing classification error, in contrast to other feature learning techniques that produce unsupervised data representation models. The initial experiments on lung CT scans utilized shallow architectures, despite the fact that the term “deep learning” implies the utilization of numerous successive learning layers. In [18], a modified restricted Boltzmann machine (RBM) that included particular CNN features was utilized for the purpose of feature extraction and categorization of lung tissue. Weight sharing was done among the hidden neurons that were tightly connected to label (output) neurons during the supervised training of the entire network using contrastive divergence and gradient descent. A CNN was constructed entirely from scratch by the authors of [19], consisting of one convolutional layer and three dense layers. The shallow architecture of the network, on the other hand, prevents it from utilizing deep CNNs' descriptive capabilities. The pre-trained deep CNN (AlexNet) from [17] was used in [20] to classify complete lung slices after being fine-tuned using lung CT data. AlexNet was designed to identify natural color photos with an input size of 224×224 pixels; consequently, the authors were required to scale the images and artificially create three channels by employing various Hounsfield unit (HU) windows. In addition, there are concerns regarding knowledge transfer due to the significant differences between ordinary color images and medical images, and identifying complete slices may only provide a very rough estimation of the condition.

Anthimopoulos et al. [21] built and trained one of the first CNNs to categories the most prevalent ILD patterns, reaching a classification performance of 85.5% and exhibiting the DL identification potential for lung tissue idiosyncrasy. An experienced radiology team annotated 120 HRCTs by eliminating ambiguous lung areas and the bronco-vascular tree, which were then utilized to train and test the CNN. Christodoulidis et al. [22] developed a CNN architecture that can extract the textural variability of ILD patterns. Using transfer learning from multiple different non-medical source databases, they only achieved a 2% increase in the CNN performance. One of the downsides of this study was that they used CT scans instead of HRCT scans. Kim et al. [23] related shallow learning (SL) to DL for pattern classification. The authors made an effort to use relevant data from six texture benchmark databases for the current ILD pattern categorization task. According to these studies, they suggested to select a training source dataset that is comparable to the target domain and allow the network to learn some

characteristics before fine-tuning. In their investigation, they used a CNN architecture with four convolutional layers and two fully connected layers. Simply increasing the number of convolutional layers increased accuracy from 81.27 to 95.12%. Gao et al. [20] attempted a novel method for ILD pattern categorization, because they were aware of the difficulty of manually identifying region of interest (ROI) for automated pulmonary computer-aided diagnosis (CAD) systems. They demonstrated a more autonomous, grayscale-based holistic image identification system that was comparable to emphysema quantification [24].

A new method for creating an infinite number of arbitrary distinct ILD patterns from 2D HRCT images, which improved CNN's ability to classify patterns in lung tissue, was proposed by Bae et al. [25]. By providing a wide range of ILD patterns and stabilizing accuracy loss for the validation set, the program prevented overfitting. The accuracy on a specific area of interest or the entire lung was 89.5%, which was higher than the conventional CNN data augmentation rate of 82.1% and comparable to human capability. A CAD that could be easily implemented on standard computing software was created by Walsh et al. [26]. The preprocessing of 1157 HRCT images produced up to 500 distinct 4-slice montages (concatenations) per CT scan. This resulted in a multiplied image dataset consisting of 420,096 distinct montages for the training algorithm and 40,490 for the validation set. In this study, the neural network architecture was the convolutional neural network Inception-ResNet V2.

The knowledge gathered from training samples is taken into account when calculating the weights for deep CNNs, which have recently demonstrated remarkable proficiency in a wide range of tasks. “Off-the-shelf” features that can be used for categorization may be obtained by reusing the network feature extraction capacity, according to some studies [27, 28]. Through weight transferring methods that either freeze or precisely adjust the network's parameters, transfer learning has been achieved in other studies. In addition, the specificity of various layering weights was investigated in [29]. The network's first and last layers, which are closest to the input and are typically generic, are more task-specific. Consequently, the parameters of the first layer of the transfer learning process can be fixed, the parameters of subsequent layers can be fine-tuned, and the network's non-transferred weights are typically initialized at random.

Transfer learning can make up for the lack of training data and improve the outcome of target tasks by utilizing knowledge gained in the source domain. Because there are typically few medical images that can be used for instruction, the advantages of transfer learning make it a popular method in the medical field as well. Numerous researchers have attempted to employ well-known CNNs that have already been trained on ImageNet for medical image identification and classification tasks involving ultrasound, CT, and

X-ray imaging [30–32]. The domain dissimilarity variables still have an impact on the transferability of knowledge, despite the fact that these studies demonstrate the possibility of knowledge transfer from the domain of natural images to the domain of medical imaging. Several strategies to mitigate this effect have recently been the subject of research. Lu et al. [33] introduced a restriction between the source and target classifier predictions describe a novel approach to transfer learning for acquiring useful knowledge from source data. In “multi-stage transfer learning,” authors use an intermediary domain to connect the source and target domains, as described in [34, 35].

Methods

Datasets

In this proposed work, three different types of datasets to detect ILD were used. The primary dataset [36] was made up using the HRCT images. The dataset consists of 3045 HRCT images with three-dimensional annotations of diseased lung tissue areas and diagnostic criteria for pathologically verified ILD disorders. It includes 108 image series with almost 41 L of annotated lung tissue patterns, 128 individuals with 1 of 13 ILD histological disorders, and a thorough set of 99 medical data. It offers a.txt file with annotations and DICOM images. A statistical dataset was used to train ML models which was created by extracting features from the HRCT images.

Data Preprocessing

The initial slice thickness of HRCT images were less than 1.5 mm and MRI images varied from 5 to 10 mm. The spatial resolution ranged from 1.34 to 1.68 mm²/pixel. During the initial stage of preprocessing, each image was reduced to 299×299×3 dimensions. A dataset with 5600 images was created for the training set, 1200 images for the validation set, and 280 images for the testing in DICOM format for both HRCT and MRI images. These DICOM-format images are loaded in our work and afterwards transformed into a NumPy array for DL model training.

The DICOM images include information on the patient and the imaging methods employed. Such data from DICOM images makes it simpler for the model to simply extract additional features during training. Each DICOM image is a single file, and the header includes all of the necessary data to identify the file. These data are organised into four levels of hierarchy: patient, study, series, and instance. A patient is the person who is undergoing an examination. The study is an imaging procedure that is carried out at a predetermined time and day at the hospital. Each study contains

numerous series. A series can represent a patient who was physically scanned multiple times throughout a study or it might represent a patient who was physically scanned just once and the data were then reconstructed in different ways. A three-dimensional image's instance is treated as each slice of the image. A DICOM instance is the actual DICOM file.

InceptionV3

The InceptionV3 model, created by Google, consists of 10 blocks with a total of 312 layers. This model has 3 inception blocks, 13 convolutional layers, and 2 pooling layers. A number of 3×3 filters with a stride of 2 PX are present in each convolution layer. The final layer has the same number of output nodes as categories in the dataset. Each convolution block uses ReLU as the activation function and the SoftMax layer as the classification layer. The inception module seeks to behave as a multi-level feature extractor by performing 1×1, 3×3, and 5×5 convolutions within the same network module. Inception vN, where N is the version number disclosed by Google, has replaced the moniker of this architecture's initial iteration, GoogLeNet [37, 38].

MobileNetV2

The 16-layer blocks in the MobileNetV2 model have 33 filters and 1 PX as a stride in each layer of convolution. Google has also been working on this idea. The usage of a full convolutional division, which separates the convolution into a 33-depth and a 1×1-pointwise convolution, is the only difference between MobileNet and other CNNs. ReLU served as the activation function in MobileNetV2, while SoftMax was employed for categorization. Two distinct block types can be found in the MobileNetV2 model. Downsizing blocks have a stride of 2, whereas leftover blocks have a stride of 1. For these two blocks, three different types of layers are built. The first layer with non-linearity in 1×1 convolution is the ReLU6. Convolution based on depth was designed for the second layer. A 1×1 convolution with no non-linearity makes up the third layer [39, 40].

VGG16

One of the most popular pre-trained image categorization models is the VGG-16. The Visual Graphics Group at Oxford University created it. This VGG16, which has 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, was developed using ImageNet weights. It has numerous 33 filters with 1 PX as a stride on each layer of convolution. A SoftMax layer is the final layer used for categorization. Each block's activation function utilised the ReLU approach. The most notable aspect of VGG16 is that it constantly used the same padding and maxpool layer of a

22 filter with stride 2 and prioritised convolution layers of a 3×3 filter with stride 1 over several hyper-parameters. The first convolutional layer has 64 filters, the second has 128 filters, the third has 256 filters, the fourth and fifth have 512 filters [41, 42].

VGG19

A fixed size of (224×224) RGB image was given as input to this network which means that the matrix was of shape $(224, 224, 3)$. The mean RGB value of each pixel, calculated throughout the whole training set, was the only preprocessing that was carried out. They were able to cover the entirety of the image by using kernels that were (3×3) in size with a stride size of 1 pixel. To maintain the image's spatial resolution, spatial padding was applied. Stride 2 was used to conduct max pooling over a 2×2 pixel window. Rectified linear unit (ReLU) was used after this to add non-linearity to the model in order to enhance classification accuracy and computation time. As opposed to earlier models that used tanh or sigmoid functions, this one performed far better [43, 44].

ResNet50

The ResNet-50 model is divided into five stages, each comprising a convolution and an identity block. Each convolution block contains three convolution layers, as do the identity blocks. Over 23 million trainable parameters are available in the ResNet-50. A convolution with 64 distinct kernels, each with a stride of size 2, and a kernel size of 7×7 gives us 1 layer. Following that, it use a max pooling with a stride size of 2. The following convolution consists of three layers: a $1 \times 1, 64$ kernel, a $3 \times 3, 64$ kernel, and finally a $1 \times 1, 256$ kernel. These three levels are repeated a total of three times use nine layers in this phase. The kernel of $1 \times 1, 128$ is shown next, followed by the kernel of $3 \times 3, 128$ and, finally, the kernel of $1 \times 1, 512$. It use this procedure four times for a total of 12 layers. Following that, it has a kernel of size $1 \times 1, 256$, followed by two more kernels of size $3 \times 3, 256$ and size $1 \times 1, 1024$; this is repeated six times, giving a total of 18 layers. Finally, a $1 \times 1, 512$ kernel was added, followed by two more kernels of $3 \times 3, 512$ and $1 \times 1, 2048$. This process was done three times, giving a total of nine layers. Then, it do an average pool, finish it with a completely linked layer made up of 1000 nodes, and add a softmax function to produce one layer [45, 46].

Transfer Learning

Transfer learning is a sophisticated deep learning technique that entails training a CNN model on a problem that is comparable to the one being solved. It is possible to avoid having to create a new model from scratch by using this

method for feature representation from a previously trained model. It facilitates the flow of knowledge about the issue from one source to another. A pre-trained model is often trained on a large dataset like ImageNet, and the weights gained from the trained model can be used with the custom neural network to solve any other similar problem [47, 48]. Through this method, weights are reused to train a model more quickly and produce results with less generalisation error. These recently developed models can be utilised to make predictions directly on relatively untested problems or to train algorithms for related applications. Since all of the pre-trained models have already been trained on a sizable dataset, the final layer contains numerous parameters regarding the original dataset. Therefore, a new classification layer must be added in place of the pre-trained models' final layer. The degree of similarity between the source data and the target data affects how well the model performs [49, 50].

Full-Training Approach

Here, five deep learning models termed VGG16, VGG19, ResNet50, MobileNetV2 and InceptionV3 were used for classification of ILD. All models were trained with RMSprop optimizer with learning rate as 0.00001 each for 100 epochs. All layers in the models underwent full training using the ILD datasets. The image acquisition, preprocessing, and data augmentation stages in this method were the same as those in the transfer learning method. Since every model was built from scratch, the training process took longer than transfer learning. The number of layers, blocks, and input image resolution of each model varied. However, since each model was a sequential model, the order of the layers was always sequential. The ImageDataGenerator from keras.preprocessing was used to import each image and its associated label into the models. To prevent sending negative values to the following layers, the ReLU activation function was implemented for each layer. Two units of dense layers with a softmax classifier were added at the end after making all the convolution blocks. The output values from the softmax layer were 0 and 13, with 0 being a healthy case and 1 to 13 for various categories of ILD. The evaluation of each full-training model followed the same procedures as the transfer learning evaluation. Similar to how transfer learning was evaluated, each full-training model was evaluated.

ImageNet

It is a vast collection of labeled images intended for computer vision research. This dataset contains over 14 million photographs with over 21,000 classifications. The ImageNet LargeScale Visual Recognition Challenge (ILSVRC) was created specifically for image classification issues with transfer learning in a deep learning framework.

This challenge serves as a standard in image classification challenges based on transfer learning.

Soft-Voting Ensemble

Soft-voting ensemble is a technique used in ML and DL to combine the predictions of multiple models. In a soft-voting ensemble, each model in the ensemble assigns a probability to each possible class label, rather than making a hard prediction. The final prediction is then determined by taking the average or weighted average of the predicted probabilities from all the models. Once the models are trained, they are used to generate predictions on a new, unseen dataset (e.g., a validation or test set). Instead of making binary or categorical predictions, each model assigns probabilities to each possible class label. For example, if there are three classes (A, B, and C), a model might assign probabilities [0.2, 0.6, 0.2] for a given instance, indicating a higher likelihood for class B. To determine the final prediction, the predicted probabilities from all the models are combined. One common approach is to take the average of the predicted probabilities for each class across all models [51, 52]. Alternatively, if certain models are deemed more reliable or accurate, their predictions can be weighted more heavily in the ensemble. Weighted averaging involves assigning weights to each model's predicted probabilities based on their performance or confidence. Once the aggregated probabilities are obtained, the final prediction can be made by selecting the class label with the highest probability. In some cases, multiple class labels may have similar probabilities, leading to a tie. In such situations, further tie-breaking techniques can be applied, such as selecting the class label with the highest confidence from a specific model or applying a predetermined rule. The performance of the soft-voting ensemble is assessed using appropriate evaluation metrics, such as accuracy, precision, recall, or F1 score, on a separate evaluation dataset. These metrics help gauge the effectiveness of the ensemble in making accurate predictions compared to individual models or other ensemble techniques [53, 54]. Prediction is almost identical to the preceding example, but since this is a binary classification problem, utilise only class B:

Classifier 1 correctly predicts class B with a probability m .

Classifier 2 correctly predicts class A with a probability q .

Classifier 3 correctly predicts class B with a probability v .

Therefore, class B will be predicted by the ensemble model with probability $p = (m + (100 - q) + v)/3$.

Proposed Approach

The plan of our research is depicted in Fig. 1 which includes HRCT and MRI dataset acquisition, data extraction, preprocessing, data augmentation, selection of pre-trained models for transfer learning, feature extraction, classification using the VGG16, VGG19, ResNet50, InceptionV3, and MobileNetV2 models, and ensemble using soft-voting and hard-voting techniques are the steps that make up this process. The images from the converted ILD dataset were in RGB format with a pixel range of [0,255]. During the preprocessing phase, all of those photos were rescaled into the [0,1] range to meet the requirements of the pre-trained model. The classification layer of pre-trained models may not be useful for the new classification problem in transfer learning. As a result, we replaced it with a completely connected layer at the top layer of each model. Because the pre-trained network was frozen, only the weights of the top four layers and one classifier layer were changed during training. As a classification layer, a SoftMax layer was utilised on top of all models in this suggested strategy. In several experiments, Adam and RMSprop optimizers with various learning rates were used to train 100 epochs. For the process of fine-tuning all the models, we took into account a variety of factors, including the number of trainable layers, including the original and additional added layers, epochs, learning rate, and optimizers. Then, utilising statistical data, ML models such as Logistic, BayesNet, SGD, RandomForest, and J48 were employed to detect ILD. The statistical dataset was constructed by extracting and storing eighteen different features from HRCT images in a CSV file.

The soft-voting ensemble was applied on the proposed models, transfer learning model and ML models separately to achieve better accuracy. Later, we again applied second-tier soft-ensemble by utilizing the prediction value from each ensemble training which can ensure the prediction more accurately. The primary advantage of a two-tier soft-voting ensemble is the potential for improved prediction accuracy. By combining predictions from multiple models in a hierarchical manner, the ensemble can leverage the strengths of each model to make more accurate predictions. The first tier of models can capture different aspects or patterns of the data, while the second tier integrates their predictions for a final decision. This can help reduce bias and increase overall accuracy. A two-tier soft-voting ensemble can enhance the robustness of the prediction process. Since multiple models are involved, the ensemble can handle noisy or uncertain data more effectively. If one model produces erroneous predictions due to outliers or specific data biases, the influence of that model can be reduced or mitigated at the second-tier voting stage. This robustness makes the ensemble more reliable in real-world scenarios. Two-tier soft-voting ensembles encourage model diversity, which can

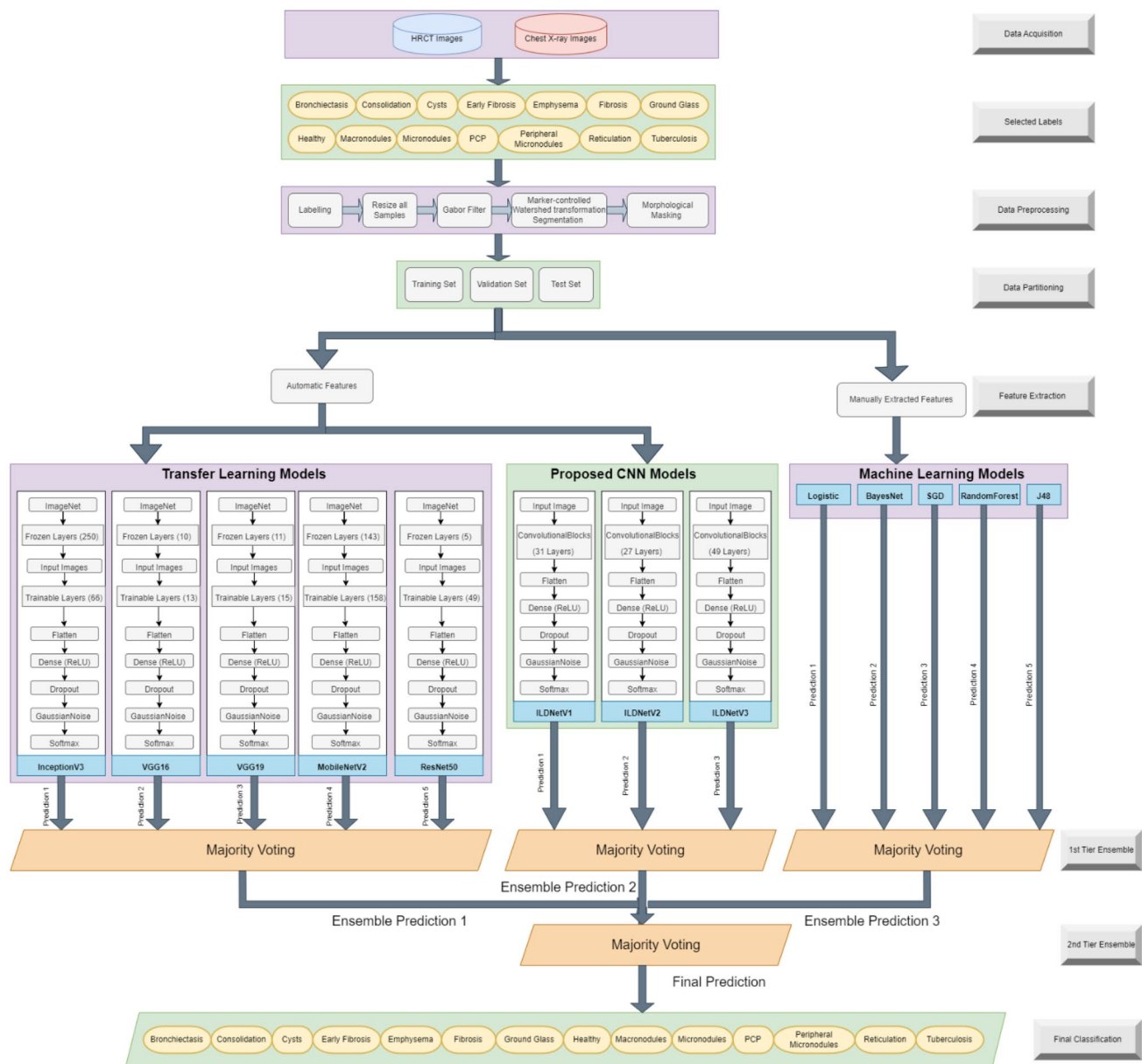


Fig. 1 The architecture diagram of the proposed work

lead to better performance. The first-tier models are typically diverse in terms of their algorithmic choices, hyperparameter settings, or input data variations. This diversity helps to capture different perspectives or representations of the underlying data, enhancing the ensemble's ability to generalize and make accurate predictions on unseen instances. Ensemble methods, including two-tier soft-voting ensembles, tend to reduce overfitting. Overfitting occurs when a model becomes too specific to the training data and fails to generalize well to new data. By combining predictions from multiple models, the ensemble can mitigate the risk of overfitting by averaging out individual model biases or errors, resulting in improved generalization performance.

Two-tier soft-voting ensembles offer flexibility and adaptability in terms of incorporating new models or replacing existing ones. If a more accurate or efficient model becomes available, it can be introduced into the ensemble at either the first or second tier. This adaptability allows the ensemble to evolve and improve over time as new models or algorithms emerge. Another advantage of two-tier soft-voting ensembles is their potential to provide more interpretable predictions. By using a hierarchical structure, the ensemble can provide insights into the decision-making process. For example, the first-tier models' predictions can be analyzed to understand which aspects or features of the data contribute more strongly to the final decision. This interpretability can

be valuable in domains where explainability is crucial, such as healthcare or finance.

CNN

The fundamental building block of any model that works with picture data is a Convolutional Neural Network. Visuals were considered when developing convolutions. There is an n -dimensional weights or filter matrix, where n is typically smaller than the image size. The input's filter size patch is used to multiply or dot product this matrix. The filter is applied sequentially to each overlapping section or filter-sized patch of the incoming data, working from left to right and then from top to bottom [55]. CNN architecture is based on convolutional layers. The convolution layers transform the image's data before passing it on to the next layer as input. The transformation is referred to as the convolutional operation. It is necessary to specify the number of filters for each convolution layer. Patterns in objects, textures, edges, forms, curves, and even colors are what these filters look for. It identifies things or patterns with deeper layers that are more intricate. An image kernel, which can be described as a tiny 3×3 or 4×4 matrix applied to the entire image, is the fundamental component of a filter.

The images' input shape are (299,299,3), because their height and width were previously specified. And the number 3 stands for the colour channel, which is represented by the fact that the images are RGB. The output size is $(299 - 3 + 1, 299 - 3 + 1) = (298, 298)$ when a First Conv2d layer Convolution operation is performed on an image of (299,299) with a kernel size of (3,3), strides and dilation are set to 1 by default, and padding is set to 'valid.' Since we defined 32 filters, the output shape is now (None,297,297,32). We obtain $((297 - 2/2) + 1, (297 - 2/2) + 1) = (148, 148)$ assuming that the input image size is (297,297), that the kernel size of the first Max Pooling layer is (2,2), and that strides are by default (2,2). The Flatten layer creates a one-dimensional vector from all of the pixels along all of the channels without taking batch size into account. The input of (7, 7, 64) is flattened to $(7 \times 7 \times 64) = 3136$ values as a result. Piecewise linear function known as the rectified linear activation function, the short-term ReLU directly outputs the input if the input is positive; if not, it will return zero. The corrected linear activation function makes it possible for models to learn more quickly and perform better by addressing the issue of vanishing gradients.

Results and Analysis

We assessed the model's performance using various accuracy metrics. The training and validation accuracies were measured to ensure that the model obtained enough knowledge during training and that overfitting was kept to a minimum. After training, the testing accuracy was estimated using 200 test images. We determined the average training and validation accuracies of the last 20 epochs for all of these metrics. Individual model outputs were subjected to soft-voting ensemble technique.

Evaluating Full-Training Models

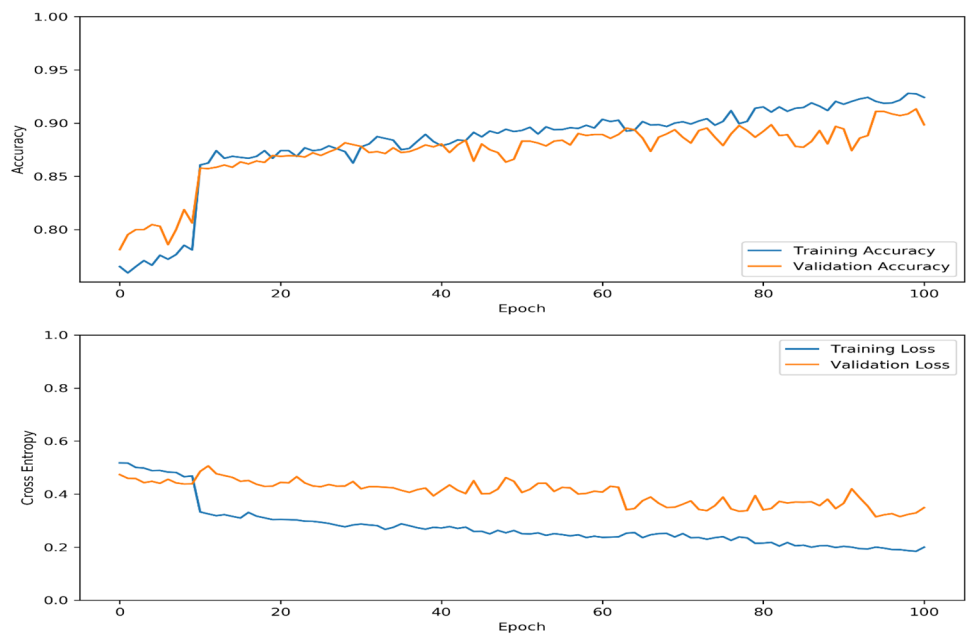
The outputs of full-training of individual models and soft-voting ensembles are presented in Table 1. In all of the experiments, the epoch accuracies grew over time. The InceptionV3 model, which underwent 30 training epochs, had the greatest test accuracy of 69.37%, while the VGG16, mobileNetV2, VGG19, and resNet50 models provided accuracy of 68.39%, 67.56%, 66.2%, and 67.42%, respectively. But at this point, the soft-voting ensemble accuracy had reached 76.56%, surpassing the test accuracy from InceptionV3. During this time, the InceptionV3 model likewise had the highest training accuracy of 72.35% and validation accuracy of 71.63%. The difference between the training and validation scores of all five models was just about 2%, ensuring that no overfitting occurred during the models' training. After 60 epochs of training, the MobileNetV2 model had the greatest training and test accuracies with 74.86% and 72.72%, respectively, whereas InceptionV3 had the highest validation accuracy with 73.44%. However, the ensemble score of 73.14% was somewhat higher than the test accuracy of the MobileNetV2 model. Among all the different models, the InceptionV3 had the best training accuracy with 87.59%, validation accuracy with 84.95%, and test accuracy with 76.11%, while VGG16 had the lowest test accuracy with 78.41% after completing training of 100 epochs. All individual models' gaps between training and validation scores were about 3% or less, ensuring that there were no overfitting issues. All three of these models were used in an ensemble with soft voting, and the accuracy was 76.56%.

The training graph of fully trained InceptionV3 model is given in Fig. 2. It includes the accuracy and loss measurements needed to assess the model's performance on both training and validation datasets. The figures clearly illustrate that no overfitting occurred during the model's training. For each epoch, the training and validation accuracies gradually improved. There was considerable volatility in both accuracies at the start of the programme, but

Table 1 The results from full-training models

Model	Trained layers	Epochs	Training Acc (%)	Validation Acc (%)	Testing Acc (%)
InceptionV3	316	30	72.35	71.63	69.37
InceptionV3	316	60	74.21	73.44	72.26
InceptionV3	316	100	87.59	84.95	78.41
VGG16	23	30	70.48	68.54	66.39
VGG16	23	60	72.34	70.83	68.37
VGG16	23	100	76.26	74.14	73.52
MobileNetV2	158	30	72.63	71.47	67.56
MobileNetV2	158	60	74.86	73.21	72.72
MobileNetV2	158	100	77.06	76.19	75.32
VGG19	26	30	69.36	67.76	66.2
VGG19	26	60	72.03	70.62	68.18
VGG19	26	100	76.38	74.28	73.86
ResNet50	55	30	70.64	67.94	67.42
ResNet50	55	60	73.26	71.52	70.08
ResNet50	55	100	76.56	74.36	74.28
Ensemble		30			70.08
Ensemble		60			73.14
Ensemble		100			76.56

Fig. 2 The training graph of fully-trained InceptionV3 model



this settled down as the training progressed. It is also seen that the loss measurements dropped steadily until the final epoch. Based on this research, we can infer that the model produced efficient outcomes with no overfitting.

Evaluating Transfer Learning Models

Table 2 lists the results of soft-voting ensembles and individual models for transfer learning. In all transfer

learning experiments, the accuracy grew with the passing of each epoch. After 30 training epochs, the InceptionV3 model had the highest test accuracy of 79.28%, while the test accuracies of the VGG16, MobileNetV2, VGG19, and ResNet50 were 76.36%, 78.52%, 76.68%, and 78.26%, respectively. The soft-voting ensemble accuracy, which was higher than the InceptionV3 test accuracy at the time, was 79.48%. The InceptionV3 model likewise had the highest validation accuracy with 81.74%, whereas

Table 2 The results from transfer learning models

Model	Layers trained	Epochs	Training Acc (%)	Validation Acc (%)	Testing Acc (%)
InceptionV3	66	30	82.05	81.74	79.28
InceptionV3	66	60	88.26	87.32	85.31
InceptionV3	66	100	94.64	94.52	92.48
VGG16	13	30	79.67	78.31	76.36
VGG16	13	60	85.08	84.13	83.59
VGG16	13	100	91.08	90.29	89.22
MobileNetV2	15	30	81.37	79.06	78.52
MobileNetV2	15	60	88.16	85.34	84.94
MobileNetV2	15	100	93.22	91.19	90.56
VGG19	15	30	79.88	78.54	76.68
VGG19	15	60	85.14	84.32	83.72
VGG19	15	100	91.36	90.41	89.56
ResNet50	10	30	82.13	79.22	78.26
ResNet50	10	60	86.57	84.44	84.08
ResNet50	10	100	91.68	90.06	89.84
Ensemble		30			79.48
Ensemble		60			85.62
Ensemble		100			92.81

ResNet50 had the highest training accuracy with 82.13%. There was no overfitting during model training as evidenced by the less than 2% variations in scores between training and validation for all five models. The InceptionV3 model once more showed the best test accuracy of 85.31% after 60 training epochs. However, the ensemble score of 85.62% was a little higher than the InceptionV3 model. At that time, the InceptionV3 showed the highest training accuracy with 88.26% and validation accuracy with 87.32%. Among all the individual models, it also

achieved the best test accuracy with 92.48%, training accuracy with 94.64%, and validation accuracy with 94.52%, whereas the least test accuracy of 89.22% for 100 epochs resulted from VGG16. The accuracy of the soft-voting ensemble of all five models was 92.81%, the highest so far. All individual model gaps between training and validation scores were less than 2%, ensuring that no overfitting concerns existed. The training graph of transfer learning MobileNetV2 model is given in Fig. 3.

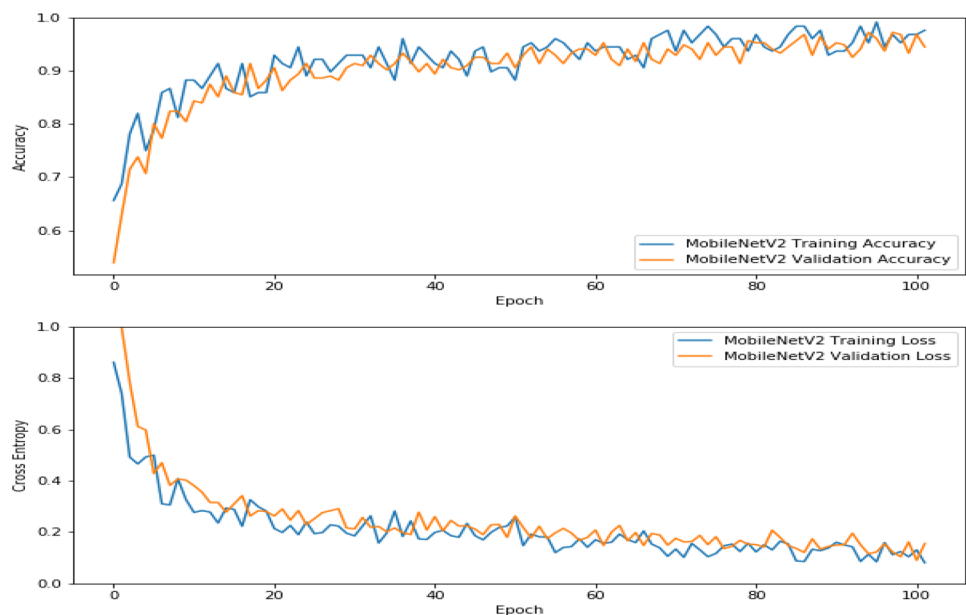
Fig. 3 The training graph of transfer learning MobileNetV2 model

Table 3 Training results of all developed CNN models using RMSprop optimizer

Proposed model	Layers trained	Epochs	Training Acc (%)	Validation Acc (%)	Testing Acc (%)
ILDNetV1	36	30	84.28	83.47	82.36
ILDNetV1	36	60	90.12	88.54	86.82
ILDNetV1	36	100	96.37	94.88	94.08
ILDNetV2	32	30	82.62	80.19	80.86
ILDNetV2	32	60	87.94	85.18	85.64
ILDNetV2	32	100	94.37	91.94	92.04
ILDNetV3	55	30	83.39	81.73	81.14
ILDNetV3	55	60	89.22	86.27	86.38
ILDNetV3	55	100	95.26	94.14	93.72
Ensemble		30			84.12
Ensemble		60			88.17
Ensemble		100			97.42

Evaluating Newly Constructed CNN Models

We developed three CNN models for ILD prediction and results from those models are demonstrated in Table 3. ILDNetV1 achieved highest training accuracy of 96.37%, validation accuracy of 94.88% and test accuracy of 94.08% after completing 100 epochs of training. The accuracy trend was same at 30th and 60th epochs break. The ILDNetV2 and ILDNetV3 also performed well and reached test accuracy of 92.04% and 93.72% after 100 epochs of training. We applied soft-voting ensemble to achieve better accuracy since all models were achieved good results without any over-fitting. The ensemble test accuracies were higher than all three developed CNNs with test accuracy of 84.12% for 30 epochs, 88.17% for 60 epochs and 97.42% for 100 epochs. The training graph of developed CNN ILDNetV1 is given in Fig. 4.

In subsequent experiments, alternative optimizers, including Adam, Adagrad, and Adadelta, were employed following the same methodology as the previous employment of the RMSprop optimizer. Employing these optimizers, the newly devised trio of CNN models underwent training for a total of 100 epochs. The outcomes of these training experiments are presented in Table 4. Despite the notable performance of the Adam optimizer, achieving an ensemble accuracy of 94.86%, it fell short of matching the level set by the RMSprop optimizer. The utilization of the Adagrad optimizer led to discernible accuracy discrepancies between training and validation scores. The ensemble test accuracy attained using Adagrad reached a modest 92.76%. Although the disparity between training and validation accuracies observed with the Adadelta optimizer was less than with Adagrad, the resultant accuracy metrics remained below expectations. After a hundred epochs of training, the

Fig. 4 The training graph of developed CNN ILDNetV1 using RMSprop optimizer

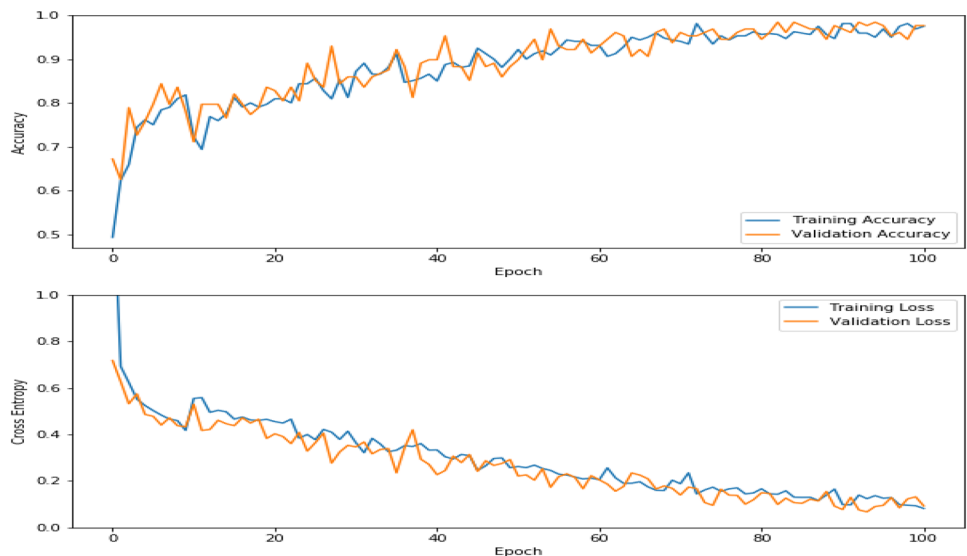


Table 4 Training results of all developed CNN models using other optimizers

Proposed model	Layers trained	Optimizer	Training Acc (%)	Validation Acc (%)	Testing Acc (%)
ILDNetV1	36	Adam	94.14	92.08	92.46
ILDNetV2	32	Adam	92.62	89.78	90.16
ILDNetV3	55	Adam	93.15	92.26	92.54
Ensemble		Adam			94.86
ILDNetV1	36	Adagrad	95.44	91.24	91.72
ILDNetV2	32	Adagrad	94.67	88.78	89.16
ILDNetV3	55	Adagrad	95.46	92.32	91.96
Ensemble		Adagrad			92.76
ILDNetV1	36	Adadelta	92.37	91.26	90.70
ILDNetV2	32	Adadelta	90.79	89.18	89.22
ILDNetV3	55	Adadelta	91.64	90.38	90.16
Ensemble		Adadelta			91.84

Adadelta optimizer yielded an ensemble accuracy of merely 91.84%. Through these conducted experiments, it is evident that the RMSprop optimizer exhibited superior performance compared to the Adam, Adagrad, and Adadelta optimizers. In our modeling process, we implemented the EarlyStopping method in Python, which consistently terminated training for all models upon reaching 100 epochs. This strategic utilization of the EarlyStopping technique effectively mitigated issues related to overfitting. The rationale behind this decision lies in the observation that, following the 100th epoch, there was limited discernible improvement in the validation accuracy over the subsequent six epochs. Consequently, we adopted a standardized training protocol of limiting all models to a maximum of 100 epochs.

Evaluating ML Classifiers

Then we used ML classifiers such as Logistic, BayesNet, SGD, RandomForest and J48 for detecting ILD using statistical data. The statistical dataset was created by extracting eighteen different features from HRCT images and storing it in CSV file. We considered five ILD combinations to train the ML classifiers. The combinations were created using five ILD cases such as Consolidation, Fibrosis, Groundglass,

Micronodules and Reticulation with Healthy cases. The J48 showed highest accuracy with all combinations (Table 5). Among all the combinations, J48 achieved 93.08% accuracy with C1 combination which was the highest accuracy from ML models. The other accuracy measurements of this model is given in Fig. 5.

Our next target was to find out which features contribute more in accuracy. Therefore, we considered the highest accuracy providing J48 with each features separately for training. The training results of J48 with each feature are given in Table 6. It shows that all features are contributing almost equal level in accuracy even though diagonal:SD (standard deviation) was the highest. The highest accuracy of 85.20% was obtained from Healthy_Micronodules combination for diagonal:SD feature.

In the next phase, the Gabor filter was applied on the HRCT and trained all ML models in the same manner. Gabor filters are used to analyze textures in images and are widely used in computer vision and image processing tasks. The image were converted into the binary format with pixel values either 0 or 1. The threshold value is set to 127. The J48 showed highest accuracy this time too. It achieved 95.55% accuracy with C1 combination which was the highest accuracy from ML models (Table 7).

Table 5 Training results of all used ML classifiers

ML Classifiers	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Logistic	84.46	50.79	52.50	63.61	78.39
BayesNet	81.46	81.56	83.76	86.69	76.68
SGD	87.60	75.09	70.57	85.35	78.75
RandomForest	84.99	81.20	82.66	87.30	81.69
J48	93.08	83.88	85.84	92.19	88.40

C1: Healthy_Consolidation C2: Healthy_Fibrosis C3: Healthy_GroundGlass C4: Healthy_Micronodules C5: Healthy_Reticulation

Fig. 5 Measurement scores from developed CNN ILD-NetV1

Correctly Classified Instances	713	93.0809 %							
Incorrectly Classified Instances	53	6.9191 %							
Kappa statistic	0.7803								
Mean absolute error	0.0956								
Root mean squared error	0.2502								
Relative absolute error	28.2475 %								
Root relative squared error	60.8472 %								
Total Number of Instances	766								
=== Detailed Accuracy By Class ===									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.745	0.018	0.918	0.745	0.823	0.787	0.912	0.846	consolidation
	0.982	0.255	0.934	0.982	0.957	0.787	0.912	0.957	Healthy
Weighted Avg.	0.931	0.204	0.930	0.931	0.928	0.787	0.912	0.933	
=== Confusion Matrix ===									
a b <-- classified as									
123	42	a = consolidation							
11	590	b = Healthy							

Table 6 Training results of J48 for each feature

Feature	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Sum of 1	81.20	80.83	83.52	84.86	73.38
Sum of 0	78.72	80.83	83.39	84.74	73.26
Mean	78.46	80.83	83.15	85.10	73.26
Column:SD	80.94	76.19	77.53	74.97	76.31
Row:SD	78.46	79.61	80.71	82.42	73.14
Diagonal:SD	84.20	81.05	83.92	85.20	78.51
FFT:SD	81.20	80.83	83.52	84.86	73.38
DCT:SD	78.46	80.83	83.27	84.86	73.26
MSER_Mean	78.20	79.49	81.81	81.81	74.85
MSER_SD	78.46	79.24	81.56	82.30	75.21
BRISK_Features	78.72	79.12	81.20	82.05	75.46
FAST_Features	78.46	79.12	81.20	81.56	74.48
Harris	79.50	79.61	81.81	82.78	72.28
Kaze_Mean	78.46	79.49	82.42	83.15	73.26
Kaze_Std	78.46	79.61	82.54	83.27	73.26
MinEigen	78.98	79.61	82.30	82.78	72.89
Surf_Mean	78.46	75.95	75.95	76.68	73.26
Surf_Std	78.46	77.05	76.80	77.66	73.26

In the next step, the highest accuracy providing J48 was considered again with each features separately for training. The training results of J48 model with each feature are given in Table 8. It demonstrates that all features contribute almost equally to accuracy: The highest diagonal:SD was found. The Healthy_Micronodules combination achieved the highest diagonal accuracy of 86.43% diagonal:SD feature.

Evaluating ML Classifiers After Applying Segmentation and Masking

The marker-controlled watershed transformation segmentation technique was applied on the images to achieve better results (Table 9). Marker-controlled watershed transformation is a segmentation technique used in image processing and computer vision. It is based on the concept of

Table 7 Training results of all used ML classifiers after applying Gabor filter

ML classifiers	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Logistic	86.70	51.48	54.10	65.40	80.95
BayesNet	83.62	82.66	86.31	89.13	79.19
SGD	89.92	76.10	72.72	87.75	81.33
RandomForest	87.24	82.30	85.17	89.75	84.36
J48	95.55	85.01	88.45	94.78	91.29

Table 8 Training results of J48 for each feature

Feature	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Sum of 1	83.35	81.92	86.06	87.24	75.78
Sum of 0	80.81	81.92	85.93	87.12	75.66
Mean	80.54	81.92	85.68	87.49	75.66
Column:SD	83.08	77.22	79.89	77.08	78.81
Row:SD	80.54	80.68	83.16	84.74	75.53
Diagonal:SD	86.43	82.14	86.47	87.59	81.08
FFT:SD	83.35	81.92	86.06	87.24	75.78
DCT:SD	80.54	81.92	85.80	87.24	75.66
MSER_Mean	80.27	80.56	84.30	84.11	77.30
MSER_SD	80.54	80.31	84.04	84.61	77.67
BRISK_Features	80.81	80.19	83.67	84.36	77.93
FAST_Features	80.54	80.19	83.67	83.85	76.92
Harris	81.61	80.68	84.30	85.11	74.64
Kaze_Mean	80.54	80.56	84.93	85.49	75.66
Kaze_Std	80.54	80.68	85.05	85.61	75.66
MinEigen	81.07	80.68	84.80	85.11	75.27
Surf_Mean	80.54	76.98	78.26	78.83	75.66
Surf_Std	80.54	78.09	79.13	79.84	75.77

watersheds, which are regions in an image that are separated by watershed boundaries. In marker-controlled watershed transformation, markers or seed points are manually or automatically placed in the image to guide the segmentation process. These markers indicate the regions of interest or objects that need to be segmented. The watershed transformation algorithm then treats the image as a topographic surface, where the intensity values represent the elevation. The algorithm floods the image from the markers, and as the water level rises, it forms basins around the markers. The boundaries between these basins are the watershed boundaries, which represent the segmentation result. The marker-controlled watershed transformation is particularly useful when dealing with complex images that have overlapping or touching objects. By providing accurate markers, the algorithm can accurately segment the objects of interest.

The morphological masking approach was utilized to create a mask for the images of the dataset as shown in

Table 10. Morphological masking is a technique used in image processing and computer vision to extract or enhance specific regions or features in an image using morphological operations. Morphological operations are mathematical operations that manipulate the shape and structure of objects in an image. The two most commonly used morphological operations are erosion and dilation. In morphological masking, a binary mask, also known as a structuring element, is defined to represent the desired region or feature to be extracted or enhanced. The structuring element is a small binary image that defines the shape and size of the neighborhood around each pixel. To apply morphological masking, the structuring element is scanned over the image, pixel by pixel. At each pixel location, the corresponding pixels in the image and the structuring element are compared. If the pixels match according to a specific condition, the pixel at that location is preserved or modified based on the desired operation.

Table 9 Marker-controlled watershed transformation segmentation on HRCT images




Marker	Sample image
Internal marker	
External marker	
Watershed marker	

Table 10 Morphological masking on HRCT images

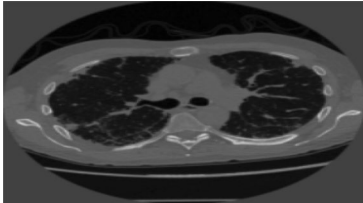
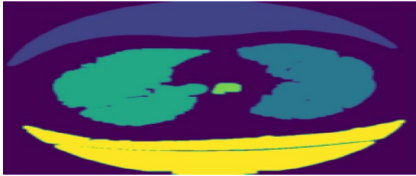



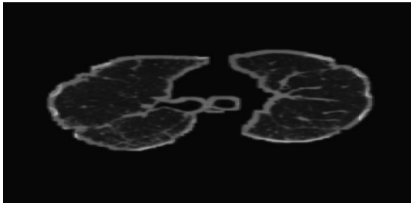
Image type	Sample image	Image type	Sample image
Original		Color labels	
Threshold		Final mask	
After erosion and dilation		Apply mask on original	

Table 11 Training results of all used ML classifiers after applying segmentation and masking

ML classifiers	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Logistic	86.87	51.58	54.21	65.56	81.11
BayesNet	83.78	82.82	86.49	89.34	79.34
SGD	90.10	76.25	72.87	87.96	81.48
RandomForest	87.41	82.46	85.35	89.97	84.52
J48	95.73	85.18	88.64	95.01	91.47

The same five ML models were again used to assess the performance of these segmentation and masking techniques. The J48 model showed highest accuracy this time too. It achieved 95.73% accuracy with C1 combination which was the highest accuracy from ML models (Table 11).

In the following step, the ML model with the highest accuracy, the J48 model, was looked at once more, with each feature used separately for training. Table 12 displays the J48 model's training results for each feature. It exhibits that all highlights contribute similarly to precision: The most extreme diagonal: SD was found. With an accuracy of 86.60%, the Healthy_Micronodules combination had the best diagonal accuracy: SD include.

Assessing Performance of the Proposed Two-Tier Ensemble Framework Using the Untrained Images

To check the efficiency of our proposed work for classifying and categorizing ILD diseases, we applied some untrained HRCT images of various ILD category into the trained model. The proposed approach was able to predict ILD category accurately with high similarity score. The predictions were there happened by considering the majority voting ensemble mechanism with predictions from our own three newly developed CNN models, five pre-trained deep learning models and five ML models. The predictions and their merging from the various models ensure the sufficiency of prediction results and capacity and efficiency for our system's prediction ability. Some samples of such prediction results are demonstrated in Figs. 6 and 7.

The given results represent the performance of our proposed two-tier ensemble prediction approaches for

Table 12 Training results of J48 model for each feature after applying segmentation and masking

Feature	Accuracy from C1 (%)	Accuracy from C2 (%)	Accuracy from C3 (%)	Accuracy from C4 (%)	Accuracy from C5 (%)
Sum of 1	83.51	82.08	86.23	87.48	75.93
Sum of 0	80.96	82.08	86.09	87.36	75.80
Mean	80.70	82.08	85.84	87.73	75.80
Column:SD	83.25	77.37	80.04	77.29	78.96
Row:SD	80.70	80.84	83.33	84.97	75.68
diagonal:SD	86.60	82.31	86.64	87.83	81.23
FFT:SD	83.51	82.08	86.23	87.48	75.93
DCT:SD	80.70	82.08	85.97	87.48	75.80
MSER_Mean	80.43	80.72	84.46	84.34	77.45
MSER_SD	80.70	80.47	84.20	84.84	77.82
BRISK_Features	80.96	80.35	83.83	84.59	78.08
FAST_Features	80.70	80.35	83.83	84.08	77.06
Harris	81.77	80.84	84.46	85.34	74.79
Kaze_Mean	80.70	80.72	85.09	85.72	75.80
Kaze_Std	80.70	80.84	85.21	85.84	75.80
MinEigen	81.23	80.84	84.97	85.34	75.42
Surf_Mean	80.70	77.13	78.41	79.05	75.80
Surf_Std	80.70	78.24	79.29	80.06	75.80

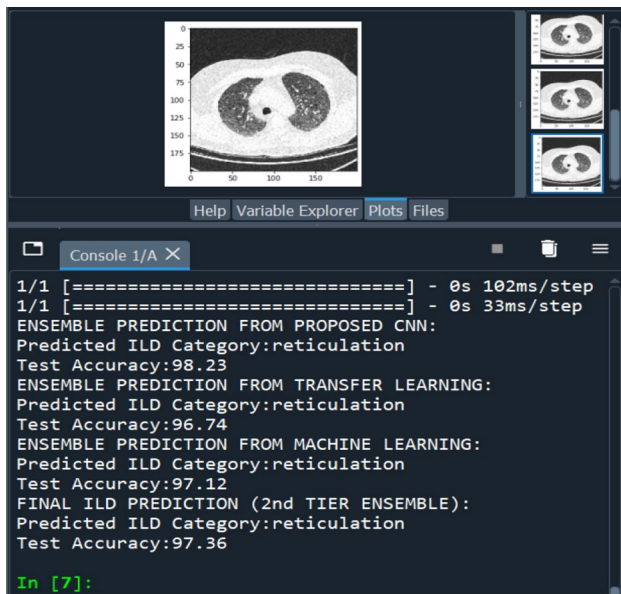


Fig. 6 Model prediction for first sample image

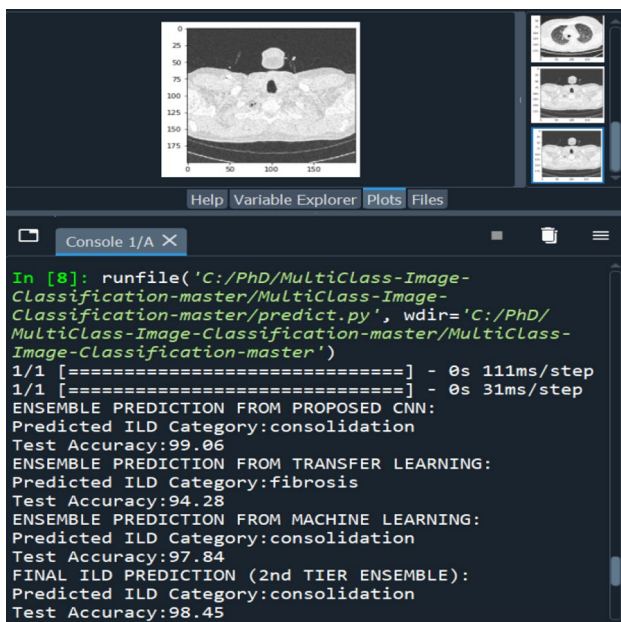


Fig. 7 Model prediction for second sample image

classifying ILD using two different untrained HRCT Images. It combines the predictions of multiple base models or classifiers to make a final prediction. According to Fig. 5, the ensemble prediction from proposed CNN models achieves a test accuracy of 98.23% and predicts the ILD category as “reticulation”. In the transfer learning ensemble approach, the test accuracy is slightly lower than the proposed CNN ensemble, at 96.74%. It also predicts the ILD category as “reticulation”. The Ensemble Prediction from Machine

Learning achieves a test accuracy of 97.12% and, like the previous two, predicts the ILD category as “reticulation”. This is the final ensemble prediction that combines the outputs of the previous three ensembles or models. It achieves a test accuracy of 97.36% and predicts the ILD category as “reticulation”. All three ensemble approaches (Proposed CNNs, Transfer Learning, and Machine Learning) perform well in classifying ILD categories, with high test accuracies ranging from 96.74 to 98.23%. The final 2nd Tier Ensemble maintains a high accuracy of 97.36%. Since it is a soft-voting ensemble approach, it predicts the ILD category as “reticulation” with all three votes, because all three first-tier ensemble approaches also predict the same ILD category.

According to Fig. 6, the result suggests that the proposed CNN models achieved an ensemble test accuracy of 99.06% when predicting the ILD category, and it classified the ILD as “consolidation”. The transfer learning ensemble approach achieved a test accuracy of 94.28%. The predicted ILD category was “fibrosis”. The ensemble prediction approach which used machine learning techniques, achieved a test accuracy of 97.84%. It classified the ILD as “consolidation”. The final prediction appears to be a combination of the predictions from the three previous ensemble methods in a second-tier ensemble approach. The resulting predicted ILD category is “consolidation”, and the test accuracy is 98.45%. The reason for predicting the category as “consolidation” is that it achieved most votes (from proposed CNN and Machine learning ensemble), only ensemble of transfer learning predicted a different category. It seems that the ensemble prediction method using multiple techniques has achieved high test accuracy for ILD category prediction. This approach likely aims to increase prediction accuracy by leveraging the strengths of various models or techniques.

Conclusion

In this proposed work, a new deep learning approach was proposed to detect ILD using HRCT images. In the first stage, three CNN models were developed and also used five full trained and five pre-trained models on the ImageNet dataset such as VGG16, VGG19, ResNet50, MobileNetV2, and InceptionV3 to detect ILD from HRCT images. Our proposed first model achieved highest individual model test accuracy with 94.08% than all other deep learning models. In each experiment, the RMSprop optimizers were used to train the models for 100 epochs using 25,000 augmented images generated from 2000 original images. In the second stage, a new algorithm was developed to extract various features from HRCT images and applied those features into five ML algorithms such as Logistic, BayesNet, SGD, RandomForest and J48. The J48 model showed better accuracy among ML models which reached highest accuracy of

93.08%. The diagonal-wise standard deviation feature was contributing more on models' training by analyzing each features separately with J48 model. The ensemble of models helped to achieve better results. Therefore, the majority voting mechanism with each kind of models separately was applied in the third stage. We achieved highest ensemble test accuracy of 97.42% after ensemble of all our three newly built CNN models. In the final stage, the two-tier ensemble concept was applied. The outputs of these ensemble models were fed into majority voting ensemble once again which helped to ensure the reliability of the model's prediction on unseen images. Our results suggest that the proposed approach can be used to improve ILD detection accuracy compared to other deep learning methods, which may assist the doctors. When compared with the previous state-of-the-art methods, our approach achieved better results.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s42979-023-02524-3>.

Funding No funding received.

Data Availability The data that support the findings of this study are available from St. John's Medical College, Bengaluru, India but restrictions apply to the availability of these data, which were used under licence for the current study, and so are not publicly available. Data are, however, available from the authors upon reasonable request and with permission of St. John's Medical College, Bengaluru, India.

Declarations

Conflict of Interest No conflict of interest.

References

- Hodnett PA, Naidich DP. Fibrosing interstitial lung disease: a practical high-resolution computed tomography-based approach to diagnosis and management and a review of the literature. *Am J Respir Crit Care Med*. 2013;188(2):141–9. https://doi.org/10.1164/RCCM.201208-1544CI/SUPPL_FILE/DISCLOSURES.PDF.
- Wells AU. The revised ATS/ERS/JRS/ALAT diagnostic criteria for idiopathic pulmonary fibrosis (IPF)—practical implications. *Respir Res*. 2023;14(Suppl 1):1–6. <https://doi.org/10.1186/1465-9921-14-S1-S2/TABLES/4>.
- Fernández Pérez ER, et al. Incidence, prevalence, and clinical course of idiopathic pulmonary fibrosis: a population-based study. *Chest*. 2010;137(1):129–37. <https://doi.org/10.1378/CHEST.09-1002>.
- Collard HR, King TE, Bartelson BB, Vourlekis JS, Schwarz MI, Brown KK. Changes in clinical and physiologic variables predict survival in idiopathic pulmonary fibrosis. *Am J Respir Crit Care Med*. 2012;168(5):538–42. <https://doi.org/10.1164/RCCM.200211-1311OC>.
- Flaherty KR, et al. Idiopathic Interstitial Pneumonia. *Am J Respir Crit Care Med*. 2012;170(8):904–10. <https://doi.org/10.1164/RCCM.200402-1470C>.
- Xu R, Hirano Y, Tachibana R, Kido S. Classification of diffuse lung disease patterns on high-resolution computed tomography by a bag of words approach. *Lect Notes Comput Sci*. 2011;6893(3):183–90. https://doi.org/10.1007/978-3-642-23626-6_23/COVER.
- Gangeh MJ, Sørensen L, Shaker SB, Kamel MS, De Bruijne M, Loog M. A texton-based approach for the classification of lung parenchyma in CT images. *Lect Notes Comput Sci*. 2010;6363(3):595–602. https://doi.org/10.1007/978-3-642-15711-0_74/COVER.
- Pradeep IK, Jaya Bhaskar M, Satyanarayana B. Data science and deep learning applications in the e-commerce industry: a survey. *Indian J Comput Sci Eng*. 2020;11(5):497–509.
- Sivachandiran S, Jagan Mohan K, Mohammed Nazer G. Intelligent deep learning enabled crowd detection and classification model in real time surveillance videos.
- Bejnordi BE, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*. 2017;318(22):2199–210. <https://doi.org/10.1001/JAMA.2017.14585>.
- McAdams HP, Samei E, Dobbins J, Tourassi GD, Ravin CE. Recent advances in chest radiography. *Radiology*. 2006. <https://doi.org/10.1148/radiol.2413051535>.
- Cicero M, et al. Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs. *Investig Radiol*. 2017;52(5):281–7. <https://doi.org/10.1097/RLI.00000000000000341>.
- Lakhani P, Sundaram B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*. 2017;284(2):574–82. <https://doi.org/10.1148/RADIOL.2017162326>.
- Gonzalez G, et al. Disease staging and prognosis in smokers using deep learning in chest computed tomography. *Am J Respir Crit Care Med*. 2018;197(2):193–203. https://doi.org/10.1164/RCCM.201705-0860OC/SUPPL_FILE/DISCLOSURES.PDF.
- Heitmann KR, Kauczor HU, Mildenerberger P, Uthmann T, Perl J, Thelen M. Automatic detection of ground glass opacities on lung HRCT using multiple neural networks. *Eur Radiol*. 2014;7(9):1463–72. <https://doi.org/10.1007/S003300050318>.
- Uppaluri R, Hoffman EA, Sonka M, Hartley PG, Hunninghake GW, McLennan G. Computer recognition of regional lung disease patterns. *Am J Respir Crit Care Med*. 2012;160(2):648–54. <https://doi.org/10.1164/AJRCCM.160.2.9804094>.
- Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84–90. <https://doi.org/10.1145/3065386>.
- van Tulder G, de Bruijne M. Learning features for tissue classification with the classification restricted Boltzmann machine. *Lect Notes Comput Sci*. 2014;8848:47–58. https://doi.org/10.1007/978-3-319-13972-2_5/COVER.
- Li Q, Cai W, Wang X, Zhou Y, Feng DD, Chen M (2014) Medical image classification with convolutional neural network. In: 2014 13th Int. Conf. Control Autom. Robot. Vision, ICARCV 2014; 2014. p. 844–8. <https://doi.org/10.1109/ICARCV.2014.7064414>.
- Gao M, et al. Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *Comput Methods Biomech Biomed Eng*. 2016;6(1):1–6. <https://doi.org/10.1080/21681163.2015.1124249>.
- Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Trans Med Imaging*. 2016;35(5):1207–16. <https://doi.org/10.1109/TMI.2016.2535865>.
- Christodoulidis S, Anthimopoulos M, Ebner L, Christe A, Mougiakakou S. Multisource transfer learning with convolutional neural networks for lung pattern analysis. *IEEE J Biomed Health Inform*. 2017;21(1):76–84. <https://doi.org/10.1109/JBHI.2016.2636929>.

23. Kim GB, et al. Comparison of shallow and deep learning methods on classifying the regional pattern of diffuse lung disease. *J Digit Imaging*. 2018;31(4):415–24. <https://doi.org/10.1007/S10278-017-0028-9/FIGURES/5>.
24. Wang Z, et al. Optimal threshold in CT quantification of emphysema. *Eur Radiol*. 2012;23(4):975–84. <https://doi.org/10.1007/S00330-012-2683-Z>.
25. Bae HJ, et al. A Perlin noise-based augmentation strategy for deep learning with small data samples of HRCT images. *Sci Rep*. 2018;8(1):1–7. <https://doi.org/10.1038/s41598-018-36047-2>.
26. Walsh SLF, Calandriello L, Silva M, Sverzellati N. Deep learning for classifying fibrotic lung disease on high-resolution computed tomography: a case-cohort study. *Lancet Respir Med*. 2018;6(11):837–45. [https://doi.org/10.1016/S2213-2600\(18\)30286-8](https://doi.org/10.1016/S2213-2600(18)30286-8).
27. Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S. CNN features off-the-shelf: an astounding baseline for recognition; 2014. p. 806–13.
28. Zheng L, Zhao Y, Wang S, Wang J, Tian Q. Good practice in CNN feature transfer (2016). <https://doi.org/10.48550/arxiv.1604.00133>.
29. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? *Adv Neural Inf Process Syst*. 2014;27:3320–8.
30. Cheplygina V, Pena IP, Pedersen JH, Lynch DA, Sorensen L, De Bruijne M. Transfer learning for multicenter classification of chronic obstructive pulmonary disease. *IEEE J Biomed Health Inform*. 2018;22(5):1486–96. <https://doi.org/10.1109/JBHI.2017.2769800>.
31. Wei X, Chen J, Cai C. Using deep convolutional neural networks and transfer learning for mammography mass lesion classification. *J Comput Theor Nanosci*. 2017;14(8):3802–6. <https://doi.org/10.1166/JCTN.2017.6676>.
32. Yap MH, et al. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE J Biomed Health Inform*. 2018;22(4):1218–26. <https://doi.org/10.1109/JBHI.2017.2731873>.
33. Lu Y, Chen L, Saidi A. Optimal transport for deep joint transfer learning (2017). <https://doi.org/10.48550/arxiv.1709.02995>.
34. Samala RK, Chan HP, Hadjiiski L, Helvie MA, Richter CD, Cha KH. Breast cancer diagnosis in digital breast tomosynthesis: effects of training sample size on multi-stage transfer learning using deep neural nets. *IEEE Trans Med Imaging*. 2019;38(3):686–96. <https://doi.org/10.1109/TMI.2018.2870343>.
35. Suzuki A, Sakanashi H, Kido S, Shouno H. Feature representation analysis of deep convolutional neural network using two-stage feature transfer—an application for diffuse lung disease classification (2018). <https://doi.org/10.48550/arxiv.1810.06282>.
36. Raju AHB, Augustine P. Identification of interstitial lung diseases using deep learning. *Int J Electr Comput Eng*. 2020;10(6):6283–91. <https://doi.org/10.11591/ijece.v10i6.pp6283-6291>.
37. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision; 2016. p. 2818–26.
38. Lin C, et al. Transfer learning based traffic sign recognition using inception-v3 model. *Period Polytech Transp Eng*. 2019;47(3):242–50.
39. Chollet F. Xception: deep learning with depthwise separable convolutions; 2017. p. 1251–8.
40. Dong K, et al. MobileNetV2 model for image classification. In: 2020 2nd International conference on information technology and computer application (ITCA). IEEE; 2020.
41. Han S, Mao H, Dally WJ. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. In: 4th Int. Conf. Learn. Represent. ICLR 2016—Conf. Track Proc; 2015. <https://doi.org/10.48550/arxiv.1510.00149>.
42. Hridayami P, Ketut Gede Darma Putra I, Wibawa KS. Fish species recognition using VGG16 deep convolutional neural network. *J Comput Sci Eng*. 2019;13(3):124–30.
43. Understanding the VGG19 Architecture. <https://iq.opengenus.org/vgg19-architecture/> (Accessed 8 Dec 2022).
44. Khan MSM, et al. Cataract detection using convolutional neural network with VGG-19 model. In: 2021 IEEE World AI IoT Congress (AIIoT). IEEE; 2021.
45. Understanding and Coding a ResNet in Keras | by Priya Dwivedi | Towards Data Science. <https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33> (Accessed 28 Dec 2022).
46. Wen L, Li X, Gao L. A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Comput Appl*. 2020;32:6111–24.
47. Moses DA. Deep learning applied to automatic disease detection using chest X-rays. *J Med Imaging Radiat Oncol*. 2021;65(5):498–517. <https://doi.org/10.1111/1754-9485.13273>.
48. Kundu R, Das R, Geem ZW, Han GT, Sarkar R. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLoS One*. 2021;16(9): e0256630. <https://doi.org/10.1371/JOURNAL.PONE.0256630>.
49. Alharbi AH, Hosni Mahmoud HA. Pneumonia transfer learning deep learning model from segmented X-rays. *Healthcare*. 2022;10(6):987. <https://doi.org/10.3390/HEALTHCARE10060987>.
50. Niu S, et al. A decade survey of transfer learning (2010–2020). *IEEE Trans Artif Intell*. 2020;1(2):151–66.
51. Kumari S, Kumar D, Mittal M. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *Int J Cogn Comput Eng*. 2021;2:40–6.
52. Sherazi SWA, Bae J-W, Lee JY. A soft voting ensemble classifier for early prediction and diagnosis of occurrences of major adverse cardiovascular events for STEMI and NSTEMI during 2-year follow-up in patients with acute coronary syndrome. *PLoS One*. 2021;16(6): e0249338.
53. 1.11. Ensemble methods—scikit-learn 1.2.0 documentation. <https://scikit-learn.org/stable/modules/ensemble.html> (Accessed 28 Dec 2022).
54. How to develop voting ensembles with Python—MachineLearningMastery.com. <https://machinelearningmastery.com/voting-ensembles-with-python/> (Accessed 28 Dec 2022).
55. Deep learning for image classification in Python with CNN. <https://www.projectpro.io/article/deep-learning-for-image-classification-in-python-with-cnn/418>.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.