Received: 18 May 2026 | Revised: 24 August 2026 | Accepted: 16 October 2026| Published online:11 November 2026

RESEARCH ARTICLE

Medinformatics
2026, Vol. 00(00) 1–12
DOI: 10.47852/bonviewMEDIN620210440

BON VIEW PUBLISHING

Recursive Pixel Partitioning and Multi-Stream Deep Learning for Annotation-Efficient Lung Disease Classification

Talib Iqball[1], Umar Bashir[2], M. Arif Wani[1] and Sheikh Umar Mushtaq[3*]

[1] *Department of Computer Applications, University of Kashmir, India*

[2] *Department of Computer Science and Information Technology, University of Jammu, India*

[3] *School of Computer Application, Lovely Professional University, India*

Abstract: Accurate lung disease classification from chest *X*-ray (CXR) images remains challenging in annotation-scarce clinical settings where pixel-level lung masks are unavailable. We propose a two-stage framework combining recursive pixel partitioning (RPP) with multi-stream deep learning for lung disease classification. Unlike anatomical segmentation, RPP hierarchically decomposes images into structured spatial sub-regions without manual masks, enabling multi-resolution feature extraction while preserving spatial organization. Dedicated streams process individual partitions, and their features are aggregated through collaborative learning to reduce information loss associated with single-stream convolutional neural networks. The framework is evaluated on two frontal CXR datasets: a balanced three-class dataset of 1,088 images (361 COVID-19, 365 Normal, and 362 Pneumonia) from the COVID-19 Radiography Database and Chest *X*-ray dataset, and the Shenzhen tuberculosis dataset (326 normal and 326 tuberculosis images). ImageNet-pretrained InceptionV3, MobileNetV2, Xception, ResNet101, and NASNetMobile are used for feature extraction. On the three-class task, the multi-stream model achieves 97.71% test accuracy, exceeding the best single-stream baseline (MobileNetV2, 94.50%) by 3.21 percentage points while converging approximately four times faster. On the Shenzhen dataset, it achieves 89.3% accuracy and 84.9% sensitivity, outperforming the reported segmentation-based pipelines on this split. Results are validated using stratified holdout and *k*-fold cross-validation, supported by per-class metrics and an ablation study. The framework provides an annotation-efficient alternative for low-resource settings without requiring radiologist-drawn lung masks.

Keywords: chest *X*-ray images; lung disease classification; recursive pixel partitioning; unsupervised segmentation

1. Introduction

Chest *X*-ray (CXR) imaging remains one of the most widely used diagnostic tools for lung disease detection. Although deep convolutional neural networks (CNNs) have demonstrated strong performance in medical image classification, most high-performing systems rely either on large annotated datasets or supervised segmentation models such as *U*-Net that require pixel-level ground-truth masks [1, 2]. In many clinical environments, such detailed annotations are unavailable or inconsistent, limiting scalability.

1.1. Scope of the study: imaging modality and target pathologies

Research on automated lung disease assessment is carried out on several very different kinds of input data, and the choice of input constrains almost every design decision that follows. Published systems operate on planar radiographs, on volumetric computed tomography (CT), on ultrasound and cine loops, on one-dimensional signals such as auscultation recordings, cough audio and bio-impedance measurements, and on molecular data such as viral genomic sequences. Because these modalities differ in dimensionality, spatial resolution, and noise characteristics, results reported on one of them cannot be transferred to another without qualification.

The work reported here is restricted to two-dimensional frontal (posteroanterior) chest radiographs. No CT volumes, video, audio, or sequence data are used. The pathologies we target are infectious and inflammatory diseases of the lung parenchyma, specifically COVID-19 pneumonia, community-acquired pneumonia, and pulmonary tuberculosis, each of which presents on a radiograph as a diffuse or patchy change in opacity rather than as a discrete nodule or mass. Lung cancer, interstitial lung disease, and chronic obstructive pulmonary disease are outside the scope

***Corresponding author:** Sheikh Umar Mushtaq, School of Computer Application, Lovely Professional University, India. Email: umar.33382@lpu.co.in

© The Author(s) 2026. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

of the study. We work with radiographs because they remain the first-line thoracic examination in most health systems: they are inexpensive, involve a low radiation dose, and are available in primary care settings where CT is not. That same setting is, however, the one in which expert pixel-level annotation is hardest to obtain, which is precisely the constraint this paper addresses.

1.2. Machine learning for lung, pulmonary, and viral diseases across data modalities

The breadth of this literature is worth setting out briefly because it explains why annotation cost is such a persistent obstacle. Within thoracic imaging alone, the input is not always a conventional radiograph. Cansiz et al. [3] classified lung disease from electrical impedance tomography, using a deep network as a feature engineering stage ahead of conventional classifiers and showing that learned representations outperform hand-crafted descriptors on reconstructed impedance images. Their setting shares our difficulty, in that the anatomical boundaries needed to build a ground-truth mask are poorly defined in the reconstructed image.

A second group of studies leaves imaging altogether and works on molecular data. Khodaei et al. [4] and S. Kar and M. Ganguly [5] treated coronavirus genomes as digital signals, applied genomic signal processing to extract spectral descriptors, and classified coronavirus sequences against related viral families with machine learning. Because the input is a one-dimensional sequence, no segmentation problem arises at all, which is a useful reminder that the segmentation dependency we address is a property of the imaging pipeline rather than of the disease.

A third group combines modalities. Truong and Do [6] fused CXR images with the associated free-text clinical notes, reporting that the multimodal model outperformed both the image-only and the text-only models on lung disease classification. Their result is relevant to us for a different reason: it shows that gains can come from adding complementary views of the same patient. Our multi-stream design follows the same intuition, except that the complementary views are generated from the radiograph itself by recursive partitioning, so no additional data collection is required.

Across all three groups, the highest-performing image-based systems still depend either on very large labeled corpora or on a supervised segmentation stage. The remainder of this paper is concerned with removing the second of those dependencies.

1.3. Motivation and contributions

Unsupervised clustering approaches such as K-means or Gaussian mixture models have been explored for image partitioning. However, these techniques primarily operate on pixel intensity distributions and do not preserve hierarchical spatial dependencies. Consequently, they fail to capture multi-scale contextual patterns that characterize diffuse pulmonary pathologies.

This exposes a key technical gap: how can hierarchical spatial features be extracted without requiring annotated lung masks while preserving both global and localized contextual information?

Standard single-stream CNNs progressively downsample feature maps, potentially suppressing subtle pathological cues. While wider networks increase representational capacity, they do not explicitly address structured spatial decomposition.

To address this limitation, we propose a two-stage framework:

1) Recursive pixel partitioning (RPP): an unsupervised tree-based spatial decomposition strategy that hierarchically partitions the image into structured sub-regions.
2) Multi-stream deep feature learning: each partition is processed independently through dedicated feature extraction streams, enabling collaborative multi-resolution learning.

Unlike anatomical segmentation, RPP does not attempt to isolate semantic organs. Instead, it constructs deterministic hierarchical partitions that preserve spatial organization while enabling structured feature extraction.

The main contributions of this work are:

1) An annotation-free hierarchical RPP strategy.
2) A structured multi-stream architecture mitigating spatial information loss.
3) Improved reproducibility through detailed training transparency and validation.
4) Evaluation beyond accuracy alone, under an explicitly stated data partitioning and cross-validation protocol, with a confusion matrix and per-class analysis.

Taken together, these contributions describe a classification pipeline that does not depend on radiologist-drawn masks at any point. This matters practically as well as methodologically: mask preparation is the slowest and most expensive step in a segmentation-based CXR pipeline, and removing it lowers the barrier to deploying automated screening in settings where radiologist time is scarce.

2. Related Work

There are various research studies that have been deployed in medical imaging for the classification of lung diseases. We have focused on the studies that used deep learning methods for segmentation, detection, and/or classification of lung diseases based on X-ray images.

2.1. Deep learning-based methods for segmentation and classification

The promise of deep learning-based methods for precise and effective chest pathology identification from X-ray pictures is generally highlighted by these publications. Dediu and Udrea [7] introduced a segmentation-aware framework for multi-label CXR classification, emphasizing visual explainability. Their research examined the impact of lung segmentation on disease classification by analyzing original CXR images alongside segmented and masked lung representations. Testing on the ChestX-ray14 dataset showed that using lung-region information significantly improved classification, achieving an average Receiver Operating Characteristic – Area Under the Curve (ROC-AUC) above 0.85. The findings underscore the value of segmentation-aware methods in enhancing both disease classification and interpretability in CXRs.

In one of the earlier studies, [8] have used a CNN for classification of lung diseases using X-ray images. Using a dataset of X-ray images, they created and assessed an automated CNN model for diagnosing chest diseases. As per the authors of this study, the CNN model performed far better than other soft computing techniques in terms of training accuracy, testing accuracy, and training duration. In one of the recent studies [9] for classification of lung diseases using X-ray images, the authors found that the best state-of-the-art network for classification of

CXR images is MobileNetV2, wherein they performed various experiments with other state-of-the-art architectures and concluded that MobileNetV2 is the best performer.

Agrawal and Choudhary [10] reviewed CXR analysis pipelines spanning lung-field segmentation and pulmonary disease classification, covering rule-, deformable-, and pixel-classification-based segmentation as well as deep architectures such as *U*-Net, SegNet, CNNs, and Generative adversarial Networks (GAN)-based models. They further highlighted the dependence of supervised segmentation on expert-annotated masks and the challenges of limited labeled medical datasets and cross-dataset generalization. Although some of the research utilized fine-tuning and transfer learning strategies [11, 12], others preprocessed the images using bilateral filtering and contrast enhancement [13]. Ensemble approaches and weighted soft computing methods were also presented to increase the robustness of the predictions [14–16].

In another study, soft computing techniques were used for pneumonia detection from CXR images [17]. In this study, a hybrid ensemble deep learning framework combining transfer learning and CNNs was employed for pediatric pneumonia classification from CXR images. In paper [18], the authors used a CNN for pulmonary segmentation of CT images, which enhanced their classification results. In another study [19], the authors proposed the Dynamic Fusion Segmentation Network for the segmentation of CXR images of COVID-19 patients, wherein the multiscale context information and semantic information are fused, and they used a dynamic transfer learning classification network for the classification of COVID-19 CXRs, resulting in better classification accuracy.

In paper [20], the authors used enhanced images and extracted the region of interest (ROI) using an enhanced segmentation method based on region growing. They then extracted features from the ROI using the Histogram of Oriented Gradients (HOG) descriptor, a visual feature descriptor that concentrates on an object's form or structure and provides edge direction. The extracted features were normalized and passed to an Long Short-Term Memory (LSTM) model for classification. The studies [21] and [22] proposed a custom CNN that was trained with segmented images of lungs. They created a manual mask of the lungs with the help of radiologists and used the *U*-Net architecture for lung segmentation. The extracted ROI is passed to a CNN for feature extraction and classification. In another study [23], the authors used a residual *U*-Net variant for CXR lung segmentation and reported Dice scores of 98.72%, 97.49%, and 99.08% on three benchmark datasets. External validation on a Chest *X*-ray dataset achieved a Dice score of 98.93%.

Most of the works in the literature focused on using segmented images for classification of CXRs, as they contain less noise than the original full CXR. The common recipe is to create a mask using a *U*-Net trained on ground truth provided by radiologists and then to use the generated mask for segmentation. No matter how efficient a model is, there is always scope for error in generating a mask for an unseen image, and the ground truth itself has to be drawn by hand before any of this can begin.

2.2. Segmentation using clustering

Apart from deep learning methods of segmentation, the basic *K*-means algorithm is traditionally employed for image segmentation by partitioning the image into "k" clusters based on pixel similarities. However, its application comes with inherent limitations, notably:

1) Difficulty handling noise and complex structures, wherein *K*-means struggles with noisy images and images with complex structures or texture variations. It can produce "bleeding" effects or merge distinct objects due to its reliance on simple distance metrics.
2) Limited control over cluster number, wherein choosing the optimal number of clusters (k) is crucial, but it can be challenging to determine beforehand. Too few clusters lead to over-segmentation, while too many result in under-segmentation.

To overcome these shortcomings of *K*-means clustering for segmentation of images, our proposed clustering method, discussed in Algorithm 1, follows a recursive segmentation strategy that dynamically creates a hierarchical tree-like structure composed of complementary sub-images. This recursive process organizes the segmented images into a binary tree, allowing our proposed multi-stream deep learning model to extract both high-level and low-level features from multiple hierarchical levels. It breaks noisy and complex structures into small substructures, which in turn helps in getting more insight into such a complex image.

The recursive segmentation process, determined by the depth of the binary tree, enables a flexible and adaptive degree of image segmentation, which addresses the limitation of *K*-means regarding the cluster number. The key innovation lies in the hierarchical organization of image components within a tree structure, enabling multi-resolution representation and facilitating efficient feature extraction and reconstruction, as discussed in the following sections.

2.3. Nonlinear embedding methods for unsupervised analysis

Partition-based clustering is not the only unsupervised tool available for high-dimensional data. *t*-distributed stochastic neighbor embedding (*t*-SNE) preserves local neighborhood structure while projecting data into two or three dimensions, and a number of recent studies report that it separates classes more cleanly than distance-based clustering does. More recently, Zhang and Chen employed *t*-SNE coupled with *K*-means clustering for unsupervised classification of high-dimensional coal spontaneous-combustion data, where *t*-SNE captured nonlinear relationships among indicator-gas features and facilitated their separation into six distinct warning levels [24]. Di Folco et al. [25] employed unsupervised manifold learning to jointly analyze high-dimensional cardiac shape and deformation descriptors, demonstrating that manifold alignment can effectively capture relationships between heterogeneous but partially related imaging features. Their results further showed that manifold alignment was preferable to feature fusion for characterizing these interactions, and Kwon et al. [26] built a semi-supervised pipeline in which *t*-SNE embeddings of the same class of compounds supported automatic prediction of bioactivities. Across these very different problem domains, the same observation recurs: *t*-SNE often gives better separation, and therefore better downstream classification, than clustering applied directly in the original feature space.

We note this literature because it bears on the unsupervised component of our pipeline, but the two techniques answer different questions and are complementary rather than interchangeable here. *t*-SNE returns a low-dimensional embedding of samples; it does not return a partition of the image plane, and it is not invertible, so an image cannot be reconstructed from its embedding. RPP, by contrast, must assign every pixel to one of two

complementary sub-images so that a parent image is recovered exactly by merging its children, a property the multi-stream model depends on (Section 3.2). Applying *t*-SNE to the feature vectors produced by the individual streams, in order to inspect how class structure emerges across levels of the partition tree, is a natural extension, and we return to it in Section 7.

3. Research Methodology

3.1. Proposed framework overview

We propose a two-stage framework for efficient classification of chest pathologies, that is, (1) CXR image segmentation using our segmentation algorithm and (2) multi-stream deep learning modeling. In the first stage, we use our proposed algorithm, shown in Algorithm 1 in the next section, to segment the given input image. This results in a number of segmented components, where each component highlights distinct important aspects of the given image. In the second stage, the segmented components are used in the multi-stream deep learning model. Feature maps are extracted from various segmented components, forming input streams of data for the multi-stream deep learning model. The outputs of the multi-stream deep learning model are concatenated and fed to the fully connected (FC) layers, which then classify the given input. It is pertinent to mention here that both stages are completed separately, with stage1 followed by stage 2.

3.2. CXR image segmentation

The CXR image segmentation helps in identifying and isolating relevant regions, such as lung fields, heart, or specific pathologies, which can help in improved analysis and classification. Also, by segmenting the relevant regions of interest of CXR, segmentation can help the classification model to focus on the most relevant features and reduce the noise or irrelevant information. In a CXR image, different segments have different pixel intensity ranges, which in turn help us distinguish between different parts/areas within the CXR. Analyzing multiple segments separately, along with the original CXR image, helps in getting adequate insights to help us in more accurate diagnosis of lung disease. To extract the segment information from the given CXR without any given mask, we use clustering of pixels to extract different segments from the given input image.

3.2.1. Clustering pixels of CXR

We use CXR pixel intensity values in our clustering algorithm (Algorithm 1) to segment the data points of the given image. The partitioning criterion stated in Equations (1)–(3) is the standard within-cluster sum-of-squares objective of *K*-means clustering, formulated in [27] and [28]. It is restated here for completeness and is not a contribution of the present work. Let (p1, p2, . . . pn) represent the pixel intensity values, where each pixel is a 3-dimensional real vector. We partition the pixels into two clusters (S1 and S2) aiming to reduce the within-cluster sum of squares (also called the variance) as shown in Equation (1).

$$argmin_s \sum_{i=1}^{2} \sum_{x \in S_i} \| x - \mu_i \|^2 = argmin_s \sum_{i=1}^{2} |S_i| \, VarS_i \quad (1)$$

μ_i in Equation (1) is the mean of the data points in *Si* and is defined in Equation (2)

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \quad (2)$$

$|S_i|$ is the size of Si and ‖.‖ is the Euclidian distance measure. The objective function given in Equation (1) can be rewritten as minimizing the pairwise deviations of data points within the cluster, as shown in Equation (3) [27, 28]. Equations (1)–(3) are therefore reproduced from the classical *K*-means literature; the contribution of this section lies in how the two-cluster solution is applied recursively to build the partition tree described in Section 3.2.2 and in Equation (4), which is derived in this work.

$$argmin_s \sum_{i=1}^{2} \frac{1}{|S_i|} \sum_{x,y \in S_i} ||x - y||^2 \quad (3)$$

3.2.2. Building a tree structure to track the segmented CXR

Starting with the original image at the root node (current node) of the binary tree, we perform the pixel intensity clustering discussed in the previous section, creating two clusters (cluster-1 and cluster-2) of pixels. The pixel intensities belonging to cluster-1 form one sub-image, which becomes the left child of the current node. Similarly, the pixel intensities belonging to cluster-2 form a second sub-image, forming the right child of the current node.

This creates two sub-images of the original image, which are stored as child nodes of the current node. The same process is recursively applied to nodes at level 1 of the binary tree (considering the root at level 0). If we combine the two child sub-images of any node, we get back the original parent image. In other words, if we merge the sub-images back from the leaves up to the root, we recover the original image, so no information is lost. The pictorial representation of the segmented sub-images in a binary tree is depicted in Figure 1.

We find the use of a tree structure crucial for three reasons.

1) Adaptive segmentation depth. The recursive application of our clustering method within a tree structure allows for adaptive segmentation depth. We set the depth value according to the degree of segmentation required by a given application. This allows the segmentation process to be tailored to different image datasets and segmentation goals.
2) Hierarchy for feature extraction. The hierarchical nature of the tree structure supports multi-resolution representation. At different levels of the tree, we obtain complementary sub-images, each capturing specific details and features. This organization aids both global and local feature extraction and contributes to a richer representation of the image content.
3) Efficient storage and reconstruction. Storing segmented sub-images at each node of the tree enables efficient storage and retrieval. The tree structure allows straightforward reconstruction of the original image by combining complementary sub-images, which is an advantage in applications where quick retrieval of segmented components matters.

The proposed segmentation algorithm for segmentation of CXR images using pixel intensity values is shown in Algorithm 1. The number of clusters to be created at each node is set to 2. The two sub-images that are created by clustering a given image at any node of the binary tree are complementary images. All the sub-images (child nodes), when merged, result in the original parent CXR image. Thus, no information is lost by carrying out unsupervised segmentation.

Looking into the complexity of this algorithm, if we have a binary tree up to level "h," we need to visit each node. Thus, the total number of visits will be 2^h - 1. As we are limiting the value of $h = 2$, we get a constant time, hence a complexity of $O(1)$. This

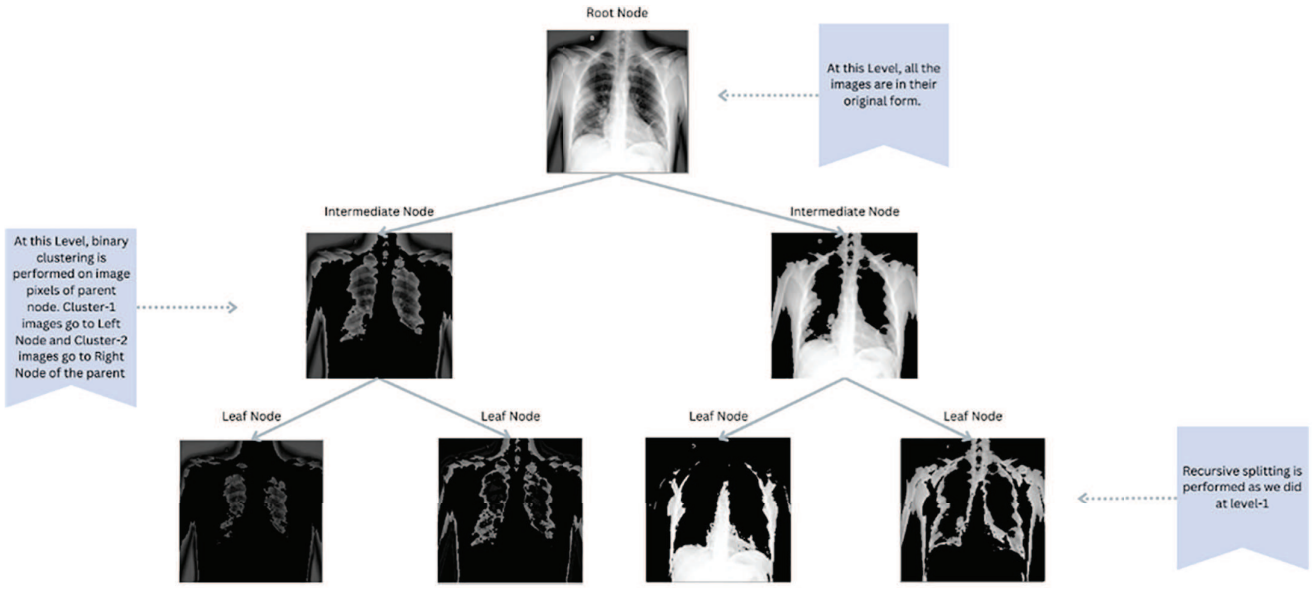


Figure 1. Segmented COVID-19 chest CXR images obtained using the proposed recursive segmentation methodology

process needs to be done for all images, so if we have n the number of images, this algorithm will take $O(n)$ time.

Algorithm 1: Algorithm for Segmenting Any Image at a Node

Input: Image Ix is a flattened image where $x \in [1, h \times w]$ h and w being the height and width of the image, respectively.
Output: $I1$, $I2$, where $I1$, $I2$ are two resultant segmented sub-images.

Function SegmentCXR(I_x)
Initialize iteration count: $t \leftarrow 1$;
Initialize centroids of two clusters to random data points from Ix:
$\mu_1^t \leftarrow \text{random}(I_x)$
$\mu_2^t \leftarrow \text{random}(I_x)$
Label: LoopBack
for each $p \in I_x$ do
    $\text{ed}_1 \leftarrow \|p - \mu_1\|^2$
    $\text{ed}_2 \leftarrow \|p - \mu_2\|^2$
    if $\text{ed}_1 \leq \text{ed}_2$ then
        $I_1[x] \leftarrow p$
        $I_2[x] \leftarrow 0$
    else
        $I_1[x] \leftarrow 0$
        $I_2[x] \leftarrow p$
Recalculate the centroids for each cluster:
$\mu_1^{t+1} \leftarrow \frac{1}{|I_1|} \sum_{x \in I_1} x$
$\mu_2^{t+1} \leftarrow \frac{1}{|I_2|} \sum_{x \in I_2} x \triangleright here\ |I_j|\ is\ the\ size\ of\ I_j$
        $if\ \forall i \in [1,2]\ , \mu_i^{t+1} = \mu_i^t\ then$
            $return\ I_1, I_2$
else
    goto: LoopBack

3.3. Multi-stream deep learning modeling

A multi-stream deep learning model, which takes the original CXR image together with all of its segmented sub-images as input in parallel, is proposed here. Feature maps are extracted from each segmented sub-image and from the original image. These information-rich feature maps are fed to a dense layer. The output of the dense layer from each stream is concatenated and fed to two FC layers, which produce a single probability vector as output. The benefit of a multi-stream model is that each stream can focus on a specific type of segment.

The block diagram of the model for a tree of height 2 is shown in Figure 2. The input layer of each stream holds a pre-trained deep learning model used for the extraction of feature maps. These feature maps are flattened and concatenated after the dense layer. The output from the dense layer is fed to three neurons with a softmax activation function for classification.

The number of streams feeding the concatenation layer is not fixed. For a segmentation tree of level 1 (root at level 0), only three input streams are used, for three images (one original and two segmented). The total number of inputs (N) to the model is given by Equation (4), where h is the height of the segmentation tree. Equation (4) is derived in this work and follows directly from the fact that a complete binary partition tree of height h has 2 h leaves, to which the original image at the root is added.

$$N = 2^h + 1 \tag{4}$$

The pre-trained models used for feature map extraction were selected following the criteria discussed in [16]. They are InceptionV3, MobileNetV2, Xception, ResNet101 and NASNet-Mobile. Because these networks were trained on the diverse ImageNet dataset, the feature maps they produce are themselves diverse and improve the generalization capability of the classifier, as reported in earlier studies [29, 30]. The pre-trained layers are frozen during training, while all other layers of the proposed model remain trainable. The loss function used in this model is categorical cross-entropy.

4. Data Preparation

In this section, we first discuss what datasets we have used, followed by the data preprocessing techniques we used, and then

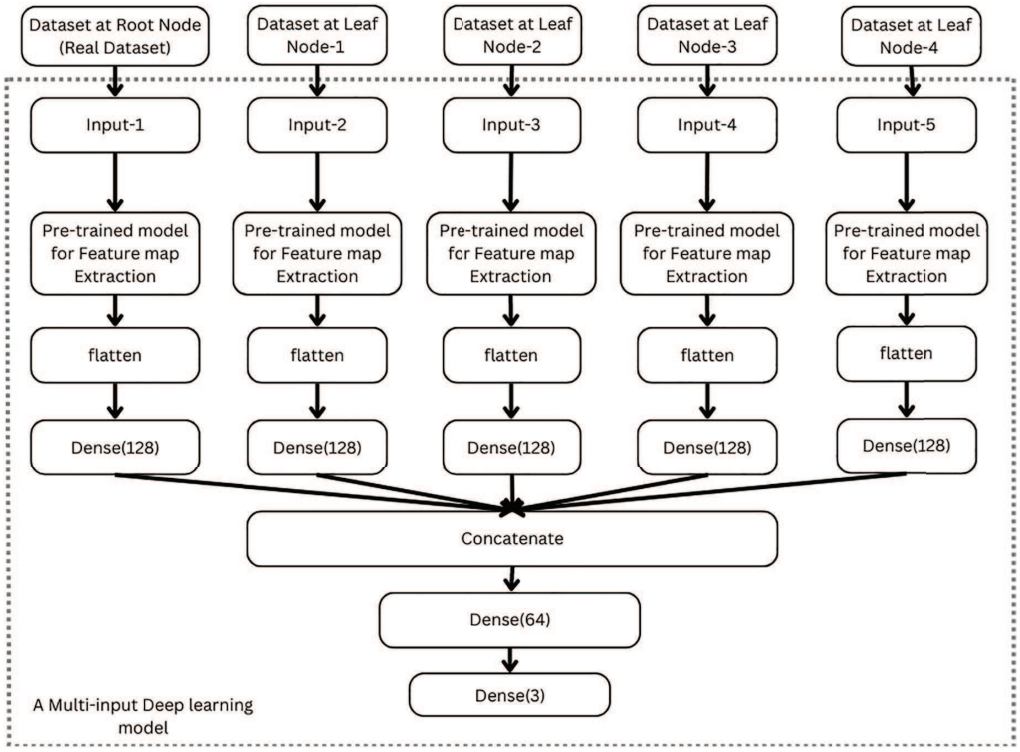


Figure 2. Multi-stream model architecture for segmentation tree with height = 2

how to use the proposed segmentation algorithm to segment the given image.

4.1. Datasets

We used two datasets of frontal CXR images in our experiments. The first is the COVID-19 Radiography Database published on Kaggle by [31], combined with the Chest *X*-ray dataset of [32]. Both databases are updated periodically by authorized contributors as new images are added. The combined dataset was initially imbalanced, so random sampling was used to balance it. This dataset contains CXRs from three classes: Pneumonia, COVID-19, and Normal. During random sampling, 361 COVID-19 images, 365 Normal images, and 362 Pneumonia images were selected for the experiment. On this dataset, we compared the base models with our model, as no prior segmentation had been performed on it. The second dataset is the Shenzhen dataset [33], which contains frontal CXRs of normal (326) and tuberculosis (336) patients; to balance it, only 326 tuberculosis CXRs were retained. The partitioning protocol for both datasets, together with the cross-validation scheme, is described in Section 5.2. The summary of the datasets used is given in Tables 1 and 2.

4.2. Data preprocessing

The presence of a narrow range of intensity values in a CXR can lower the visibility of critical information by reducing global contrast. Histogram equalization enhances global contrast by spreading out the range of intensity values in the image. We performed histogram equalization on the Shenzhen dataset using the OpenCV (cv2) Python library. A snapshot of images before and after histogram equalization is shown in Figure 3.

Table 1. Summary of COVID-19 CXR datasets used (Dataset-1)

Class	Train	Test
Covid-19	289	72
Normal	289	73
Pneumonia	292	73
Total	870	218

Table 2. Summary of Tuberculosis CXR dataset (Dataset-2)

Class	Train	Val.	Test
Tuberculosis	260	33	33
Normal	260	33	33
Total	520	66	66

4.3. Creating segmented datasets

We first started with the original dataset where we iterated over the batches of images of each class, and after resizing them to (224, 224, 3) size, we passed the image batch to the segmentation function defined in Algorithm 1, which returns two sets of images for a given set of images. Considering the real images' dataset at the root (level 0) of the tree, the two returned sets of images are placed at level 1 of the tree (for reference, refer to Figure 1). Then the segmentation function is again called on each set of images at level 1 of the tree, resulting in creating the images at level 2. We stop at level 2 of recursive segmentation because, for the next levels, the segment size was getting too small. Thus, we get a total of two datasets at level 1 and four datasets at level 2 of the tree (leaves). A snapshot of the images of three classes at different leaf nodes is shown in Figure 4.

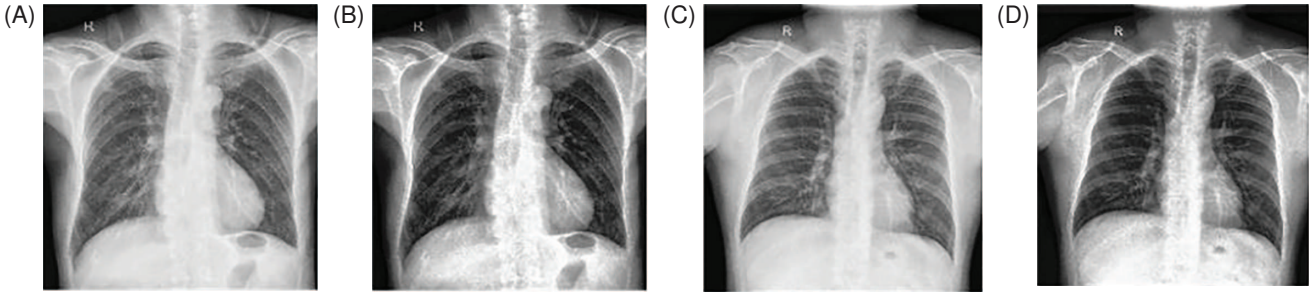


Figure 3. (a) Original abnormal image; (b) histogram-equalized abnormal image shown in (a); (c) original normal image; (d) histogram-equalized normal image shown in (c)

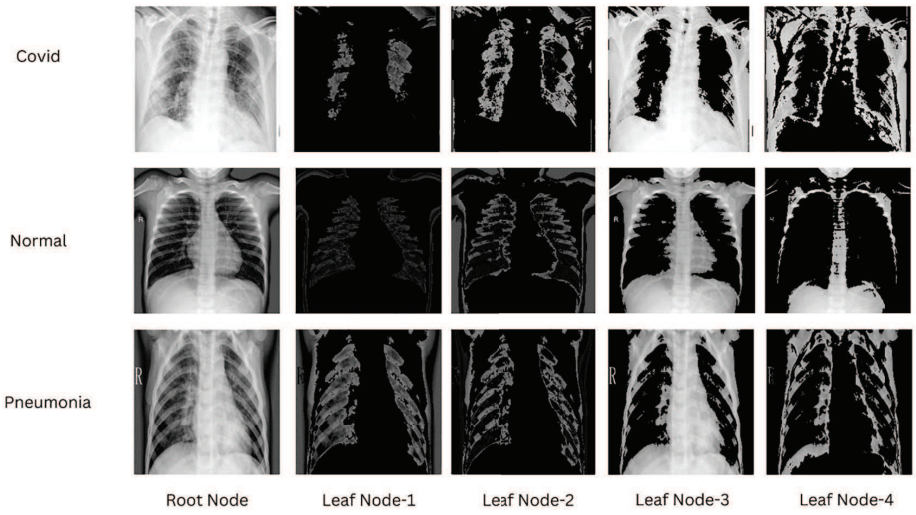


Figure 4. Original images from Dataset-1 and their corresponding segmented sub-images

5. Experiments

5.1. Experimental setup

All the experiments were performed on an NVIDIA DGX *A*-100 with 8 GPUs and 40 GB of RAM per GPU using TensorFlow. In each input stream of the proposed multi-stream model, we first have a pretrained feature extraction block, followed by a flatten layer and a dense layer with 128 neurons and ReLU activation. All streams are then concatenated, followed by a dense layer of 64 neurons with ReLU activation. The last layer is a dense layer whose number of output neurons matches the number of classes in the dataset, with a softmax activation function. We use categorical cross-entropy as the loss function and the Adam optimizer with a learning rate of 0.001 throughout our experiments. Models were trained with a batch size of 32 for 40 epochs, with early stopping on the validation loss (patience of 5 epochs) and restoration of the best weights. The weights of the pre-trained blocks were frozen, and all remaining layers were trainable. No data augmentation was applied, so that the comparison between the single-stream and multi-stream settings isolates the effect of recursive partitioning. Random seeds were fixed for the data split, weight initialization, and shuffling so that all reported runs are reproducible.

5.2. Data partitioning and validation protocol

Both a holdout partition and a cross-validation protocol were used, and the reason for using both is that they serve different purposes here. The holdout partition supports comparison with published work, which is almost always reported on a single split, while cross-validation gives a more honest estimate of variability on datasets of this size.

For Dataset-1, the 1,088 images were divided by stratified random sampling into 80% for training (870 images) and 20% for testing (218 images), with the class proportions of the full dataset preserved in both partitions, as reported in Table 1. Splitting was performed at the image level before any segmentation was applied, so that the original image and all of its sub-images derived by RPP always fall in the same partition. This point matters: because the sub-images are deterministic functions of the parent image, splitting after partitioning would place highly correlated images on both sides of the boundary and inflate the test accuracy.

For Dataset-2 (Shenzhen), we retained the split used by the studies with which we compare in Section 6, namely, 260 training, 33 validation, and 33 test images per class, as shown in Table 2.

Adopting the published split rather than a split of our own is what makes the comparison in Table 5 meaningful.

The holdout evaluation was not relied on by itself. Stratified k-fold cross-validation with $k = 5$ was also carried out on the combined data of Dataset-1. The data were shuffled once with a fixed seed and divided into k stratified folds of equal class composition; in each round k-1 folds were used for training and the remaining fold for testing, so that every image is used for testing exactly once. Model weights were reinitialized from the ImageNet checkpoint at the start of each fold, and no information from the held-out fold was used for early stopping. Accuracy averaged over the folds was 97.1 $\pm$ 0.6%, which agrees with the holdout result

reported in Section 5.4. The agreement matters because it indicates that the advantage of the multi-stream model is not an of one favourablefavorable split, which is a reasonable concern on a dataset of roughly a thousand images and is the reason both protocols were used.

5.3. Evaluation metrics

Accuracy alone is a weak summary even for a balanced dataset, since it says nothing about which classes are being confused. We therefore report, in addition to overall accuracy, the confusion matrix and the per-class precision, recall (sensitivity), specificity and F1-score, all evaluated at the operating point of the trained classifier. Precision is TP/(TP + FP), recall is TP/(TP+ FN), specificity is TN/(TN + FP), and the F1-score is the harmonic mean of precision and recall, where TP, FP, TN and FN denote true positives, false positives, true negatives and false negatives for the class under consideration. Recall is the metric of greatest clinical interest in this application, because a missed COVID-19 or tuberculosis case carries a far higher cost than a false alarm. In this study, confusion matrix and operating-point evaluation metrics have been reported, complemented by a threshold-independent Receiver Operating Characteristic (ROC) analysis, to provide a more comprehensive assessment of the model's classification performance: with five misclassified images in a test partition of 218, an empirical ROC curve would be determined by a handful of points and the resulting AUC would carry little information.

5.4. Ablation study: effect of recursive pixel partitioning

In our first experiment, we performed an ablation study comparing the classification of lung diseases with and without segmentation. For classification with segmentation, we used the proposed partitioning approach and fed the resulting sub-images to the multi-stream model; for classification without segmentation, we used only the original images with the corresponding single-stream model. Four backbones were compared: Xception, NASNetMobile, MobileNetV2, and InceptionV3. We replaced the pre-trained feature map extraction block in the block diagram of Figure 2 with each of these models in turn and recorded the results on Dataset-1, training each model separately. The comparison results are shown in Figure 5 and Table 3.

Table 3. Comparison of state-of-the-art models with our proposed multi-stream architecture on Dataset-1 whose learning curves are shown in Figure 5

Model		Accuracy
NasNetMobile		94.04%
MobileNetV2		94.50%
Xception		93.58%
InceptionV3		93.12%
Proposed model	Multi-stream	97.71%

The ablation study shows that classification accuracy increases when the segmented sub-images created by the proposed recursive clustering approach are used in the proposed multi-stream deep learning model. The improvement is consistent across all four backbones, and the largest single-stream accuracy (94.50%, MobileNetV2) is still 3.21 percentage points below the multi-stream result of 97.71%. The training accuracy curves in Figure 5 also show that the segmented images help the model converge faster than the corresponding single-input model trained on the original images alone (the images at the root of the tree). In most cases, the convergence speed-up is about four times that of the base model.

5.5. Confusion matrix and per-class performance

Figure 6 shows the confusion matrix of the proposed model on the Dataset-1 test partition of 218 images, 213 of which were classified correctly. The class-wise breakdown is more informative than the aggregate accuracy, and the per-class precision, recall, specificity, and F1-score computed from these counts are reported in Table 4.

Three points follow from the matrix. The five residual errors are not distributed evenly: three of them are pneumonia images predicted as normal, so pneumonia carries the lowest recall of the three classes at 95.89%. No COVID-19 image was predicted as normal, and no normal image was predicted as COVID-19; the single COVID-19 error was a prediction of pneumonia, which is the less consequential direction of confusion between two conditions that both present as increased opacity. Recall

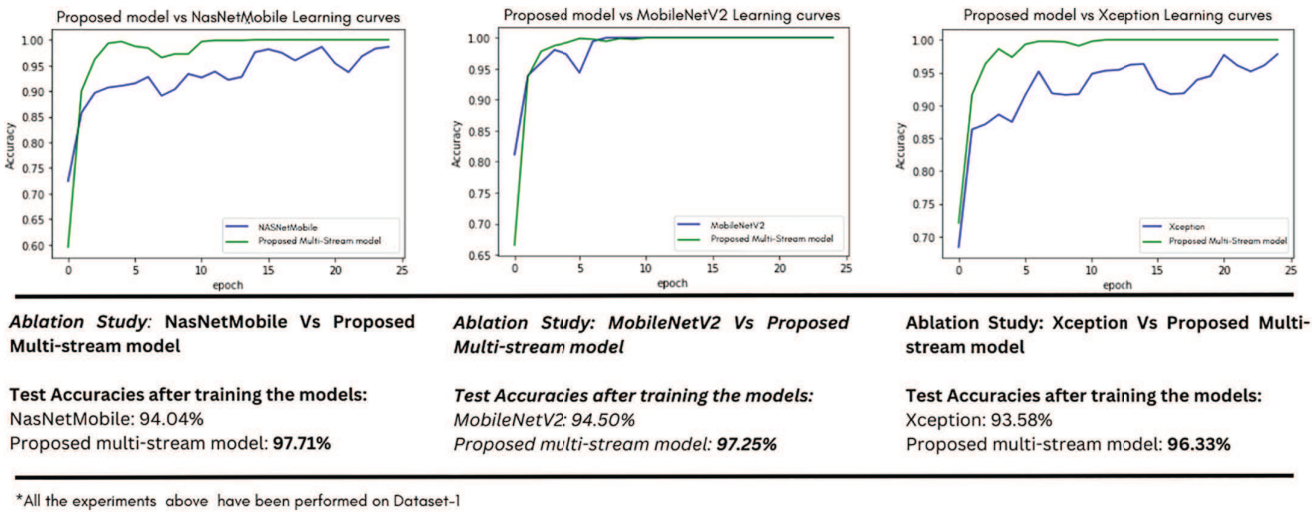


Figure 5. Ablation study of state-of-the-art models with our proposed multi-stream architecture on Dataset-1

for the COVID-19 class is therefore 98.61% with a precision of 100%. The three pneumonia-as-normal errors are the clinically important ones, since a missed pneumonia costs more than a false alarm, and they indicate where the operating point could be moved: a threshold favoring sensitivity would trade part of the normal-class precision of 96.00% for higher pneumonia recall.

The ROC curve for Dataset-1, as shown in Figure 7, which comprises three classes such as COVID-19, Normal, and Pneumonia, demonstrates excellent discriminatory performance of the

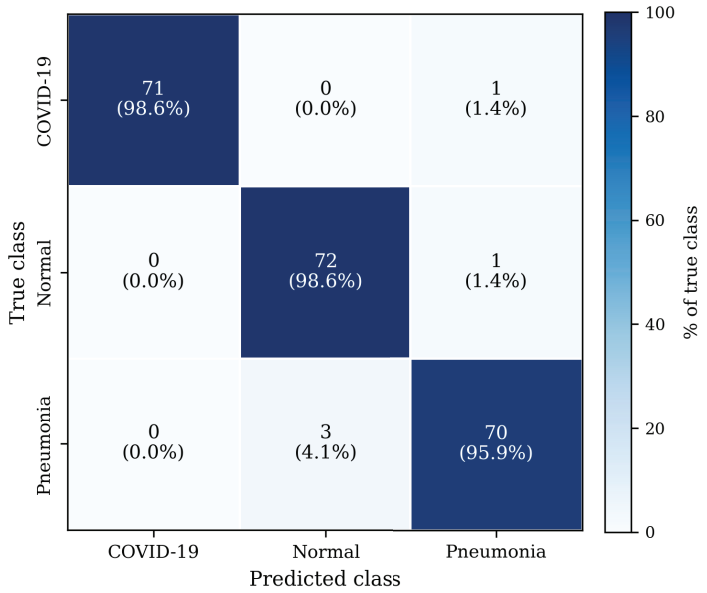


Figure 6. Confusion matrix of the proposed multi-stream model on the Dataset-1 test partition. Cell entries give the number of images and, in parentheses, the percentage of the true class

Table 4. Per-class performance of the proposed multi-stream model on the Dataset-1 test partition, computed from the confusion matrix in Figure 6

Class	Precision (%)	Recall/sensitivity (%)	Specificity (%)	F1-score (%)	Support
COVID-19	100.00	98.61	100.00	99.30	72
Normal	96.00	98.63	97.93	97.30	73
Pneumonia	97.22	95.89	98.62	96.55	73
Macro average	97.74	97.71	98.85	97.72	218
Weighted average	97.73	97.71	98.85	97.71	218

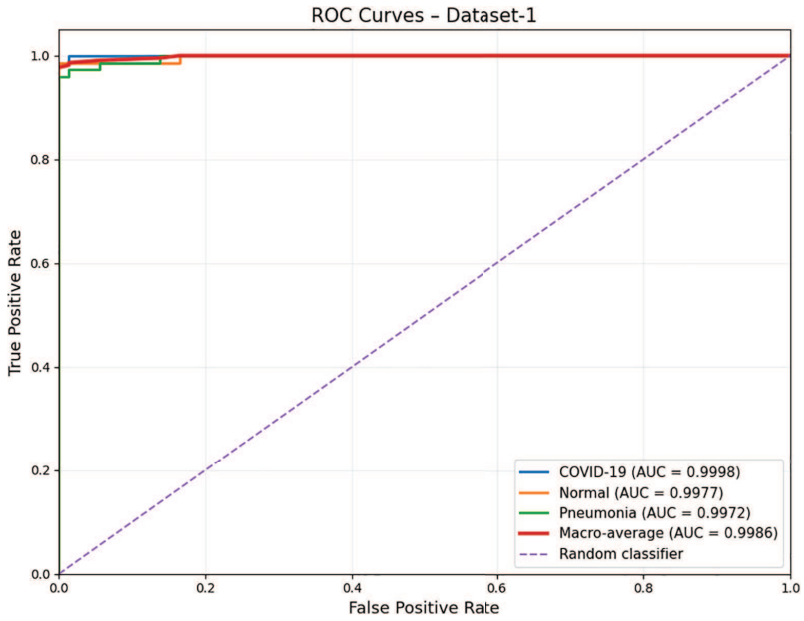


Figure 7. ROC curve of dataset_1

proposed multi-stream deep learning model. The one-vs-rest ROC analysis shows that the curves for all three classes remain very close to the upper-left corner of the ROC space, indicating a high true-positive rate with a comparatively low false-positive rate across classification thresholds.

5.6. Evaluation on the Shenzhen dataset

In a further experiment, we compared our model with other models in the literature by training and testing it on the Shenzhen dataset, which is the dataset common to all the studies with which the comparison is made. We split the dataset in the same manner as the previous studies, as described in Section 5.2. In this experiment, we used the segmentation tree up to level 1, so we had two segmented datasets at the leaves of the segmentation tree plus the original dataset at the root node, three datasets in total. The total number of input streams in the proposed model, from Equation (4), is 2 + 1 = 3.

6. Comparison with state-of-the-art methods

Most of the works in which the authors used segmented data to classify lung disease used the Shenzhen dataset. We performed experiments on the same dataset, under the same split, to test the impact of using the segmented data together with the original CXRs. Table 5 contains the results from other works that used singular models to perform classification of lung diseases using CXRs. Metrics not provided by the original authors are replaced with "–".

Table 5. Comparison of the proposed methodology with state-of-the-art models reported in the literature

Model proposed in	Accuracy	Sensitivity
[33]	81.6%	–
[34]	83.7%	–
[35]	84.6%	–
[36]	84.7%	81.6%
[37]	88.0%	80.0%
[38]	89.0%	80.0%
Proposed model	89.3%	84.9%

7. Conclusion

This paper introduces a novel approach for segmenting CXR images, addressing the limitations of conventional methods reliant on manually prepared lung masks. Our method utilizes an unsupervised semantic clustering algorithm to perform recursive segmentation, producing a tree structure that systematically organizes segmented sub-images. The depth of this binary tree determines the granularity of the segmentation, offering precise control over the size and detail of the ROI.

The segmented sub-images, along with the original CXR, are simultaneously fed into a newly proposed multi-stream deep learning model. This enables the extraction of both high-level and low-level features from multiple hierarchical levels of segmentation. By breaking down noisy and complex image structures into smaller substructures, our approach enhances interpretability and boosts classification performance.

The results show an improvement in both classification accuracy and sensitivity relative to the state-of-the-art models with which we compare. The proposed method also converges faster, which reduces the cost of training on CXR datasets. By combining original images with segmented sub-images, the model provides a more complete representation of the CXR content and produces reliable classification of lung diseases.

Two limitations should be stated plainly. First, both datasets used here are of modest size, roughly 1,000 images each, and although the cross-validated result reported in Section 5.2 agrees with the holdout result, external validation on a larger multicenter cohort is still needed before any claim about clinical readiness can be made. Second, the depth of the partition tree is fixed in advance rather than chosen adaptively per image; we stopped at level 2 because the sub-images became too sparse beyond it, but a criterion that decides the depth from the image itself would be preferable. A third limitation is that the evaluation reports operating-point metrics only; a threshold-independent ROC analysis would require a test partition large enough for the curve to be estimated reliably, which the present datasets do not provide. In future work, we intend to extend the model to other medical imaging domains and to evaluate its generalizability across more diverse datasets. We also intend to examine the feature vectors produced by the individual streams using nonlinear embedding methods such as *t*-SNE [29–31], which would make it possible to see how class structure emerges across the levels of the partition tree and to test directly whether an embedding of the stream features separates the classes better than the partition itself.

Acknowledgment

The authors would like to thank the anonymous referees for their support and input.

Funding Support

This work has been conducted in the absence of any funding.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The dataset used in this study by the authors is publicly available and can be accessed at https://www.kaggle.com/datasets/nih-chest-xrays/data. Also, another data source used in this study is available at https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

Author Contribution Statement

Talib Iqball: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Umar Bashir:** Validation, Writing – review & editing, Visualization. **M. Arif Wani:** Conceptualization, Methodology, Software, Resources, Writing – review & editing,

Supervision, Project administration. **Sheikh Umar Mushtaq:** Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Project administration.

References

[1] Ishwerlal, R. D., Agarwal, R., & Sujatha, K. S. (2024). Lung disease classification using chest X-ray image: An optimal ensemble of classification with hybrid training. *Biomedical Signal Processing and Control*, *91*, 105941. https://doi.org/10.1016/j.bspc.2023.105941

[2] Matsuo, H., Nishio, M., Fujimoto, K., Deperrois, N., Matsunaga, T., Nooralahzadeh, F., . . . , & Murakami, T. (2026). Artificial intelligence for chest radiography: An overview of techniques, challenges, and future directions. *npj Health Systems*, *3*(1), 41. https://doi.org/10.1038/s44401-026-00087-y

[3] Cansiz, B., Kilinc, C. U., & Serbes, G. (2025). Deep learning-driven feature engineering for lung disease classification through electrical impedance tomography imaging. *Biomedical Signal Processing and Control*, *100*, 107124. https://doi.org/10.1016/j.bspc.2024.107124

[4] Khodaei, A., Mozaffari-Tazehkand, B., & Sharifi, H. (2025). A genomic signal processing approach for identification and classification of coronavirus sequences. *Soft Computing*, *29*(1), 321–338. https://doi.org/10.1007/s00500-024-10377-y

[5] Kar, S., & Ganguly, M. (2024). Application of genomic signal processing as a tool for high-performance classification of SARS-CoV-2 variants: A machine learning-based approach. *Soft Computing*, *28*(4), 2891–2918. https://doi.org/10.1007/s00500-023-09577-9

[6] Truong, T.-D., & Do, T.-N. (2025). Multimodal approach for lung disease classification: Fusing chest X-ray images and clinical texts. *SN Computer Science*, *6*(6), 607. https://doi.org/10.1007/s42979-025-04138-3

[7] Dediu, V., & Udrea, A. (2026). Segmentation-aware multi-label chest X-ray classification and visual explainability analysis. *Journal of Control Engineering and Applied Informatics*, *28*(2), 24–36. https://doi.org/10.61416/ceai.v28i2.9953

[8] Çallı, E., Sogancioglu, E., Van Ginneken, B., Van Leeuwen, K. G., & Murphy, K. (2021). Deep learning for chest X-ray analysis: A survey. *Medical Image Analysis*, *72*, 102125. https://doi.org/10.1016/j.media.2021.102125

[9] Shamrat, F. J. M., Azam, S., Karim, A., Ahmed, K., Bui, F. M., & De Boer, F. (2023). High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. *Computers in Biology and Medicine*, *155*, 106646. https://doi.org/10.1016/j.compbiomed.2023.106646

[10] Agrawal, T., & Choudhary, P. (2023). Segmentation and classification on chest radiography: A systematic survey. *The Visual Computer*, *39*(3), 875–913. https://doi.org/10.1007/s00371-021-02352-7

[11] Yadav, V. K., & Singhai, J. (2025). Improved multiclass lung disease classification using segmentation and deep learning from chest X-ray images. *IETE Technical Review*, *42*(3), 318–331. https://doi.org/10.1080/02564602.2025.2501936

[12] Fu, X., Lin, R., Du, W., Tavares, A., & Liang, Y. (2025). Explainable hybrid transformer for multi-classification of lung disease using chest X-rays. *Scientific Reports*, *15*(1), 6650. https://doi.org/10.1038/s41598-025-90607-x

[13] Munir, K., Tanveer, M. U., Alyamani, H. J., Bermak, A., & Rehman, A. U. (2025). Pneux-net: An enhanced feature extraction and transformation approach for pneumonia detection in X-ray images. *IEEE Access*, *13*, 84024–84037. https://doi.org/10.1109/ACCESS.2025.3568885

[14] Alshanketi, F., Alharbi, A., Kuruvilla, M., Mahzoon, V., Siddiqui, S. T., Rana, N., & Tahir, A. (2025). Pneumonia detection from chest X-ray images using deep learning and transfer learning for imbalanced datasets. *Journal of Imaging Informatics in Medicine*, *38*(4), 2021–2040. https://doi.org/10.1007/s10278-024-01334-0

[15] Asswin, C. R., KS, D. K., Dora, A., Ravi, V., Sowmya, V., Gopalakrishnan, E. A., & Soman, K. P. (2023). Transfer learning approach for pediatric pneumonia diagnosis using channel attention deep CNN architectures. *Engineering Applications of Artificial Intelligence*, *123*, 106416. https://doi.org/10.1016/j.engappai.2023.106416

[16] Iqball, T., & Wani, M. A. (2023). Weighted ensemble model for image classification. *International Journal of Information Technology*, *15*(2), 557–564. https://doi.org/10.1007/s41870-022-01149-8

[17] Yunianta, A. (2026). A hybrid ensemble deep learning framework for pediatric pneumonia classification using transfer learning and convolutional neural networks. *Mathematical and Computational Applications*, *31*(3), 96. https://doi.org/10.3390/mca31030096

[18] Xu, Y., Souza, L. F., Silva, I. C., Marques, A. G., Silva, F. H., Nunes, V. X., . . . , & Rebouças Filho, P. P. (2021). A soft computing automatic based in deep learning with use of fine-tuning for pulmonary segmentation in computed tomography images. *Applied Soft Computing*, *112*, 107810. https://doi.org/10.1016/j.asoc.2021.107810

[19] Zhang, X., Jiang, R., Huang, P., Wang, T., Hu, M., Scarsbrook, A. F., & Frangi, A. F. (2022). Dynamic feature learning for COVID-19 segmentation and classification. *Computers in Biology and Medicine*, *150*, 106136. https://doi.org/10.1016/j.compbiomed.2022.106136

[20] Goyal, S., & Singh, R. (2023). Detection and classification of lung diseases for pneumonia and COVID-19 using machine and deep learning techniques. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 3239–3259. https://doi.org/10.1007/s12652-021-03464-7

[21] Zaidi, S. Z. Y., Akram, M. U., Jameel, A., & Alghamdi, N. S. (2021). Lung segmentation-based pulmonary disease classification using deep neural networks. *IEEE Access*, *9*, 125202–125214. https://doi.org/10.1109/ACCESS.2021.3110904

[22] Crasta, L. J., Neema, R., & Pais, A. R. (2024). A novel deep learning architecture for lung cancer detection and diagnosis from computed tomography image analysis. *Healthcare Analytics*, *5*, 100316. https://doi.org/10.1016/j.health.2024.100316

[23] Jannat, M., Birahim, S. A., Hasan, M. A., Roy, T., Sultana, L., Sarker, H., . . . , & Abdallah, H. A. (2025). Lung segmentation with lightweight convolutional attention residual U-Net. *Diagnostics*, *15*(7), 854. https://doi.org/10.3390/diagnostics15070854

[24] Zhang, P., & Chen, X. (2025). An unsupervised learning approach for coal spontaneous combustion warning level classification using t-SNE and k-means clustering. *Applied Sciences*, *15*(7), 3756. https://doi.org/10.3390/app15073756

[25] Di Folco, M., Moceri, P., Clarysse, P., & Duchateau, N. (2022). Characterizing interactions between cardiac shape and deformation by non-linear manifold learning. *Medical*

Image Analysis, *75*, 102278. https://doi.org/10.1016/j.media.2021.102278

[26] Kwon, H., Ali, Z. A., & Wong, B. M. (2023). Harnessing semi-supervised machine learning to automatically predict bioactivities of per- and polyfluoroalkyl substances (PFASs). *Environmental Science & Technology Letters*, *10*(11), 1017–1022. https://doi.org/10.1021/acs.estlett.2c00530

[27] Sreevalsan-Nair, J. (2023). Fuzzy C-means clustering. In B. S. Daya. Sagar, Q. Cheng, J. McKinley, & F. Agterberg (Eds.), *Encyclopedia of mathematical geosciences* (pp. 447–449). Springer International Publishing. https://doi.org/10.1007/978-3-030-26050-7_171-1

[28] Liu, H., Chen, J., Dy, J., & Fu, Y. (2023). Transforming complex problems into K-means solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(7), 9149–9168. https://doi.org/10.1109/TPAMI.2023.3237667

[29] Gupta, A., Gupta, S., & Katarya, R. (2021). InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using chest X-ray. *Applied Soft Computing*, *99*, 106859. https://doi.org/10.1016/j.asoc.2020.106859

[30] Iqball, T., & Wani, M. A. (2022). Generative adversarial networks for data augmentation in X-ray medical imaging. In R. Razavi-Far, A. Ruiz-Garcia, V. Palade, & J. Schmidhuber (Eds.), *Generative adversarial learning: Architectures and applications* (pp. 341–355). Springer International Publishing. https://doi.org/10.1007/978-3-030-91390-8_14

[31] Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., . . . , & Islam, M. T. (2020). Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, *8*, 132665–132676. https://doi.org/10.1109/ACCESS.2020.3010287

[32] Maguolo, G., & Nanni, L. (2021). A critic evaluation of methods for COVID-19 automatic detection from X-ray images. *Information Fusion*, *76*, 1–7. https://doi.org/10.1016/j.inffus.2021.04.008

[33] Rohilla, A., Hooda, R., & Mittal, A. (2017). TB detection in chest radiograph using deep learning architecture. In *ICETETSM-17*, 136–147. https://api.semanticscholar.org/CorpusID:212555478

[34] Hwang, S., Kim, H. E., Jeong, J., & Kim, H. J. (2016). A novel approach for tuberculosis screening based on deep convolutional neural networks. In *Medical Imaging 2016: Computer-Aided Diagnosis, 9785, 97852W*. https://doi.org/10.1117/12.2216198

[35] Lopes, U. K., & Valiati, J. F. (2017). Pre-trained convolutional neural networks as feature extractors for tuberculosis detection. *Computers in Biology and Medicine*, *89*, 135–143. https://doi.org/10.1016/j.compbiomed.2017.08.001

[36] Nafisah, S. I., & Muhammad, G. (2024). Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence. *Neural Computing and Applications*, *36*(1), 111–131. https://doi.org/10.1007/s00521-022-07258-6

[37] Islam, M. T., Aowal, M. A., Minhaz, A. T., & Ashraf, K. (2017). Abnormality detection and localization in chest X-rays using deep convolutional neural networks. *arXiv Preprint: 1705.09850*.

[38] Zak, M., & Krzyżak, A. (2020). Classification of lung diseases using deep learning models. In *International Conference on Computational Science*, *12139*, 621–634. https://doi.org/10.1007/978-3-030-50420-5_47

How to Cite: Iqball, T., Bashir, U., Arif Wani, M., & Mushtaq, S. U. (2026). Recursive Pixel Partitioning and Multi-Stream Deep Learning for Annotation-Efficient Lung Disease Classification. *Medinformatics*. https://doi.org/10.47852/bonviewMEDIN620210440