

Multi-Classification of Lung Diseases from X-Ray Images based on Self Attention Convolutional Neural Network

Jie Zhu^{*,1} and Xinyao Ma¹

¹College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321004, China.

Abstract Currently, lung diseases are increasing, causing millions of infections and deaths annually. Early detection is critical, but misdiagnosis can occur due to factors like physician fatigue. Thus, machine learning techniques have gained prominence in healthcare as alternative or complementary tools for analyzing medical data because of their ability to assist in accurate and efficient diagnosis. Although many studies focus on X-ray-based lung disease models, most methods are binary classification methods (for example, COVID-19 vs. normal). Therefore, in this paper, we apply a multi-classification approach based on a convolutional neural network (CNN) model with a self-attention mechanism. This model can classify six classes: tuberculosis, pneumonia, COVID-19, opacity, pneumothorax, and normal X-ray images. A total of 45,241 lung X-ray images from multiple databases were collected to evaluate the model. The proposed model is compared with several deep learning models to verify its performance. The CNN model with a self-attention mechanism can classify the six types of lung X-ray images with a high detection accuracy.

Keywords: Lung disease, Machine learning, X-ray image, Deep learning, Self-Attention, CNN

1 Introduction

Lung diseases have always affected our human health in a variety of ways and are one of the leading causes of death around the world. Moreover, it has been proven that these diseases (lung nodules, pneumonia, chronic bronchitis, etc.) are consistently associated with an increased risk of cancer, even among smokers [1]. At the end of 2019, COVID-19 broke out, resulting in significant global mortality. Even if COVID-19 is no longer endemic, more severe respiratory diseases remain a critical concern. So urgent and accurate methods are needed to detect lung diseases promptly. Early diagnosis and detection can greatly reduce the risk of life-threatening lung diseases so that they can be prevented before they happen. According to the World Health Organization (WHO), an estimated

*Corresponding author: J. Zhu. Email: 2022207001230@zjnu.edu.cn

3.2 million people die each year from diseases linked to domestic air pollution, such as asthma and pneumonia. Therefore, it is crucial to adopt efficient diagnostic methods to assist in the early detection of lung lesions [2]. In today's medical diagnosis, many reasons can lead to a doctor's misdiagnosis (such as the doctor's inexperience, the doctor's fatigue level, subtle or minor changes in some diseases, and other reasons) [3, 4]. Before there was an automated diagnostic model, the Centers for Disease Control reported that a single pneumonia alone was responsible for more than 50,000 deaths in the United States alone. Pneumonia is also one of the leading causes of death in children, and according to a 2019 report by the WHO, pneumonia is the cause of more than 800,000 child deaths worldwide. Thus, pneumonia kills more children than any other infection. According to UNICEF, about 2,200 children died in just one day [5]. A projection X-ray is used to capture subtle lesions, soft tissues, and varying densities within the human body. Generally, X-rays are used to observe lung tissue diseases, the chest cavity, and the outline of the lungs and heart. X-ray scans can not only diagnose common chest diseases but also determine cardiac hypertrophy, spinal lesions, lung abnormalities, etc. It is a medical method to detect the diseased part in time and participate in the treatment as early as possible [6]. In recent years, with the rise of machine learning, including deep learning, lung disease detection has significantly benefited from these advances [7, 8]. However, X-ray images are completely different from natural images. Medical images are not easy to obtain because of privacy and other reasons. Moreover, the high similarity in the X-ray images poses significant challenges for training neural networks.

Numerous studies have incorporated traditional machine learning (ML) techniques within their classification models to tackle various challenges [9]. Among these methods, the support vector machine (SVM) has been notably employed within the layers of neural networks to detect lung cancer and reduce processing complexity [10]. This detection was enhanced in the second stage, which used the SURF technique with a genetic algorithm to extract the features. In [11], the VGG16 network was combined with SVM to enhance the classification results. To solve the low-resolution problem extracted by the neural network layers, the CNN and KNN were combined to detect COVID-19 in the CT scan images in [12]. In the field of deep learning (DL), many studies used VGG16 [13–15], VGG19 [16–18], ResNet [16, 19–21] to extract the features from X-ray images.

Inspired by the successful applications of conventional neural networks (CNNs) in medical image processing, we apply a light model to classify lung disease from X-ray images. Many of the previous studies were proposed for binary classification. Thus, unlike many existing models, we apply a light CNN model for multiple classifications with comparative classification performance. The advantages of the proposed model are that it only has a 30-layer network structure and uses separable convolution to reduce the computation complexity significantly. The proposed model is examined with many X-ray images, and it achieves excellent results based on several indicators, such as accuracy, recall, and F1-score values. In addition to the advantages of the proposed model, we collect many X-ray images from many existing resources to create a large X-ray image database with about 45,241 images.

The main contributions of this paper can be described as follows:

- Apply a lightweight network for lung disease multi-classification process, which includes six lung X-ray images (five different disease and normal images), which may be used to facilitate the accurate diagnosis of lung diseases in some countries and regions with medical shortages.

- Employ an attention mechanism integrated with CNN to mitigate the model's overfitting, enhancing its ability to classify lung diseases from X-ray images accurately. This attention mechanism focuses on the most relevant features in the medical images, ensuring the model generalizes unseen data better and enhances its diagnostic accuracy.
- Implement extensive experiments based on different evaluation indicators using public datasets collected from Kaggle and GitHub to verify the performance of the presented model, which outperformed several existing models.

The structure of this paper is organized as follows: Section 2 reviews the contributions of several previous works. In Section 3, we provide an overview of general CNNs. Section 4 presents the proposed model and discusses the parameter tuning for the network architecture. Section 5 describes the evaluation experiments, including comparison experiments to assess the effectiveness of the proposed model architecture. Finally, Section 6 presents the conclusion.

2 Literature review

In [22], a CNN model employing hybrid deep learning that combines heading and base models was developed to improve feature extraction. VGG16 and VGG19, two pre-trained deep learning structures, were incorporated into the base model. The heading section comprised a dropout layer and dense layers of size three with distinct activation functions. To help doctors make better decisions, the authors in [23] suggested a hybrid bidirectional LSTM (BiDLSTM)-Mask Region-Based CNN (Mask-RCNN) lung disease prediction system. Crystal improves Mask-RCNN model scalability and convergence by fine-tuning hyperparameters. The BiDLSTM architecture linked to the Mask RCNN model's fully connected layer predicted long-range lung disease relationships for diverse datasets. Additionally, a framework suggested in [24] used three chicken swarm optimization (CSO)-built DL multiple neural networks (DLMNNs). Farhan et al. [25] introduced a hybrid deep learning algorithm (HDLA) architecture that utilizes X-ray images for automatic feature extraction, detection, and preprocessing. BioBert semantic vectors were coupled with visual vectors obtained from convNeXt network fine-tuning to transfer the two categories of features into a unified metric space in [26]. A prototype of semantic vectors was developed for every metric space class. Following this, the image-label metric relationship is investigated at the X-ray image and illness category levels, and an innovative dual-weighted metric loss function is constructed. A Multi-task Supervised Contrastive Learning (MTSCL) framework was presented in [27] to learn robust features from the target dataset. It used two hierarchical categorization steps. The MTSCL model detected lung anomalies in the first stage, and in the second stage, tree-based algorithms identified these as pneumonia or COVID-19. The second stage took advantage of image preprocessing with lung masks and MTSCL model-generated lesion masks, which improved framework results. Moreover, in [28], Bayesian counterparts of two distinct off-the-shelf CNNs (DenseNet-121 and EfficientNet-B6) were utilized to provide new methodological indications that can strengthen the classification extraction. Ong and Cheng [29] proposed a network integrating the DenseNet network and VGG16 network for X-ray image classification. They added an attention mechanism to the model to extract depth features. The dataset was derived from chest X-rays taken from pediatric patients aged 1 to 5 years at Guangzhou Women and Children's Medical Center [30]. In

the pre-processing of the dataset, the ResNet network was used for semantic segmentation first, and all images were uniformly adjusted to 512×512 in the second step. The third step is to transform the image data into a 255-level gray value image by enhancing the lung's shadow area. The last step is to remove the redundant part. Then, they fed it into the network they had developed. The experimental results showed that binary and triple classification accuracy has achieved good accuracy rates. Noasir et al. [31] presented a multimodal image classification of COVID-19 forces in X-ray using computed tomography. Fang and wang [32] analyzed COVID-19 with mixed CT and X-ray data, enhanced feature extraction with adaptive algorithms, and improved the accuracy of the model with residual blocks and dense blocks, which could also effectively improve the robustness of the proposed model. The results of the experiment were good and achieved high detection rates. Hassantabar et al. [33] used CNNs to detect and diagnose COVID-19 disease in X-rays. The author proposed two methods: the DNN model based on data fractal features and the CNN model based on lung images. The results show that the accuracy of the CNN architecture was 93.2%, which was much higher than that of DNN technology, which was 83.4%.

3 Preliminaries

3.1 CNN Neural Network

With the rapid advancement of deep learning, neural networks, particularly CNNs, have demonstrated significant success across various domains. Neural networks are designed to simulate the function of the human brain by simulating interconnected neurons, forming a computational model inspired by the biological nervous system. In this model, input data is transformed into a feature matrix, which is then processed by the network. The network learns to identify patterns and make decisions based on these features through an adaptive process that optimizes its parameters during training. Ultimately, the trained model is capable of performing complex tasks by simulating the behavior of biological neurons in processing and interpreting the input data.

When a value is transmitted via a link, it is scaled by the link's associated weight. For a given input X , the value arriving at the neuron is computed as $W \times X$, where W is the weight matrix. The neural network adapts by adjusting these weights during training. The term b represents a special parameter known as the bias, which is added to the neuron's input. The bias can be considered as the weight of a constant input of 1, allowing the neuron to shift its activation independently of its input. This makes the total input to the neuron $W \times X + b$, where the bias ensures that the neuron can produce a non-zero output even when the input is zero.

As illustrated in Figure 1, y represents the final output value of the neuron. The operation sums the weighted inputs received by the neuron, and the result is passed through an activation function. The neuron's activation depends on whether the weighted sum exceeds a certain threshold θ . The neuron will produce an output if the input exceeds this threshold. The relationship is mathematically defined in Eq. (1):

$$y = W \times X + b \quad (1)$$

In deep learning, convolutional neural networks (CNNs) play a central role that is distinct from traditional fully connected neural networks. A CNN typically comprises several key layers, including the input, convolutional, pooling, and output layers. The

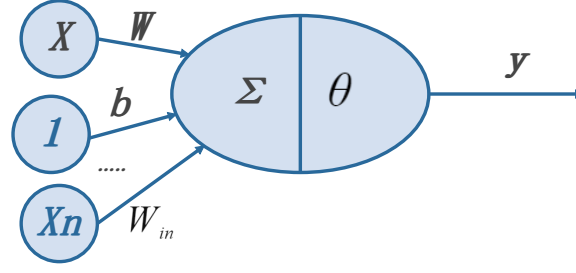


Figure 1: Neuron activation formula.

convolutional and pooling layers serve as the feature extractor, which is critical for identifying patterns in the data. The convolutional layer consists of neurons that are locally connected to their neighboring neurons, forming multiple feature maps. Each feature map is shared by neurons arranged in a grid-like structure. This weight-sharing mechanism significantly reduces the number of parameters, thereby minimizing the risk of overfitting and improving the network's generalization ability. Additionally, pooling layers further reduce the spatial dimensions of the data, aiding in computational efficiency and preventing overfitting.

3.1.1 Convolutional Layer

The convolutional layer is a fundamental component of CNNs and is responsible for feature extraction from input images or data. This layer consists of multiple convolutional kernels (also referred to as filters), each designed to extract specific features. Each convolutional kernel has its own set of parameters, including weights and biases, which are adjusted during the learning process. The parameters of the convolutional layer include the kernel size, stride, and padding, which collectively determine the size of the output feature map. The primary function of the convolutional layer is to perform localized operations on the input data by moving the filter across the image with a specified stride. As the filter moves horizontally and vertically across the image, it detects various features such as edges, textures, and other patterns. These extracted features form the output feature maps, which capture increasingly complex representations of the input data as the network deepens. Convolutional layers are particularly effective at detecting higher-level features, such as shapes and objects, by applying multiple filters that specialize in different tasks like edge detection, sharpening, or blurring. This process aids in understanding the structural and contextual information within the image.

Assuming the input image has dimensions $W1 \times H1 \times D1$, four hyperparameters must be set for the convolutional layer: the number of filters (K), stride (S), filter size (F), and padding (P). The output dimensions of the convolutional layer, denoted as $W2 \times H2 \times D2$, can be computed using the following formula:

$$W2 = \frac{W1 - F + 2P}{S} + 1 \quad (2)$$

$$H2 = \frac{H1 - F + 2P}{S} + 1 \quad (3)$$

$$D2 = D1 = K \quad (4)$$

These equations define the dimensions of the output feature map after applying convolution, where $W2$ and $H2$ represent the spatial dimensions, and $D2$ denotes the number of feature maps (determined by the number of filters K).

3.1.2 Pooling Layer

The pooling layer performs downsampling operations, typically using either max pooling or average pooling. Its primary functions are as follows:

- **Feature dimension reduction:** Pooling reduces the spatial dimensions of the input, thereby decreasing the size of the model. This reduces the computational cost and the number of parameters in the subsequent layers.
- **Invariance:** Pooling layers provide translation invariance, and to some extent, scale and rotation invariance, by preserving the most prominent features regardless of minor variations in the input.
- **Increase the receptive field:** Pooling increases the receptive field of the following layers, as it reduces the resolution of the feature maps while retaining essential information. This allows the network to capture more global features as it progresses through deeper layers.
- **Reduce overfitting:** By reducing the feature dimension, pooling lowers the number of parameters and computation, which helps in mitigating overfitting while retaining the most useful information.

Pooling layers are used repeatedly in convolutional networks to progressively reduce the spatial size of the feature maps, making the network more computationally efficient and less prone to overfitting. The most common type of pooling is max pooling, though other variants such as average pooling and sum pooling are also used. Max pooling operates with two key parameters: filter size (F) and stride length (S).

Assuming the input feature map has dimensions $W1 \times H1 \times D1$, the output dimensions after pooling, denoted as $W2 \times H2 \times D2$, can be computed using the following equations:

$$W2 = \frac{W1 - F}{S} + 1 \quad (5)$$

$$H2 = \frac{H1 - F}{S} + 1 \quad (6)$$

$$D2 = D1 \quad (7)$$

Here, $W2$ and $H2$ represent the spatial dimensions of the output feature map, and $D2$ remains the same as the input depth $D1$, as pooling operates independently across each feature map. Therefore, convolution layer and pooling layer are usually used together, as shown in Figure 2.

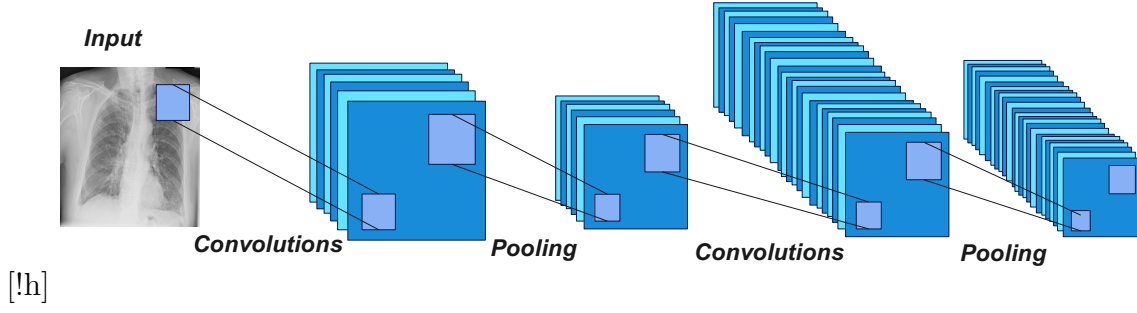


Figure 2: Application between convolutional networks and pooling layers

3.1.3 Activation Function

In deep learning, particularly in neural networks, activation functions are applied after the output of each layer (e.g., Sigmoid, Tanh, ReLU, etc.) to introduce non-linearity into the model. Their primary function is to transform the output of a neuron and determine whether to pass the signal to the next layer. Activation functions allow neural networks to learn complex patterns in data by mimicking the behavior of biological neurons, where a neuron fires if the input exceeds a certain threshold.

Activation functions can be categorized into linear and nonlinear functions. Common nonlinear activation functions include Sigmoid, Tanh, ReLU, and Swish. Among these, the ReLU (Rectified Linear Unit) function is widely used in CNN layers due to its simplicity and efficiency in mitigating the vanishing gradient problem and reducing computational complexity. The ReLU function is defined as:

$$\text{ReLU}(x) = \max(0, x) \quad (8)$$

The softmax activation function is used in the final layer of the proposed network to normalize the output of the previous layer, ensuring the values are within the range (0,1). Softmax is particularly useful for handling multi-class classification problems by converting logits into probabilities. It is commonly used in conjunction with a cross-entropy loss function for training deep networks in supervised learning. The softmax function is defined as:

$$P(y|x) = \frac{e^{h(x, y_i)}}{\sum_{j=1}^n e^{h(x, y_j)}} \quad (9)$$

Here, $P(y|x)$ represents the probability of class y_i given input x , and $h(x, y_i)$ is the logit for class y_i . Softmax is typically applied to the output layer in neural networks for multi-class classification.

3.1.4 Flatten Layer

In the proposed model, after the pooling layer, a flatten layer is applied to convert the multidimensional feature maps into a one-dimensional vector. This transformation prepares the data for the fully connected layers that follow. In this model, the flatten layer is followed by four fully connected layers, where activation functions are used to classify the output into different categories. The main function of the flatten layer is to reshape the multidimensional data into a single column vector, allowing the entire feature map to be passed to the fully connected layers for further processing and classification.

3.1.5 Fully Connected Layer

The fully connected (FC) layer is typically placed at the end of a CNN. Its role is to synthesize and integrate the features extracted by the previous layers and use them for classification. In the fully connected layer, each node is connected to all nodes in the preceding layer, allowing the model to combine all the extracted features for final classification. The fully connected layer usually contains the most parameters in the network, making it crucial for determining the final output. It also functions as a classifier, where the positional dependencies of features are reduced, improving the network's robustness. In the proposed model, after the flatten layer, four fully connected layers are used. By applying activation functions, the output is classified into different categories.

3.1.6 Batch Normalization Layer

Batch normalization is a widely used technique in deep learning that improves both the accuracy and training efficiency of neural networks. The primary function of batch normalization is to normalize the input of each batch by ensuring that each feature has a mean of 0 and a variance of 1. This normalization makes the training process more stable and accelerates convergence. The process of batch normalization involves calculating the mean and variance of the batch data, subtracting the mean, and dividing by the standard deviation to normalize the data. After normalization, the data is scaled and shifted using learnable parameters—a scaling factor and an offset. This allows the network to retain the ability to represent complex relationships while benefiting from normalized input distributions. The primary roles of batch normalization are as follows: (1) it accelerates model training; (2) it improves the accuracy and generalization of the model; and (3) it reduces the model's reliance on careful initialization of weights.

3.2 Depth-Separable Convolutional Networks

Depth-separable convolution is an efficient form of convolution operation used in CNNs. It splits the standard convolution into two smaller, separate operations: depthwise convolution (also called channel-by-channel convolution) and pointwise convolution (also known as 1×1 convolution). In depthwise convolution, each input channel is convolved independently with a separate filter, generating one feature map per channel. However, this operation does not combine information across different channels, so it is usually followed by pointwise convolution. In pointwise convolution, a 1×1 filter is applied across all the channels to combine the feature maps generated by the depthwise convolution. Together, these two operations approximate the standard convolution but with significantly fewer parameters and computational cost.

The advantages of depth-separable convolution are as follows:

1. It reduces the number of parameters by decomposing the standard convolution into depthwise and pointwise operations, which require fewer filters.
2. The computational cost is lower compared to traditional convolution, as the depthwise and pointwise convolutions are performed separately, each with reduced complexity.
3. It produces a smaller model: Since depth-separable convolution has fewer parameters and a lower computational cost than standard convolution, it results in a lighter

and more efficient model, which is especially beneficial for lightweight neural networks.

4 Materials and methods

In this section, we describe the proposed multi-classification model for lung disease detection from X-ray images using a CNN architecture based on the Self-Attention Mechanism-6 (SAM-6). This network model consists of 2 convolutional layers, 5 max-pooling layers, 8 depthwise separable convolution layers, 1 self-attention mechanism, 1 flatten layer, 4 dense layers, 5 regularization layers, and 4 batch normalization layers. The workflow of the proposed model is shown in Figure 3.

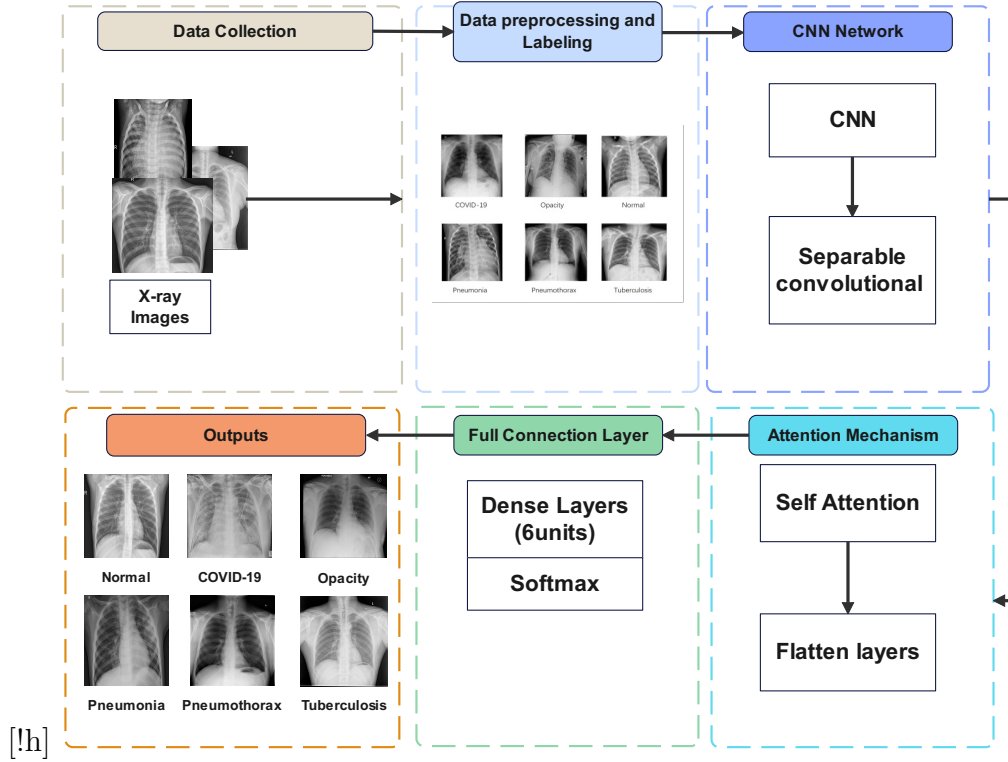


Figure 3: Flowchart of the proposed lung diseases classification model.

For classification, the input images are preprocessed to a size of 224×224 pixels. Feature extraction begins with two convolutional layers (3×3 kernel size, 16 filters) followed by a MaxPooling layer. As discussed in Section 4, convolutional layers are typically used in conjunction with pooling layers to extract features, reduce model size, and mitigate overfitting. After the first pooling layer, the data is passed into a deep separable convolutional network, starting with two depthwise separable convolutional layers (3×3 kernel size, 32 filters).

Batch normalization is applied to both the second and third depthwise separable convolutional layers, positioned before the MaxPooling layer. The parameters remain the same between the second and third layers, except for the number of filters, which doubles from 32 to 64 in the third layer. In the subsequent layers, the number of convolutional filters continues to double with each layer. To prevent overfitting, a regularization layer is added after the depthwise separable convolutional layers in the fourth and fifth layers, along with a self-attention mechanism, where the dropout rate is set to 0.2.

Finally, a fully connected layer is input in the final layers, where the number of neurons is halved in each successive layer. The last layer contains 6 neurons, corresponding to the 6 classification categories. The overall architecture of the proposed network is illustrated in Figure 4.

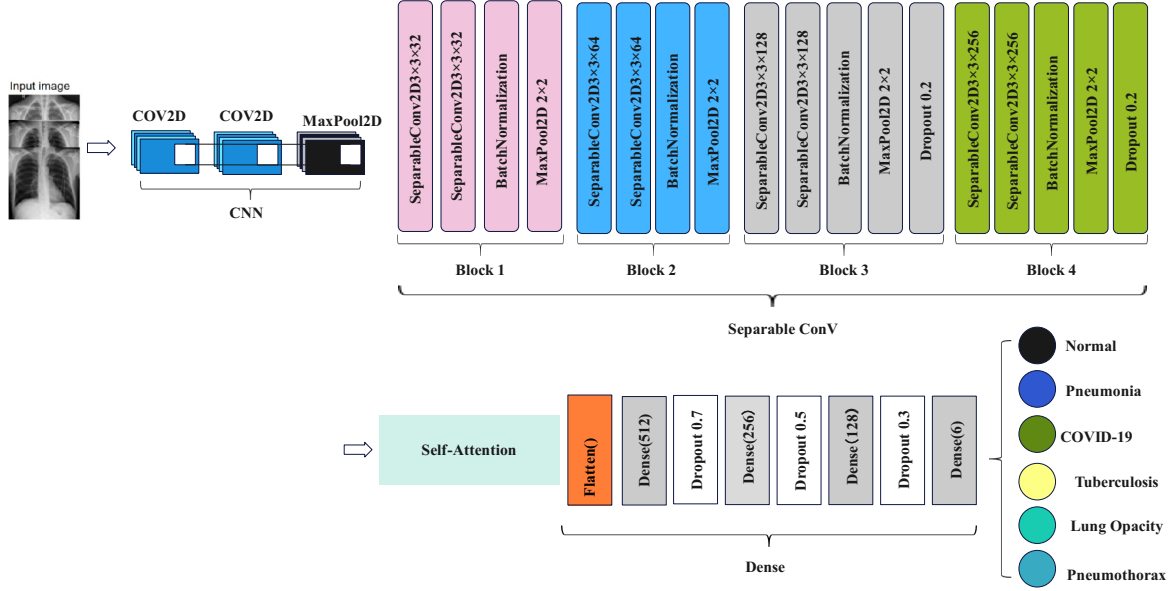


Figure 4: The proposed Self-Attention CNN network model framework.

4.1 Self-Attention Mechanism

Self-attention is one of several attention mechanisms, and since the publication of its original study [2], it has become a foundational component of many deep learning methods for various applications. Given its widespread use, understanding how self-attention works is essential. The self-attention formula is provided in Equation (10):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \times V \quad (10)$$

In this formula, the process begins by taking the dot product of the Query matrix Q and the transpose of the Key matrix K^T , then dividing by $\sqrt{d_k}$, where d_k is the dimensionality of the key vectors. This scaling factor, $\sqrt{d_k}$, prevents the dot product QK^T from becoming too large, which could lead to instability in the softmax calculation. Using $\sqrt{d_k}$ ensures that the result of QK^T follows a distribution with an expected mean, making the softmax operation more stable.

4.2 Setting Hyperparameters

In this study, extensive experiments were conducted to evaluate the effects of various hyperparameters, including batch size, optimizer, loss function, and normalization (Batch Normalization, BN). After thorough testing, the Stochastic Gradient Descent (SGD) optimizer was identified as a key factor in improving performance within the domain of medical deep learning. Additionally, categorical cross-entropy was selected as the loss function due to its superior performance, as detailed in the evaluation section.

The categorical cross-entropy loss function is expressed as follows:

$$loss = - \sum_{i=1}^n (\tilde{y}_{i1} \log y_{i1} + \tilde{y}_{i2} \log y_{i2} + \cdots + \tilde{y}_{im} \log y_{im}) \quad (11)$$

$$\frac{\delta loss}{\delta y_{i1}} = -\frac{\tilde{y}_{i1}}{y_{i1}} \quad (12)$$

In these equations, n represents the number of samples, and m denotes the number of classes. Here, $\tilde{y}$ refers to the true labels, and y represents the predicted labels. Since the loss is computed across multiple outputs, the total loss is a sum over the individual contributions for each class. According to the equations, the weight update speed is influenced by the magnitude of the prediction error.

Moreover, the SGD is a simple yet effective optimizer. It updates model parameters based on the gradient computed from a single or small batch of samples. One of the main advantages of SGD is its suitability for large datasets and complex models. The update rule for SGD is represented by Equation (13):

$$\omega_{t+1} = \omega_t - \eta \nabla_{\omega} L(\omega_t; x_i; y_i) \quad (13)$$

where ω_t is the model parameter at time step t , η is the learning rate, and $\nabla_{\omega} L(\omega_t; x_i; y_i)$ is the gradient of the loss function L with respect to the model parameters ω at data point (x_i, y_i) . The settings of the hyperparameters are summarized in Table 1.

Table 1: Hyperparameter settings.

Hyperparameter	Value
Batch_size	32
Optimizer	SGD
Loss	categorical crossentropy
Epochs	400
Learning rate	0.01

5 Experiments and Results

5.1 Datasets

The datasets of this study were collected from different public platforms, such as [34–41]. The dataset consisted of 45,241 X-ray lung images. There are six types of X-ray images: tuberculosis (4,162), COVID-19 (5,795), pulmonary opacity (6,012), pneumothorax (4,697), pneumonia (9,883), and normal lungs (14,692). Figure 5 shows samples of the X-ray images of the six different categories of five lung diseases and normal images.

The preprocessing is implemented by adjusting the pixel value of the image to adjust the pixel range of the image to be between 0 and 1. The size of all datasets is changed to 224×224 and finally put into the model for training.

5.2 Performance Indicators

For X-ray image preprocessing, the `ImageDataGenerator` in Keras is used to resize and augment the images.

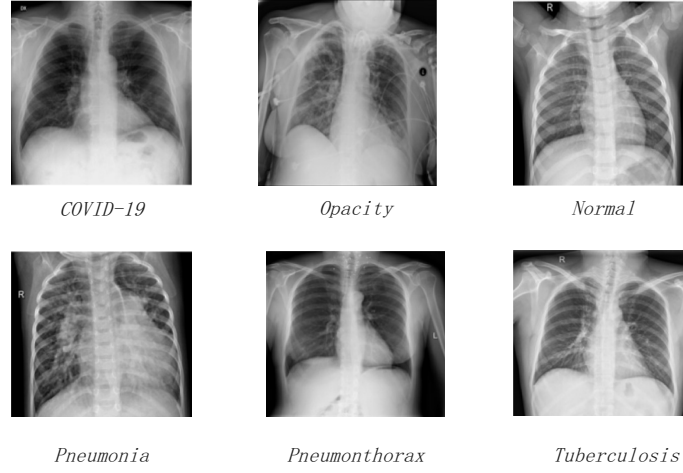


Figure 5: Examples of lung diseases: COVID-19, Opacity, Normal, Pneumonia, Pneumothorax, Tuberculosis

We used the five-fold cross-validation to evaluate the proposed model. The model is trained on 400 epochs using the SGD optimizer and the classification cross-entropy loss function.

To evaluate classification performance, the following metrics from data mining are used:

- True Positive (TP): The model correctly identifies a patient with the condition (the actual class is 1, and the prediction is 1).
- False Positive (FP): A healthy image is incorrectly classified as positive (the actual class is 0, but the prediction is 1).
- True Negative (TN): The model correctly identifies a healthy individual (the actual class is 0, and the prediction is 0).
- False Negative (FN): The model fails to detect the condition in a patient (the actual class is 1, but the prediction is 0).

In this study, we use several performance metrics to evaluate the classification results, including precision, loss, F1-Score, and recall. The formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \quad (16)$$

$$F1 - score = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity} \quad (17)$$

5.3 Results

The proposed model was evaluated using five-fold cross-validation, and the results indicated consistent and robust accuracy across all folds, as listed in Table 4. The average accuracy achieved across the five folds was 94.33%. COVID-19 detection achieved good results, with an average F1-score ranging from 92% to 96% across folds, reflecting reliable classification. At the same time, the Opacity class recorded a high F1-score of 0.90 to 0.95, while Pneumothorax exhibited slight variability but achieved an average F1-score of 0.94. Challenges were observed in the Normal class, with F1-scores lower than other classes, likely due to its similarity to other classes. Additionally, Figure 6 shows the confusion matrix of the Fold-5 results. As we can see from all results, the suggested model performed well in detecting and classifying various lung conditions with high reliability and high detection rates. The high macro and weighted averages (Table 4) further confirm the model’s balanced performance across diverse classes and distributions.

Table 2: The classification results using five-fold cross-validation for folds 1 and 2.

Class	Fold 1				Class	Fold 2			
	Precision	Recall	F1-Score	Support		Precision	Recall	F1-Score	Support
COVID-19	0.90	0.94	0.92	1178	COVID-19	0.97	0.92	0.95	1119
Opacity	0.93	0.95	0.94	2874	Opacity	0.94	0.97	0.95	2949
Normal	0.91	0.80	0.85	1240	Normal	0.88	0.88	0.88	1192
Pneumonia	1.00	0.99	0.99	2010	Pneumonia	0.99	0.99	0.99	1983
Pneumothorax	0.91	0.98	0.94	906	Pneumothorax	0.95	0.95	0.95	934
Tuberculosis	0.98	0.98	0.98	817	Tuberculosis	0.99	0.97	0.98	848
Accuracy			0.94	9025	Accuracy			0.95	9025
macro avg	0.94	0.94	0.94	9025	macro avg	0.95	0.95	0.95	9025
weighted avg	0.94	0.94	0.94	9025	weighted avg	0.95	0.95	0.95	9025

Table 3: The classification results using five-fold cross-validation for folds 4 and 5.

Class	Fold 3				Class	Fold 4			
	Precision	Recall	F1-Score	Support		Precision	Recall	F1-Score	Support
COVID-19	0.92	0.97	0.94	1108	COVID-19	0.93	0.96	0.95	1112
Opacity	0.96	0.85	0.90	3011	Opacity	0.95	0.95	0.95	2937
Normal	0.88	0.87	0.85	1212	Normal	0.87	0.88	0.87	1164
Pneumonia	0.99	0.99	0.99	1955	Pneumonia	1.00	1.00	1.00	2034
Pneumothorax	0.70	0.97	0.81	906	Pneumothorax	0.97	0.92	0.95	962
Tuberculosis	0.99	0.97	0.98	833	Tuberculosis	0.99	0.98	0.98	816
Accuracy			0.92	9025	Accuracy			0.95	9025
macro avg	0.91	0.93	0.91	9025	macro avg	0.95	0.95	0.95	9025
weighted avg	0.93	0.92	0.92	9025	weighted avg	0.95	0.95	0.95	9025

5.4 Visual lesion localization

To provide a rational explanation for the classification results, we use Grad-CAM (Gradient-weighted Class Activation Mapping) to visualize the features extracted by the model and to show the basis for its disease diagnoses. Grad-CAM is a widely used explainability technique in deep learning, particularly in CNNs, that generates visual explanations of model decisions. It is especially useful in image classification tasks as it highlights the regions of an image that the model focuses on when making predictions. By applying Grad-CAM, we can generate visual interpretations of the CNN model’s gradients, helping to address the “black box” problem often encountered in deep learning applications. As shown in Figure 7, the Grad-CAM results clearly indicate that the model accurately

Table 4: The classification results using five-fold cross-validation for fold 5.

Class	Fold 5			
	Precision	Recall	F1-Score	Support
COVID-19	0.96	0.95	0.96	1163
Opacity	0.93	0.96	0.95	2921
Normal	0.90	0.85	0.87	1204
Pneumonia	0.99	0.99	0.99	1901
Pneumothorax	0.97	0.98	0.97	988
Tuberculosis	0.99	0.98	0.98	848
Accuracy			0.95	9025
macro avg	0.95	0.95	0.95	9025
weighted avg	0.95	0.95	0.95	
Average Results for all 5-folds				
	Average Accuracy	Average Precision	Average Recall	Average F1-Score
	0.9433	0.9454	0.9433	0.9435

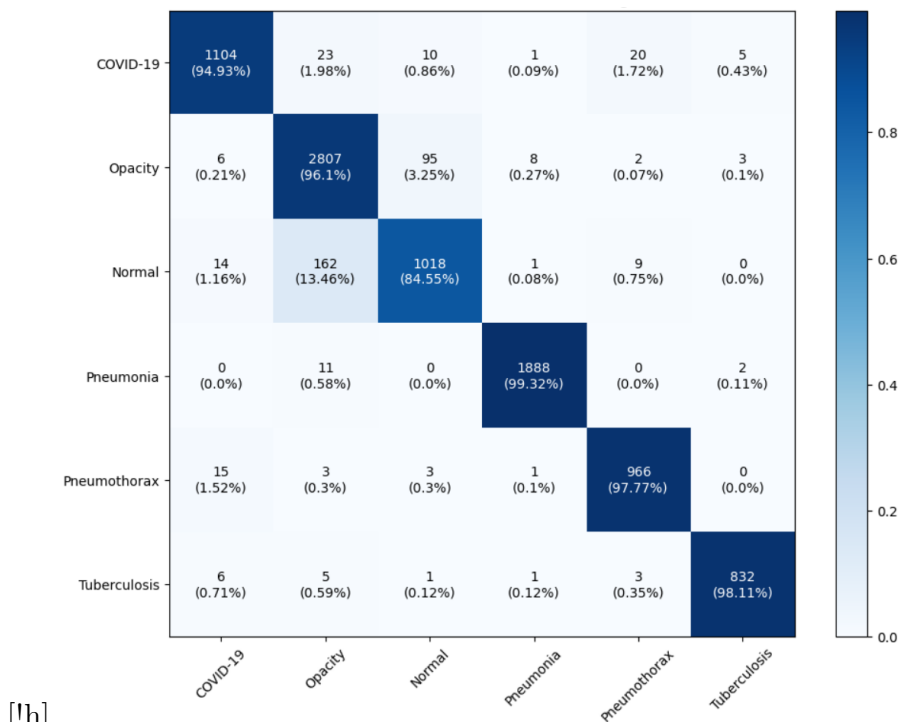


Figure 6: Confusion matrix of the Fold-5.

captures lesion information, enabling reliable disease classification based on the identified features. We further demonstrate the extraction of the most prominent features from our proposed model. In the 15 X-ray images presented, the model achieves perfect classification, with Grad-CAM effectively highlighting the key regions responsible for each diagnosis.

5.5 Comparison with other deep learning structures

To evaluate the effectiveness of the proposed model, we implement comparison experiments to assess the accuracy of the model against other deep learning models usually used in literature. The experiment compared DenseNet121 and VGG16 architectures,

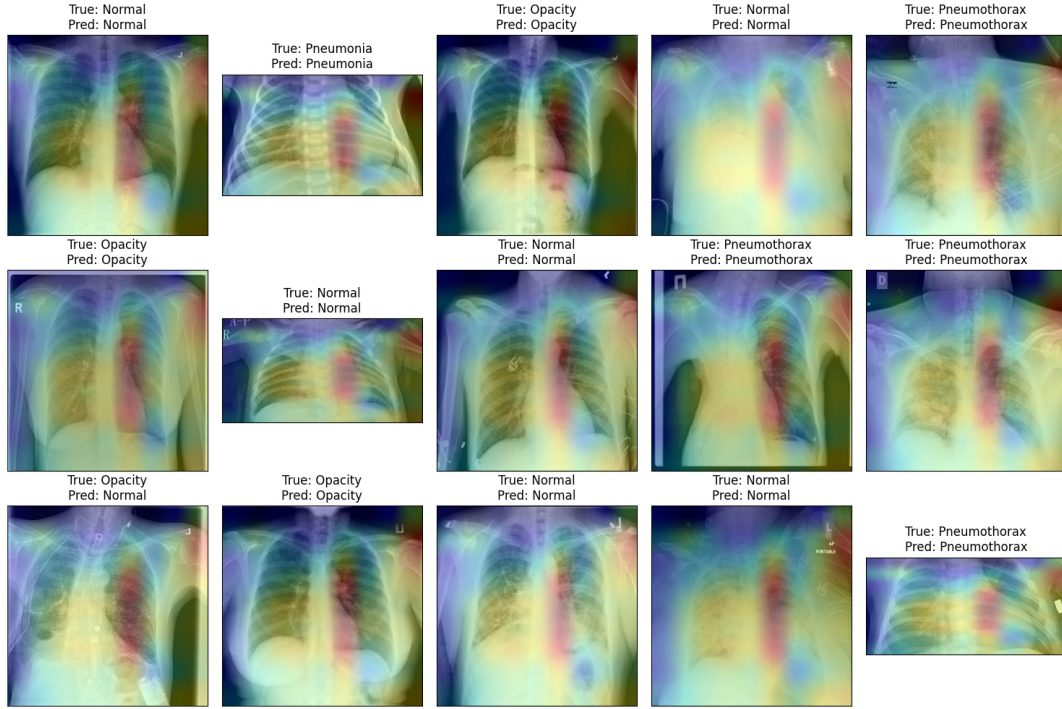


Figure 7: Grad-CAM results presented

each enhanced with the proposed CNN network (composed of convolutional layers and depthwise-separable convolutions) and versions of the CNN network without the self-attention mechanism. Additionally, we compared the complete proposed model with the same model, excluding the attention mechanism. The architecture of each compared model is presented in Figures 8, 9, 10, 11, and 12.

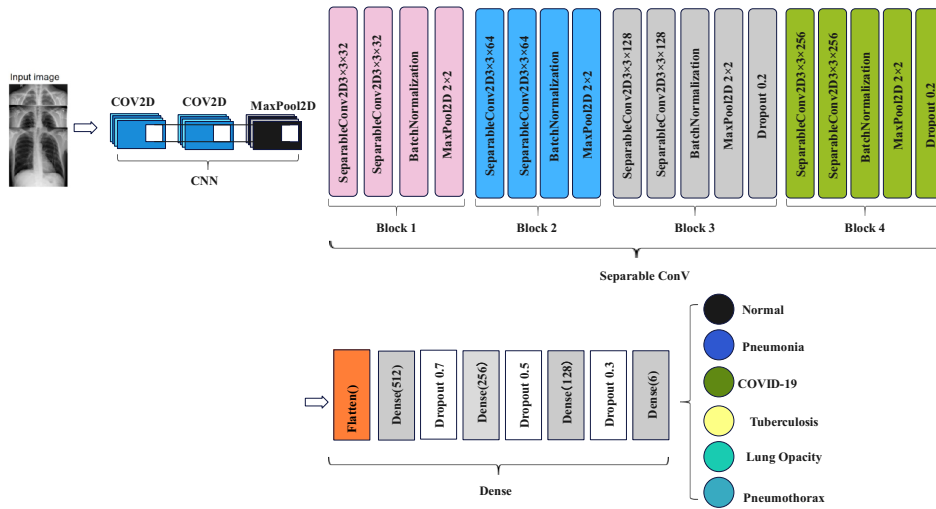


Figure 8: The framework of the CNN network model.

Table 5 presents the comparison results for different models in terms of the Precision, Recall, and F1-score metrics. The proposed model achieved the best results compared to other models. It is also clear that CNN, without attention, obtained the second rank, followed by DenseNet121+CNN, VGG16+CNN, DenseNet121, and VGG16, respectively.

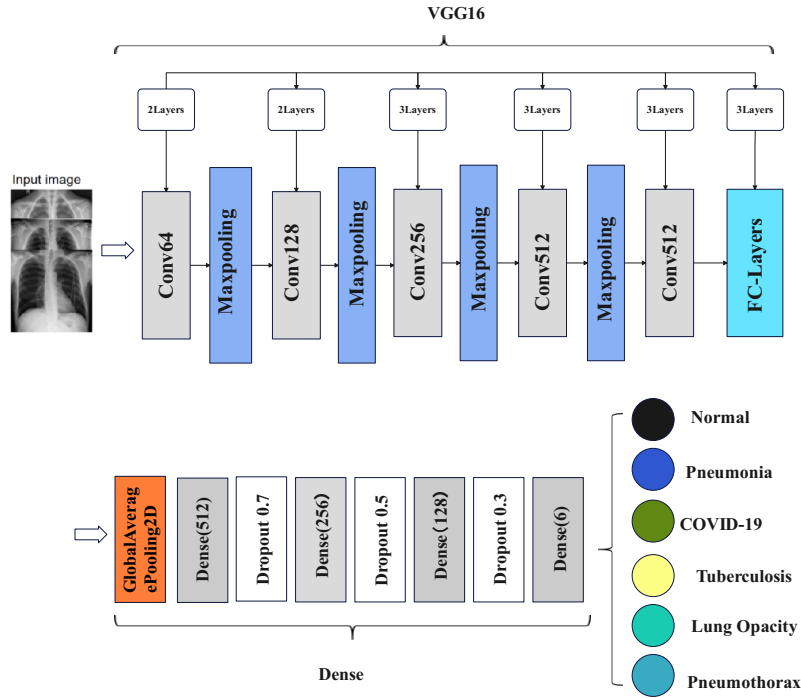


Figure 9: The framework of the VGG16 network model.

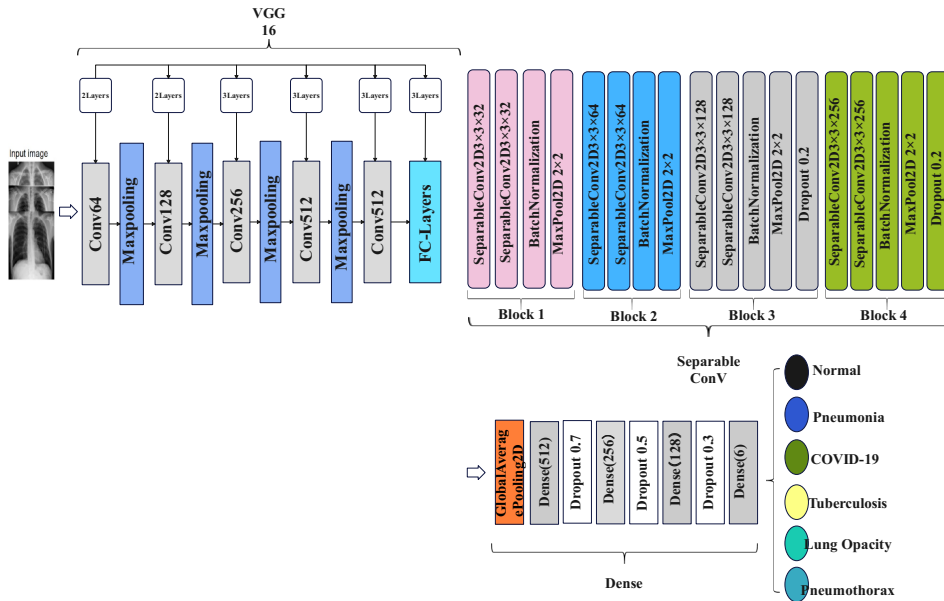


Figure 10: The framework of the VGG16+CNN network model.

However, the suggested CNN with attention model required a longer time than the CNN model as well as other compared models.

6 Conclusion

In this study, a lightweight deep learning model was presented to efficiently and accurately diagnose lung diseases using minimal computational resources. The primary objective was to develop a multi-classification model that balances high diagnostic ac-

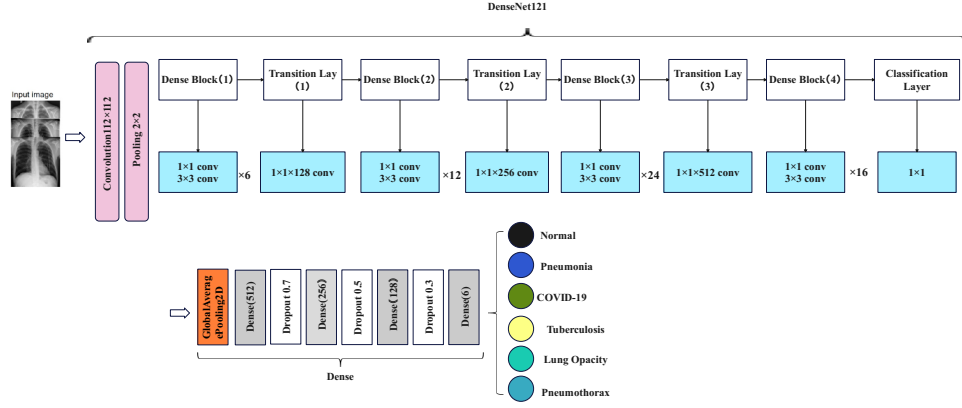


Figure 11: The framework of the DenseNet121 network model.

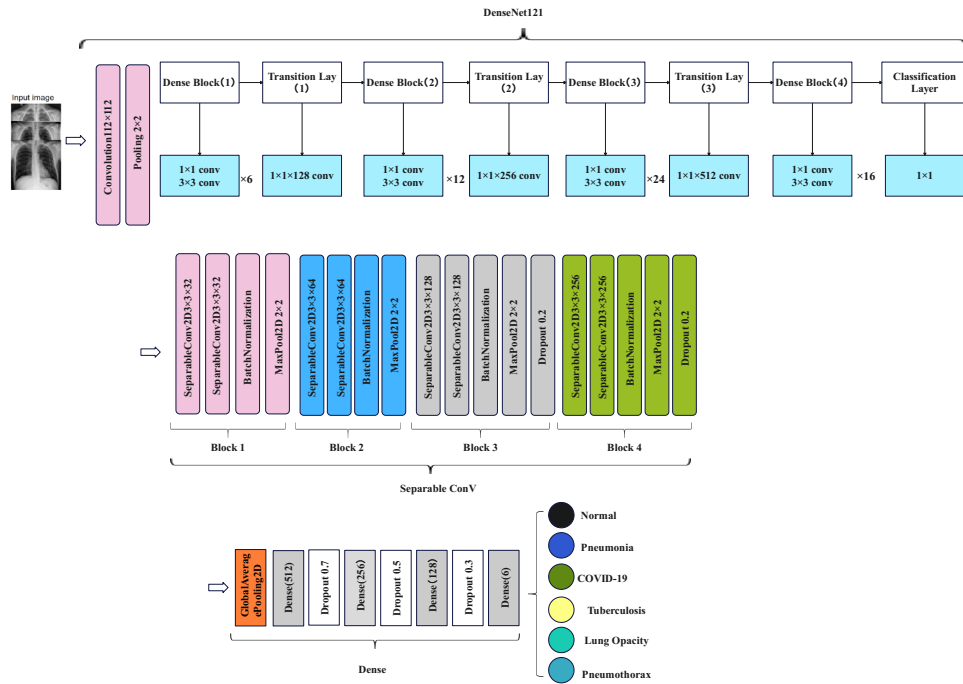


Figure 12: The framework of the DenseNet121+CNN network model framework.

Table 5: Comparison results for all models using five-fold cross-validation.

Model	Precision	Recall	F1-score
DenseNet121	0.8389	0.8243	0.8253
VGG16	0.8044	0.8079	0.8034
DenseNet121+CNN	0.8902	0.8902	0.8901
VGG16+CNN	0.8593	0.8447	0.8396
CNN	0.9187	0.9010	0.9037
Proposed model	0.9453	0.9433	0.9434

curacy with a streamlined, lightweight architecture. To address the common challenge of overfitting—often caused by high data dimensionality and complex network structures—a simple self-attention CNN model was implemented. The model’s effectiveness was validated using publicly available datasets from GitHub and Kaggle. Results demonstrate that the proposed deep learning approach accurately identifies various lung diseases and outperforms several existing models. Future research will focus on exploring

more advanced preprocessing techniques and additional methods to mitigate overfitting. Moreover, sophisticated feature extraction methods will be integrated to further enhance model performance.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Funding

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

References

- [1] Darren R Brenner, John R McLaughlin, and Rayjean J Hung. Previous lung diseases and lung cancer risk: a systematic review and meta-analysis. *PloS one*, 6(3):e17479, 2011.
- [2] FM Javed Mehedi Shamrat, Sami Azam, Asif Karim, Kawsar Ahmed, Francis M Bui, and Friso De Boer. High-precision multiclass classification of lung disease through customized mobilenetv2 from chest x-ray images. *Computers in Biology and Medicine*, 155:106646, 2023.
- [3] Aldimir Bruzadin, Maurílio Boaventura, Marilaine Colnago, Rogério Galante Negri, and Wallace Casaca. Learning label diffusion maps for semi-automatic segmentation of lung ct images with covid-19. *Neurocomputing*, 522:24–38, 2023.
- [4] Ying Chen, Longfeng Feng, Cheng Zheng, Taohui Zhou, Lan Liu, Pengfei Liu, and Yi Chen. Ldanet: Automatic lung parenchyma segmentation from ct images. *Computers in Biology and Medicine*, 155:106659, 2023.
- [5] Sergio Varela-Santos and Patricia Melin. A new modular neural network approach with fuzzy response integration for lung disease classification based on multiple objective feature optimization in chest x-ray images. *Expert Systems with Applications*, 168:114361, 2021.
- [6] Guan-Hua Huang, Qi-Jia Fu, Ming-Zhang Gu, Nan-Han Lu, Kuo-Ying Liu, and Tai-Been Chen. Deep transfer learning for the multilabel classification of chest x-ray images. *Diagnostics*, 12(6), 2022.
- [7] Stefanus Tao Hwa Kieu, Abdullah Bade, Mohd Hanafi Ahmad Hijazi, and Hoshang Kolivand. A survey of deep learning for lung disease detection on medical images: state-of-the-art, taxonomy, issues and future directions. *Journal of imaging*, 6(12):131, 2020.
- [8] Alan Kaplan, Hui Cao, J Mark FitzGerald, Nick Iannotti, Eric Yang, Janwillem WH Kocks, Konstantinos Kostikas, David Price, Helen K Reddel, Ioanna Tsiligianni, et al. Artificial intelligence/machine learning in respiratory medicine and potential role in asthma and copd diagnosis. *The Journal of Allergy and Clinical Immunology: In Practice*, 9(6):2255–2261, 2021.

- [9] Mohammed AA Al-qaness, Jie Zhu, Dalal AL-Alimi, Abdelghani Dahou, Saeed Hamood Alsamhi, Mohamed Abd Elaziz, and Ahmed A Ewees. Chest x-ray images for lung disease detection using deep learning techniques: A comprehensive survey. *Archives of Computational Methods in Engineering*, pages 1–35, 2024.
- [10] Pankaj Nanglia, Sumit Kumar, Aparna N Mahajan, Paramjit Singh, and Davinder Rathee. A hybrid algorithm for lung cancer classification using svm and neural networks. *ICT Express*, 7(3):335–341, 2021.
- [11] Sourabh Singh Verma, Ajay Prasad, and Anil Kumar. Covxmlc: High performance covid-19 detection on x-ray images using multi-model classification. *Biomedical signal processing and control*, 71:103272, 2022.
- [12] Zarin Anjuman Sejuti and Md Saiful Islam. A hybrid cnn–knn approach for identification of covid-19 with 5-fold cross validation. *Sensors International*, 4:100229, 2023.
- [13] Shagun Sharma and Kalpna Guleria. A deep learning based model for the detection of pneumonia from chest x-ray images using vgg-16 and neural networks. *Procedia Computer Science*, 218:357–366, 2023.
- [14] Ines Chouat, Amira Ehtioui, Rafik Khemakhem, Wassim Zouch, Mohamed Ghorbel, and Ahmed Ben Hamida. Lung disease detection in chest x-ray images using transfer learning. In *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 1–6. IEEE, 2022.
- [15] Kavitha Rani Balmuri, Srinivas Konda, Kishore kumar Mamidala, Madhukar Gunda, et al. Automated and reliable detection of multi-diseases on chest x-ray images using optimized ensemble transfer learning. *Expert Systems with Applications*, 246:122810, 2024.
- [16] A Victor Ikechukwu, S Murali, R Deepu, and RC Shivamurthy. Resnet-50 vs vgg-19 vs training from scratch: A comparative analysis of the segmentation and classification of pneumonia from chest x-ray images. *Global Transitions Proceedings*, 2(2):375–381, 2021.
- [17] Muhammad Attique Khan, Venkatesan Rajinikanth, Suresh Chandra Satapathy, David Taniar, Jnyana Ranjan Mohanty, Usman Tariq, and Robertas Damaševičius. Vgg19 network assisted joint segmentation and classification of lung nodules in ct images. *Diagnostics*, 11(12):2208, 2021.
- [18] Nallamothu Sri Kavya, N Veeranjanyulu, D Divya Priya, et al. Detecting covid19 and pneumonia from chest x-ray images using deep convolutional neural networks. *Materials Today: Proceedings*, 64:737–743, 2022.
- [19] Sadia Showkat and Shaima Qureshi. Efficacy of transfer learning-based resnet models in chest x-ray image classification for detecting covid-19 pneumonia. *Chemometrics and Intelligent Laboratory Systems*, 224:104534, 2022.
- [20] Murat Canayaz. Mh-covidnet: Diagnosis of covid-19 using deep neural networks and meta-heuristic-based feature selection on x-ray images. *Biomedical Signal Processing and Control*, 64:102257, 2021.

- [21] M Surya Bhupal Rao, Y Mallikarjuna Rao, C Venkataiah, GLN Murthy, M Dhara-
rani, and Manjula Jayamma. Deep learning based classification of covid-19 severity
using hierarchical deep maxout model. *Biomedical Signal Processing and Control*,
88:105653, 2024.
- [22] Mohan Abdullah, Beshir Kadir, Tamirat Tadesse Takore, et al. A hybrid deep learn-
ing cnn model for covid-19 detection from chest x-rays. *Heliyon*, 2024.
- [23] Varadharajan Indumathi and Rathinavelayutham Siva. An efficient lung disease
classification from x-ray images using hybrid mask-rcnn and bidlstm. *Biomedical
Signal Processing and Control*, 81:104340, 2023.
- [24] Awari Harshavardhan, Sreevardhan Cheerla, Anbusubramanian Parkavi, Saleth An-
gel Latha Mary, Kashif Qureshi, and Harshada Rangnath Mhaske. Deep learning
modified neural networks with chicken swarm optimization-based lungs disease detec-
tion and severity classification. *Journal of Electronic Imaging*, 32(6):062603–062603,
2023.
- [25] Abobaker Mohammed Qasem Farhan and Shangming Yang. Automatic lung dis-
ease classification from the chest x-ray images using hybrid deep learning algorithm.
Multimedia Tools and Applications, 82(25):38561–38587, 2023.
- [26] Yufei Jin, Huijuan Lu, Wenjie Zhu, and Wanli Huo. Deep learning based classification
of multi-label chest x-ray images via dual-weighted metric loss. *Computers in Biology
and Medicine*, 157:106683, 2023.
- [27] Guan-Ying Chen and Chih-Ting Lin. Multi-task supervised contrastive learning for
chest x-ray diagnosis: A two-stage hierarchical classification framework for covid-19
diagnosis. *Applied Soft Computing*, page 111478, 2024.
- [28] Julián D Arias-Londoño and Juan I Godino-Llorente. Analysis of the clever hans
effect in covid-19 detection using chest x-ray images and bayesian deep learning.
Biomedical Signal Processing and Control, 90:105831, 2024.
- [29] Lingzhi Kong and Jinyong Cheng. Classification and detection of covid-19 x-ray
images based on densenet and vgg16 feature fusion. *Biomedical Signal Processing
and Control*, 77:103772, 2022.
- [30] Paul Mooney. Chest x-ray images (pneumonia). 2018. URL [https://www.kag-
gle.com/paultimothymooney/chest-xray-pneumonia](https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia). (Last accessed: 2020 April 21),
2020.
- [31] Nida Nasir, Afreen Kansal, Feras Barneih, Omar Al-Shaltone, Talal Bonny, Mo-
hammad Al-Shabi, and Ahmed Al Shammaa. Multi-modal image classification of
covid-19 cases using computed tomography and x-rays scans. *Intelligent Systems
with Applications*, 17:200160, 2023.
- [32] Lingling Fang and Xin Wang. Covid-rdnet: A novel coronavirus pneumonia classifi-
cation model using the mixed dataset by ct and x-rays images. *Biocybernetics and
Biomedical Engineering*, 42(3):977–994, 2022.

- [33] Shayan Hassantabar, Mohsen Ahmadi, and Abbas Sharifi. Diagnosis and detection of infected tissue of covid-19 patients based on lung x-ray image using convolutional neural network approaches. *Chaos, Solitons & Fractals*, 140:110170, 2020.
- [34] Kaggle User: prashant268. Chest x-ray covid19 and pneumonia. <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>, 2020. Accessed: 2025-01.
- [35] Kaggle User: kamildinleyici. Covid normal viral opacity v2. <https://www.kaggle.com/kamildinleyici/covid-normal-viral-opacity-v2>, 2020. Accessed: 2025-01.
- [36] Kaggle User: volodymyrgavrysh. Pneumothorax binary classification task. <https://www.kaggle.com/volodymyrgavrysh/pneumothorax-binary-classification-task>, 2020. Accessed: 2025-01.
- [37] Kaggle User: donjon00. Covid-19 detection. <https://www.kaggle.com/donjon00/covid19-detection>, 2020. Accessed: 2025-01.
- [38] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 chest x-ray dataset. <https://github.com/ieee8023/covid-chestxray-dataset>, 2020. Accessed: 2025-01.
- [39] Kaggle User: sid321axn. Covid cxr image dataset research. <https://www.kaggle.com/sid321axn/covid-cxr-image-dataset-research>, 2020. Accessed: 2025-01.
- [40] Kaggle User: paultimothymooney. Chest x-ray images (pneumonia). <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>, 2018. Accessed: 2025-01.
- [41] Md. Nahiduzzaman, Md. Rabiul Islam, and Rakibul Hassan. Chestx-ray6: Prediction of multiple diseases including covid-19 from chest x-ray images using convolutional neural network. *Expert Systems with Applications*, 211:118576, 2023.