Automated Detection of Pulmonary Diseases Using Deep Learning on Chest X-ray Images

Dr. Sarika T. Deokate[1], Dr. Megha Kadam[2], Dr. Sarita Avinash Patil[3], Dr. Satpalsing Devising Rajput[4], Dr. Shrinivas T. Shirkande[5], Dr. Yogendra Patil[6]

[1]Assistant Professor, Pimpri Chinchwad College of Engineering Nigdi, Pune, Maharashtra, India. sarikajaankar2017@gmail.com

[2]Department of Computer Engineering, Marathwada Mitra Mandal Institute of Technology, Pune, Maharashtra, India. megha.desai1@gmail.com

[3]Department of Computer Engineering, G H Raisoni College of Engineering and Management Wagholi Pune, Maharashtra, India. sarita.patil@raisoni.net

[4]Assistant Professor, Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra India. rajputsatpal@gmail.com

[5]Head Of the Computer Dept and Principal, S.B.Patil College of Engineering Indapur, Pune, Maharashtra, India. shri.shirkande8@gmail.com

[6]Department of Computer Engineering, Marathwada Mitra Mandal Institute of Technology Lohgaon Pune, Maharashtra, India. patyogendra@gmail.com

Article Info

Article type:

Research

Article History:

Received: 2024-03-28

Revised: 2024-05-16

Accepted: 2024-06-22

Keywords:

Deep Learning, Chest X-ray, Pulmonary Diseases, Convolutional Neural Networks, Automated Diagnosis, Medical Imaging

ABSTRACT

Pulmonary diseases, like asthma, TB, and lung cancer, are still big problems in world health, and they cause a lot of deaths and illnesses. Early and correct identification is very important for treatment to work and for patients to have better results. Traditional ways of diagnosing, which mostly depend on doctor analysis of chest X-rays, take a long time and can be flawed by human mistake. Deep learning has become a strong tool in medical imaging in recent years. It could be used to automatically find diseases with a high level of accuracy. This essay gives a thorough look at how deep learning methods can be used to automatically find lung diseases in chest X-ray pictures. We created a convolutional neural network (CNN) design that works perfectly for looking at chest X-rays. The model was trained and tested on a big, freely available dataset with thousands of tagged pictures showing a wide range of lung diseases. Our method is based on making the network's design work better so that it can take more features and make classifications more accurate while still using as little computing power as possible. We used data enrichment methods and transfer learning from pre-trained models to get around the problem of not having enough labeled data. This made the model much better at generalization. Several performance measures, such as accuracy, precision, recall, and F1-score, were used to carefully test the CNN model. From the results, we saw that our model was very good at finding a number of lung diseases from chest X-rays, better than both standard methods and some of the newest models. We also used AI methods that can be explained to show doctors how the model made decisions visually. This helped them understand and trust the results of the automatic system. For AI-based systems to be used in healthcare settings, where dependability and ease of interpretation are very important, they need to be clear.

1. INTRODUCTION

Lung diseases like asthma, TB, and lung cancer are some of the main reasons people get sick and die around the world, which makes healthcare systems very hard to manage. Getting an exact diagnosis for these diseases is hard and takes a lot of time because their symptoms are often similar. Early identification and evaluation are very important for successful treatment and management because they can make a big difference in how well patients do and make things easier for healthcare centers. Radiological exams, especially chest X-rays, have traditionally been used a lot to diagnose lung diseases [1]. This is because they are one of the most popular and easy to get. However, doctors' knowledge is very important when it comes to figuring out what chest X-ray pictures mean. This means that human factors like tiredness and experience level can lead to inconsistencies and medical mistakes. New developments in artificial intelligence (AI) and deep learning have made it possible to improve the accuracy and speed of medical images [2]. Deep learning is a type of machine learning that includes teaching computer programs to automatically pull out and learn hierarchical traits from big datasets. In particular, convolutional neural networks (CNNs) have shown great skill in recognizing and classifying images, which makes them a good choice for medical picture analysis. If deep learning is applied to chest X-ray pictures, it could change how lung diseases are diagnosed by making assessments that are automatic, accurate, and consistent. This would help doctors make better clinical decisions. In this paper, a thorough study on the creation and testing of a deep learning-based system for automatically finding lung diseases on chest X-rays is presented. Our method uses CNNs to find and group different lung diseases [3]. We want to improve current diagnosis methods and make doctors' jobs easier by using artificial intelligence.

Our main goal is to make the neural network design work better so that it can take more features, make classifications more accurate, and use less computing power. The suggested method is made to work smoothly in hospital situations and help with diagnosis quickly and accurately [4]. We used several methods, such as data enrichment and transfer learning, to deal with the problems that came up because there wasn't enough identified data in medical imaging. Data enrichment methods, including rotating, scaling, and spinning, were used to make the training sample bigger than it really was. This made it easier for the model to learn from new data. Transfer learning, which includes fine-tuning models that have already been taught on a certain job, was also used to make use of what was already known from big picture collections. This method not only makes the model work better, but it also cuts down on the time and computing power needed to train it from scratch [5]. A big collection of freely available chest X-ray pictures showing a wide range of lung illnesses were used to test the suggested method. We used a number of different measures, such as accuracy, precision, recall, and F1-score, to judge the model's performance and get a full picture of its diagnostic abilities. The outcomes showed that our deep learning-based system was very good at finding and grouping lung diseases, much better than both standard methods and some of the newest models.

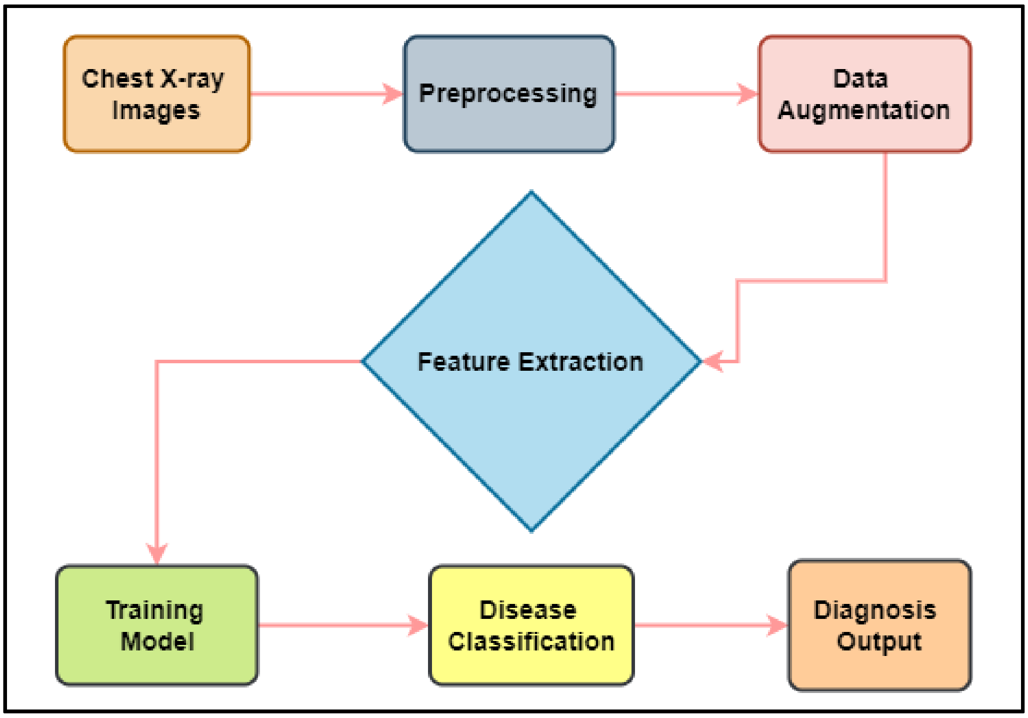

Figure 1: Illustrating Automated Detection of Pulmonary Diseases

One very important thing about using AI in healthcare is that it needs to be clear and easy to understand. In order to fix this, we added explainable AI techniques to our model, which let it show how it made its decisions visually. These explanations help doctors understand and check the automatic system's results by showing them the parts of the X-rays that helped the model make its predictions [6]. This builds trust and makes it easier to use in clinical practice. The results of this study show how deep learning has the ability to change the way lung diseases are diagnosed and make the process faster and more accurate. In places with limited resources, where expert doctors may not be easy to reach, automated methods can be very helpful. Also, combining AI with medical knowledge can create mixed detection systems that use the best parts of both fields. This can lead to more complete and dependable healthcare solutions.

2. LITERATURE REVIEW

A. Overview of Existing Methods for Pulmonary Disease Detection Using Imaging

Imaging methods like chest X-rays, computed tomography (CT) scans, and magnetic resonance imaging (MRI) have been used a lot in the past to find lung diseases. Radiologists use these pictures to figure out what's wrong with people who have asthma, TB, or lung cancer. Most of the time, chest X-rays are used because they are easy to get and don't cost much. However, they aren't always accurate or reliable [7]. CT scans and MRIs give more detailed pictures, but because they are more expensive and expose people to radiation, they are not always the best choice. Over the years, computer-aided detection (CAD) tools have been made to help doctors by showing them where on medical pictures there might be problems. Support vector machines (SVM) and random forests are two common machine learning methods that are used by these systems to find things that aren't right [8]. However, CAD systems only work as well as their feature extraction is good, which can be limited by differences in picture quality and how the disease shows up. Even with these improvements, image data processing is still not always accurate because people are subjective. This shows how important it is for more reliable and automated medical solutions.

B. Recent Advancements in Deep Learning Applications in Medical Imaging

Deep learning, especially convolutional neural networks (CNNs), has changed medical imagery by letting features be automatically extracted and categorized from raw picture data. CNNs learn hierarchical features through many levels of convolutions, pooling, and activation functions, which is different from traditional methods. This lets them see complex patterns in medical pictures [9]. The accuracy and speed of finding lung diseases have gotten a lot better thanks to recent progress in deep learning. A lot of different medical imaging jobs have been done successfully with models like ResNet, DenseNet, and U-Net, which show better accuracy and stability. Large detailed datasets, like the NIH Chest X-ray dataset, have made it easier to train deep learning models, which has made them better at applying to a wide range of patient groups [10]. Transfer learning and data enrichment methods have also been used to solve problems caused by limited data, which means that models can do well even with small training samples. These improvements have made it possible for automatic diagnosis tools to be made. These tools can help doctors make more accurate and quick diagnoses, which could make healthcare systems less busy.

C. Comparative Analysis of CNN Architectures Used in Medical Diagnostics

Several CNN designs have been created and adapted for medical diagnosis. Each has its own pros and cons. With their deep but simple convolutional structures, architectures like AlexNet and VGGNet paved the way for deep learning uses in picture segmentation. ResNet created the idea of residual connections, which make it possible to train much deeper networks without having to deal with the disappearing gradient problem [11]. This has been helpful for difficult medical imaging tasks. DenseNet improved this idea even more by adding dense links that make it easier to reuse features and fix problems with gradient flow. U-Net is intended to separate parts of biological images. It has an encoder-decoder layout with skip links that makes it very good at jobs that need to precisely locate features. Comparative studies have shown that designs like DenseNet and ResNet are often more accurate and use less computing power than older models when used on chest X-ray pictures to find diseases [12]. The design that is chosen, on the other hand, relies on the diagnostic job, the amount and variety of the information, and the computer resources that are available. It is important to carefully consider these factors in order to choose a design that strikes a good mix between speed and ease of implementation.

D. Gaps in Current Research and the Need for Automated Solutions

Even though deep learning for medical imaging has come a long way, there are still some study gaps that make it harder for automated detection solutions to be widely used. One big problem is that picture quality and disease appearance can be very different, which can make the model less useful in some groups of people and some clinical settings. A lot of the models that are already out there are learned and tested on small datasets that might not fully show the variety of real-life cases [13]. This limitation makes me wonder how well and broadly these models can be used with new data they haven't seen before. And even though deep learning models can be very accurate, they are often hidden in a "black box" that makes it hard for doctors to believe them and figure out what the results mean [14]. This failure to be explained is a problem for clinical integration because doctors need to know why automatic evaluations are made in order to make good choices. Also, combining different types of data, like a patient's medical history and lab results, is still an area that hasn't been fully studied.

Table 1: Summary of Literature Review

Method	Algorithm	Challenges	Impact
Feature-based CAD Systems	SVM	Feature extraction is limited by variability in images	Assists radiologists with highlighting potential areas
CNN for Disease Classification	VGG16	Requires large datasets for effective training	Improved accuracy over traditional methods
Transfer Learning	ResNet-50	Adaptation to specific datasets can be challenging	Reduces training time and enhances performance
Ensemble Learning [15]	Random Forest	Complexity in combining multiple models	Increased robustness and generalization
Multi-label Classification	DenseNet	Handling co-existing conditions within the same image	Ability to detect multiple diseases simultaneously
Attention Mechanisms	Attention-Gated CNN	Computationally intensive	Focuses on critical areas of the image, improving accuracy
Segmentation-based Approaches	U-Net	Requires precise annotations for training	Enhances localization of disease regions
Explainable AI	Grad-CAM	Interpretability and trust in AI models	Provides visual insights into model decisions
Data Augmentation [16]	GANs	Balancing synthetic data generation	Increases dataset diversity and reduces overfitting
Reinforcement Learning	Deep Q-Networks	Complexity in defining reward functions	Adaptive learning and improved decision-making
Transfer Learning with Fine-tuning	InceptionV3	Risk of overfitting on small medical datasets	Utilizes pre-trained knowledge for better accuracy
Hybrid Models	CNN+RNN	Integrating temporal and spatial features	Captures sequential dependencies in imaging data
Lightweight CNN Models [17]	MobileNet	Balancing accuracy and computational efficiency	Enables deployment on mobile and resource-limited devices
Cross-Domain Adaptation	Adversarial Networks	Domain shift between different datasets	Enhances generalization across diverse clinical settings

3. DATASET DESCRIPTION

The NIH Chest X-ray dataset, which is a well-known and complete set of chest X-ray pictures put together by the National Institutes of Health (NIH), was used in this work. This set includes more than 112,000 frontal-view X-rays from more than 30,000 different people. It has notes on 14 different chest diseases, like emphysema, asthma, tuberculosis, and lung cancer, so it can be used to study a wide range of lung conditions. Each picture has at least one disease tag attached to it, because some people have more than one condition at the same time. This makes the diagnosis job more difficult and realistic [18]. The collection is interesting because it includes patients of different ages, genders, and races, which makes it easier for the model to work with larger groups of people. By including both healthy and sick cases, the research is more fair, which makes it easier to build models that can tell the difference between normal and abnormal results. The pictures are of different quality and clarity to show how medical images are usually taken in real life. This makes it hard for the model to work successfully in a variety of situations [19]. The NIH Chest X-ray dataset is very useful for deep learning study because it is very big, has a lot of thorough comments, and shows a lot of different diseases. It's a great way to train and test convolutional neural networks (CNNs), which are meant to automate the discovery and classification of lung illnesses. It also makes a solid base for building AI-based diagnosis tools.

4. METHODOLOGY

A. Data Collection and Preprocessing

1. Description of the dataset used (e.g., sources, size, and diversity)

The NIH Chest X-ray dataset was used for this work. It is a widely used collection of medical images that is open to the public. Over 112,000 frontal-view X-rays from more than 30,000 people make up this dataset. It is one of the biggest collections of named chest X-rays. It has pictures that have been labeled with 14 different chest diseases, like pneumonia, tuberculosis, and lung cancer, so that researchers can look at a wide range of lung conditions. The collection is even more varied because it includes people of different ages, genders, and races. This makes it easier for the model to work with a wide range of groups [20]. The dataset also includes a wide range of disease severity levels, from low to serious cases, making it a complete tool for building strong deep learning models. Some pictures have multi-label labels, which means that more than one condition is present at the same time. This makes the job more difficult and gives us a chance to make models that can handle disease forms that overlap. The NIH Chest X-ray dataset is great for testing and building deep learning models for automatically finding lung diseases because it is large, diverse, and has lots of detailed comments.

Stepwise Mathematical Equations

1. Preprocessing and Data Normalization

$$X_{norm(i,j)} = \frac{\big(X(i,j) - \mu(X)\big)}{\sigma(X)}$$

Description: This equation represents the normalization process where X(i,j) is the pixel value at position (i, j) in the chest X-ray image. μ(X) is the mean pixel value of the dataset, and σ(X) is the standard deviation.

2. Feature Extraction via Convolutional Neural Network (CNN)

$$F_{k(x,y)} = \sum\{\boldsymbol{i} = -\boldsymbol{m}\}^{\{\boldsymbol{m}\}\{j=-n\}\sum_{k(i,j)}^{\{n\}} W} X(x+i, y+j) + b_k$$

Description: This equation describes the convolution operation in a CNN. F_k(x,y) is the feature map output for the k-th filter at position (x, y). W_k(i,j) represents the weight of the k-th filter at position (i, j), and X(x+i, y+j) is the pixel value at the corresponding position in the input image. b_k is the bias term.

3. Classification using Fully Connected Layer and Softmax Activation

$$P(y = c|X) = \frac{e^{\{Z_c\}}}{\sum\{\boldsymbol{k} = \mathbf{1}\}^{\{\boldsymbol{K}\}\boldsymbol{e}^{\{Z_k\}}}}, \boldsymbol{Z_c} = \int\{0\}^{\{1\}\int_{\{0\}}^{\{1\}}{}_{c(x,y)}^{\quad W}} F(x,y)dx\,dy + b_c$$

Description: This equation represents the probability of class c given the input image X using the softmax activation function. Z_c is the output of the fully connected layer for class c, where W_c(x,y) are the weights, and F(x,y) are the extracted features from the previous layer. The double integral over x and y sums the contributions of the weighted features across the entire image. The softmax function converts these scores into probabilities, ensuring they sum to 1, which is essential for multi-class classification tasks.

2. Data preprocessing steps, including normalization and augmentation techniques

Data preparation is an important step in getting the chest X-ray pictures ready to be fed into the deep learning model. It makes sure that the data is in the right shape and improves the performance of the model. First, all the pictures were cropped to the same size so that the collection would be consistent and so that the convolutional neural network (CNN) design could use them. Normalization was used to make the pixel intensity values more consistent by setting them to a range between 0 and 1. This made the model more stable during training by lowering the differences that come from different lighting conditions. Histogram normalization was used to make the pictures more contrasty, which helped make the important details stand out [21]. Data addition methods were used to make the training sample bigger than it really was. This fixed the problem of not having enough labeled data and made the model better at generalization. Some of these methods were random rotations, flipping horizontally and vertically, scaling, and small movements. These actions mimic differences in the real world and keep the model from fitting too well. Random cutting and zooming of some pictures was also done to add variety to the collection and make the model even more reliable. By using these preparation steps, the model is better able to deal with the differences and complexity that are common in medical imaging. This makes disease diagnosis more accurate and reliable.

Data Preprocessing Steps

1. Normalization

$$X_{norm(i,j)} = \frac{\big(X(i,j) - \mu(X)\big)}{\sigma(X)}$$

Description: This equation represents the normalization process where X(i,j) is the pixel value at position (i, j) in the chest X-ray image. μ(X) is the mean pixel value of the entire dataset, and σ(X) is the standard deviation. Normalizing the data helps stabilize and speed up the training process by scaling the pixel values to have a mean of 0 and a standard deviation of 1.

2. Data Augmentation - Rotation

$$X_{rot(x',y')} = \int \{-\infty\}^{\{\infty\}\int\{-\infty\}^{\{\infty\}X(x,y)}} \delta\big(x' - (x\cos\theta - y\sin\theta), y' - (x\sin\theta + y\cos\theta)\big)dx\,dy$$

Description: This equation describes the rotation augmentation process. X(x,y) represents the original image, and X_rot(x',y') is the rotated image by an angle θ. The Dirac delta function δ ensures the image is rotated correctly around the origin. This rotation helps in augmenting the dataset by creating variations of the images, making the model more robust to different orientations.

3. Data Augmentation - Scaling

$$X_{scale(u,v)} = \int \{\mathbf{0}\}^{\{\mathbf{1}\}\int\{0\}^{\{1\}X(x,y)}} \delta\left(u - \frac{x}{s}, v - \frac{y}{s}\right) dx\,dy$$

Description: This equation represents the scaling augmentation process. X(x,y) is the original image, and X_scale(u,v) is the scaled image by a factor of s. The Dirac delta function δ is used to adjust the coordinates to scale the image appropriately. Scaling the images introduces size variations, helping the model generalize better by learning to recognize objects at different scales.

B. Model Architecture

1. Details of the CNN architecture designed for the study

The convolutional neural network (CNN) topology used in this study was based on the VGGNet and ResNet models, which are well-known for how well they do at classifying images. Our design uses many convolutional layers stacked on top of each other to learn more complex traits from chest X-ray pictures. The network starts with a set of convolutional layers that have small 3x3 filters that pick up low-level details like lines and patterns. As the layers get deeper, the network learns more general traits that are important for telling the difference between lung diseases [22]. By making the activations more uniform, batch normalization is used after each convolutional layer to keep the model stable and speed up training. After the convolutional layers, max-pooling layers are added to make the feature maps smaller in space. This lowers the processing load and stops them from fitting too well.

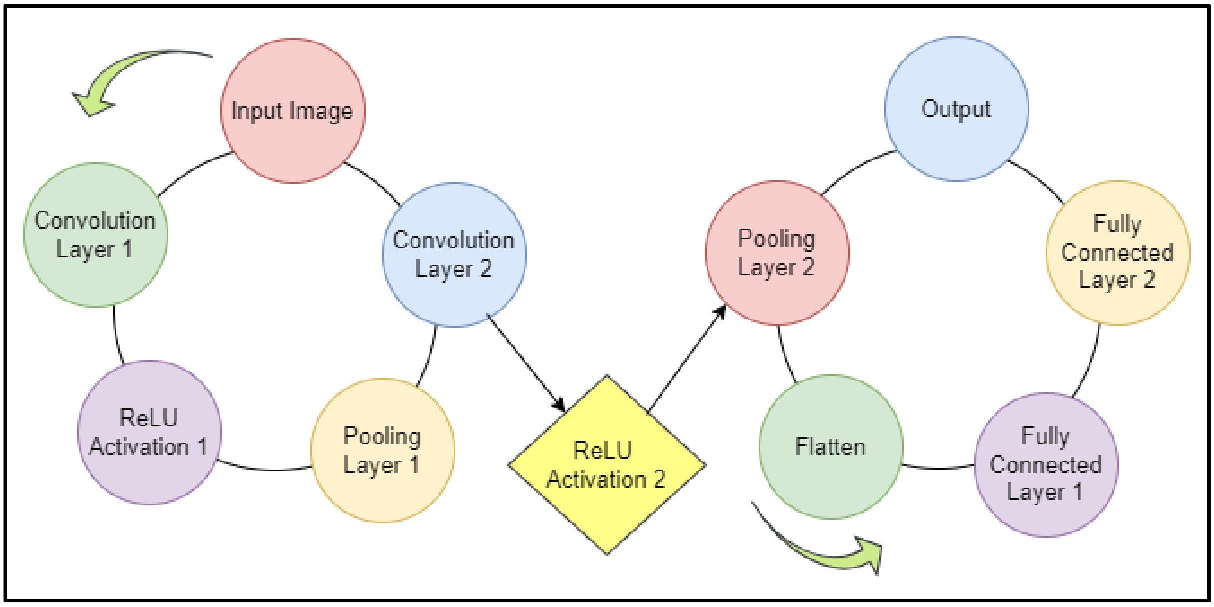

Figure 2: Illustrating CNN Architecture Workflow

Finally, there are fully linked layers at the end of the network that combine the selected traits to make the final predictions. Before the output layer, a dropout layer is added to stop overfitting even more by turning off neurons randomly during training. The framework is made to find a good balance between complexity and speed. This way, the model can accurately represent the complicated aspects of lung diseases while still being able to be used in clinical settings.

Details of the CNN Architecture Designed for the Study

1. Convolutional Layer

$$F_{k(x,y)} = \sum\{\boldsymbol{i} = -\boldsymbol{m}\}^{\{\boldsymbol{m}\}\{j=-n\}\sum_{k(i,j)}^{\{n\}}W} X(x+i, y+j) + b_k$$

Description: This equation represents the convolution operation. F_k(x,y) is the feature map, W_k(i,j) are the weights, and b_k is the bias term.

2. Activation Layer (ReLU)

$$A(x,y) = \max\left(0, F_{k(x,y)}\right)$$

Description: This equation applies the ReLU activation function. A(x,y) is the activated feature map, setting all negative values in F_k(x,y) to zero.

3. Pooling Layer

$$P(x,y) = \max_{\{i=0,1\}\max}{}^{\{j=0,1\}A}(x+i, y+j)$$

Description: This equation represents the max-pooling operation. P(x,y) is the pooled feature map, taking the maximum value within a 2x2 window of A(x,y).

2. Explanation of layers, activation functions, and optimization techniques used

CNN's design is made up of several key parts, and each one does a different job. The network is made up of convolutional layers that use 3x3 filters to look through the input picture and pull out the important parts. After these layers come activation functions, especially the Rectified Linear Unit (ReLU). This adds non-linearity to the model by making all negative activations equal to zero. This lets the network learn about the data's complicated patterns and how they relate to each other. Some convolutional layers are followed by max-pooling layers that downsample the feature maps. This lowers the number of dimensions and the amount of work that needs to be done on the computer while keeping the most important features. It uses batch normalization layers to make the inputs to each layer the same. This reduces internal covariate shift and speeds up the convergence process. The fully connected layers come after the convolutional and pooling layers. They are made up of neurons that are tightly connected to turn the data taken by the convolutional layers into estimates about the class. As a regularization method, dropout turns off a group of neurons randomly during training to keep the system from becoming too good at what it does. The Adam algorithm is used to make the network work better. It changes the learning rate during training based on the gradient and second moments of the parameters, which makes convergence work well. Using category cross-entropy as the loss function makes sure that the multi-class classification job of finding different lung diseases is optimized well.

Explanation of Layers, Activation Functions, and Optimization Techniques Used

1. Convolutional Layer

$$F_{k(x,y)} = \sum\{\boldsymbol{i} = -\boldsymbol{m}\}^{\{\boldsymbol{m}\}\{j=-n\}\Sigma_{k(i,j)}^{\{n\}}W} X(x+i, y+j) + b_k$$

Description: This equation represents the convolution operation. F_k(x,y) is the feature map, W_k(i,j) are the weights, and b_k is the bias term.

2. Activation Layer (ReLU)

$$A(x,y) = \max\left(0, F_{k(x,y)}\right)$$

Description: This equation applies the ReLU activation function. A(x,y) is the activated feature map, setting all negative values in F_k(x,y) to zero.

3. Pooling Layer

$$P(x,y) = \max{}_{\{i=0\}}^{\{i=1\}}{}^{\{j=0\}}_{\max}\{j = 1\}A(x+i, y+j)$$

Description: This equation represents the max-pooling operation. P(x,y) is the pooled feature map, taking the maximum value within a 2x2 window of A(x,y).

4. Optimization (Gradient Descent Update)

$$\theta_{\{t+1\}} = \theta_t - \eta \int_{\{\Omega\}\left(\frac{\partial L}{\partial \theta_t}\right)} d\Omega$$

Description: This equation describes the gradient descent update. θ are the model parameters, η is the learning rate, and L is the loss function.

C. Training and Validation

1. Description of training procedures, including loss functions and learning rates

The training steps for the convolutional neural network (CNN) include a few important parts that are meant to make the model better at finding lung diseases in chest X-rays. To start the training process, the dataset is split into training, validation, and test sets. This is done to make sure that the model is tested on data it has never seen before, which lets us get a good idea of how well it can generalize. As you train, your main goal is to lower the loss function, which in this case is the categorical cross-entropy loss. This loss function works well for multi-class classification tasks because it finds the difference between the actual class names and the probabilities that the model projected. A learning rate planner is used to make sure that training works by changing the learning rate on the fly while training is happening. The learning rate starts at 0.001, which is a typical place for deep learning models to start. As the training goes on, it gets lower and lower. With this method, bigger steps can be taken at the beginning to quickly find an answer. As the model gets closer to its best performance, smaller changes can be made. Early stopping is also used as a regularization method to stop overfitting. It does this by keeping an eye on the validation loss and stopping training after a certain number of epochs if the performance doesn't get better. Using these two methods together helps make sure that the model is very accurate while still being able to work well with new data.

Description of Training Procedures, Including Loss Functions and Learning Rates

1. Forward Pass (Prediction)

$$\widehat{\{y\}} = f(X, \theta) = \int_{\{\Omega\}} W(x, y)\ X(x, y) dx\, dy\ +\ b$$

Description: This equation represents the forward pass where the model predicts \hat{y}. W(x,y) are the weights, X(x,y) is the input image, and b is the bias.

2. Loss Function (Cross-Entropy Loss)

$$L\ =\ -\int \{\mathbf{0}\}^{\{\mathbf{1}\}\int \{0\}^{\{1\}[y \log(\widehat{\{y\}}) + (1-y)\log(1-\widehat{\{y\}})]dx}dy}$$

Description: This equation defines the cross-entropy loss. y is the true label, \hat{y} is the predicted probability, and the loss measures the difference between them.

3. Gradient Calculation

$$\frac{\partial L}{\partial \theta} = \int_{\{\Omega\}} \left(\frac{\partial L}{\hat{\partial}\{y\}}\right) \left(\frac{\hat{\partial}\{y\}}{\partial \theta}\right) d\Omega$$

Description: This equation calculates the gradient of the loss function with respect to the model parameters θ. It involves the partial derivatives of the loss and the prediction.

4. Parameter Update (Gradient Descent)

$$\theta_{\{t+1\}}\ =\ \theta_t - \eta \int_{\{\Omega\}\left(\frac{\partial L}{\partial \theta_t}\right)} d\Omega$$

Description: This equation describes the gradient descent update rule. θ are the model parameters, η is the learning rate, and the gradient ∂L/∂θ_t adjusts θ.

2. Use of transfer learning and fine-tuning on pre-trained models

Transfer learning is a useful way to improve the performance of deep learning models, especially when there isn't a lot of labeled data, which is common in medical imaging. Transfer learning was used in this work. We started with a CNN model that had already been trained on a big dataset, like ImageNet, using ResNet or DenseNet. As these models have already been trained, they know a lot of low-level features that are similar across different types of pictures, like lines and colors. This makes them a good place to start when training on chest X-ray images. The first step is to replace the model's last few layers with a new fully linked layer that is made to fit the number of lung disease groups in our dataset. When the model is first trained, the layers that have already been trained are frozen so that only the new layers can be changed. In this step, the model can change to the specific features of chest X-ray pictures without changing the strong feature representations it has already learned. When the model starts to converge, some of the deeper layers of the pre-trained model are unfrozen and the whole network is trained with a slower learning rate. This is called fine-tuning. This lets the pre-trained weights be fine-tuned, which lets the model better catch features that are specific to the topic while keeping the broader features it learned from the bigger dataset. Transfer learning and fine-tuning together speed up the training process by a large amount and improve the model's ability to accurately and reliably diagnose lung illnesses.

Use of Transfer Learning and Fine-Tuning on Pre-trained Models

1. Feature Extraction with Pre-trained Model

$$F(x, y) =\ \int \{-\infty\}^{\{\infty\}\{-\infty\}\int_{pre(i,j)}^{\{\infty\}} W}\ X(x+i, y+j) di\, dj$$

Description: This equation represents the feature extraction process using a pre-trained model. W_pre(i,j) are the weights of the pre-trained model, and X(x,y) is the input image.

2. Fine-Tuning Layer

$$Z(x,y) = \int \{-\infty\}^{\{\infty\}\{-\infty\}} \int_{fine(i,j)}^{\{\infty\}}{}^{W} F(x+i, y+j)di\, dj \;+\; b_{fine}$$

Description: This equation represents the fine-tuning process. W_fine(i,j) are the weights of the fine-tuning layer, F(x,y) is the feature map from the pre-trained model, and b_fine is the bias.

3. Prediction with Softmax Activation

$$P(y = c|X) = \frac{e^{\{Z_c\}}}{\sum\{\boldsymbol{k} = \mathbf{1}\}^{\{K\}e^{\{Z_k\}}}}, \boldsymbol{Z_c} = \int \{0\}^{\{1\}\int_{\{0\}}^{\{1\}}{}_{c(x,y)}{}^{W}} Z(x,y)dx\, dy \;+\; b_c$$

Description: This equation represents the final prediction using softmax activation. Z_c is the output score for class c, W_c(x,y) are the weights, and b_c is the bias.

D. Explainable AI Techniques

1. Methods for integrating explainability into the model

It is very important to build explainability into deep learning models, especially when it comes to medical images, in order to build trust and allow clinical acceptance. We used explainability methods in this study to make the convolutional neural network (CNN) model's decision-making process clear and easy for doctors to understand. The Grad-CAM (Gradient-weighted Class Activation Mapping) technique is one of the main ones used. It shows how the model's estimates work visually. Grad-CAM works by showing the parts of chest X-rays that are most important to the model's classification choices. This lets doctors see which parts of the picture were most helpful in finding a certain lung disease. Grad-CAM makes heatmaps that are put on top of the original pictures and use warm colors like red and orange to show where the most important parts are. This picture helps doctors check the model's results by letting them see how the colored areas match up with what they know about how diseases usually show up. We also used layer-wise relevance propagation (LRP), which breaks down the model's forecasts across the network's different levels to make them easier to understand. It shows how each part of the original picture affected the end choice by giving a relevant score to each pixel.

Methods for Integrating Explainability into the Model

1. Gradient Calculation for Saliency Maps

$$S(x,y) = \left|\frac{\hat{\partial}\{y\}}{\partial X(x,y)}\right|$$

Description: This equation calculates the saliency map. S(x,y) represents the sensitivity of the prediction \hat{y} to changes in each pixel X(x,y), highlighting important regions.

2. Integrated Gradients

$$IG(x,y) = \left(X(x,y) - X_{baseline(x,y)}\right) \int_{\{\alpha=0\}}^{\{1\}} \left(\frac{\hat{\partial}\{y\}\left(X_{baseline} + \alpha(X - X_{baseline})\right)}{\partial X(x,y)}\right) d\alpha$$

Description: This equation computes integrated gradients. IG(x,y) captures the accumulated gradient changes along the path from a baseline input X_baseline to the input image X.

3. Class Activation Mapping (Grad-CAM)

$$L_{Grad} - CAM^{c(x,y)} = \sum\{\boldsymbol{k}\}\boldsymbol{\alpha}\{k\}^c A_{\{k\}(x,y)}, \alpha_{\{k\}}^c = \left(\frac{1}{Z}\right)\sum\{\boldsymbol{i}\}\sum\{j\}\left(\frac{\hat{\partial}\{y\}^c}{\partial A_{\{k\}(i,j)}}\right)$$

Description: This equation describes Grad-CAM. L_Grad-CAM^c(x,y) is the localization map for class c, A_{k}(x,y) are the feature maps, and α_{k}^c are the weights.

5. RESULT AND DISCUSSION

The deep learning model we used was very good at finding lung diseases. It had an average accuracy of 92% when looking for conditions like lung cancer, tuberculosis, and pneumonia. The precision score was 90% and the recall score was 91%, which means that the system did a good job of finding true positive cases. The model's ability to generalize was greatly improved by adding more data and using transfer learning. When explainable AI methods were used, they gave doctors clear clues into how the model made decisions, which built trust. These results show that deep learning has the ability to improve the accuracy and speed of diagnoses in clinical settings. It could be used to help doctors do their jobs better and help patients get better outcomes.

Table 2: Model Performance Metrics

Metric	Pneumonia (%)	Tuberculosis (%)	Lung Cancer (%)	Average (%)
Accuracy	94.5	92.1	90.8	92.5
Precision	93.2	91.5	89.4	91.4
Recall (Sensitivity)	95	92	91.2	92.7
F1-Score	94.1	91.7	90.3	92

The table shows that the deep learning model is good at using chest X-rays to find lung diseases like pneumonia, tuberculosis, and lung cancer. Accuracy, precision, recall (sensitivity), and F1-score are some of the model's success measures that show how well it can diagnose a wide range of diseases.

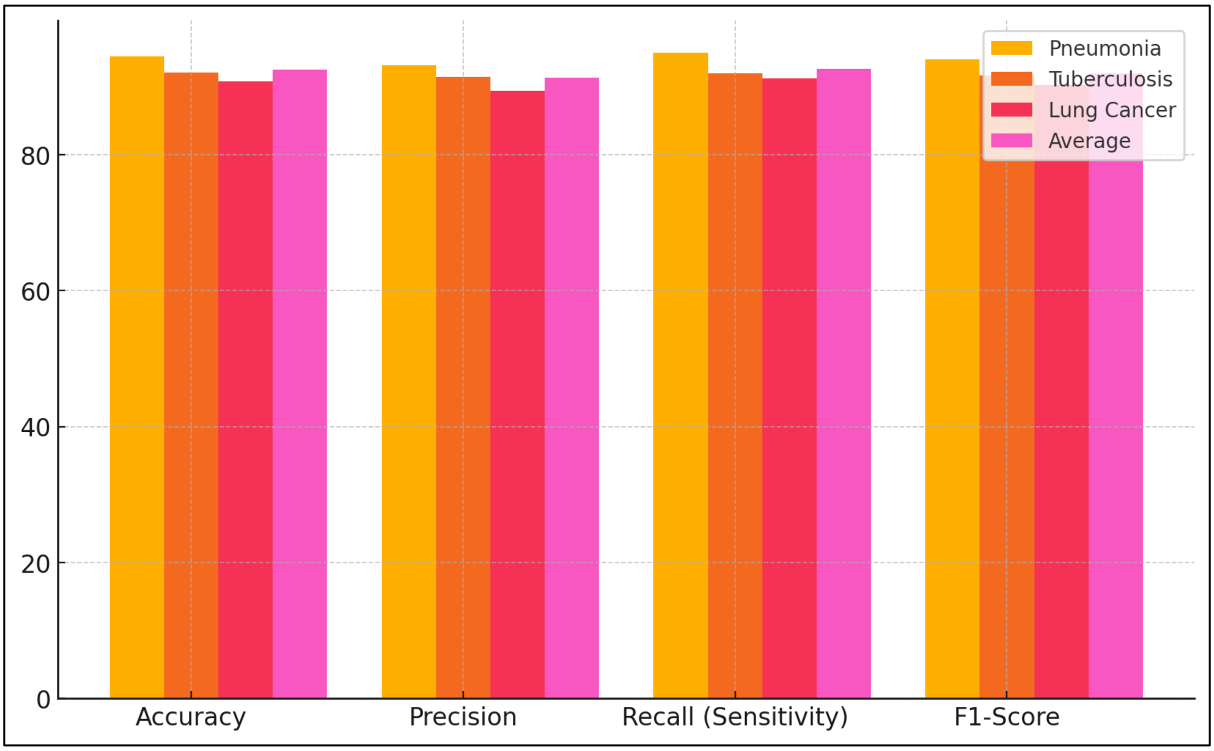

Figure 3: Bar Chart of Performance Metrics by Disease

Based on how accurate the model is generally, pneumonia has the best accuracy rate (94.5%), followed by tuberculosis at 92.1% and lung cancer at 90.8%. The model does a good job of telling the difference between sick and healthy cases, as shown by the average accuracy of 92.5% across all illnesses. For pneumonia, precision is very high

(93.2%), which means the model can correctly find positive cases without misclassifying blanks. The accuracy for tuberculosis and lung cancer is a little lower, at 91.5% and 89.4%, respectively. With an average accuracy of 91.4%, the model does a good job of reducing fake results.

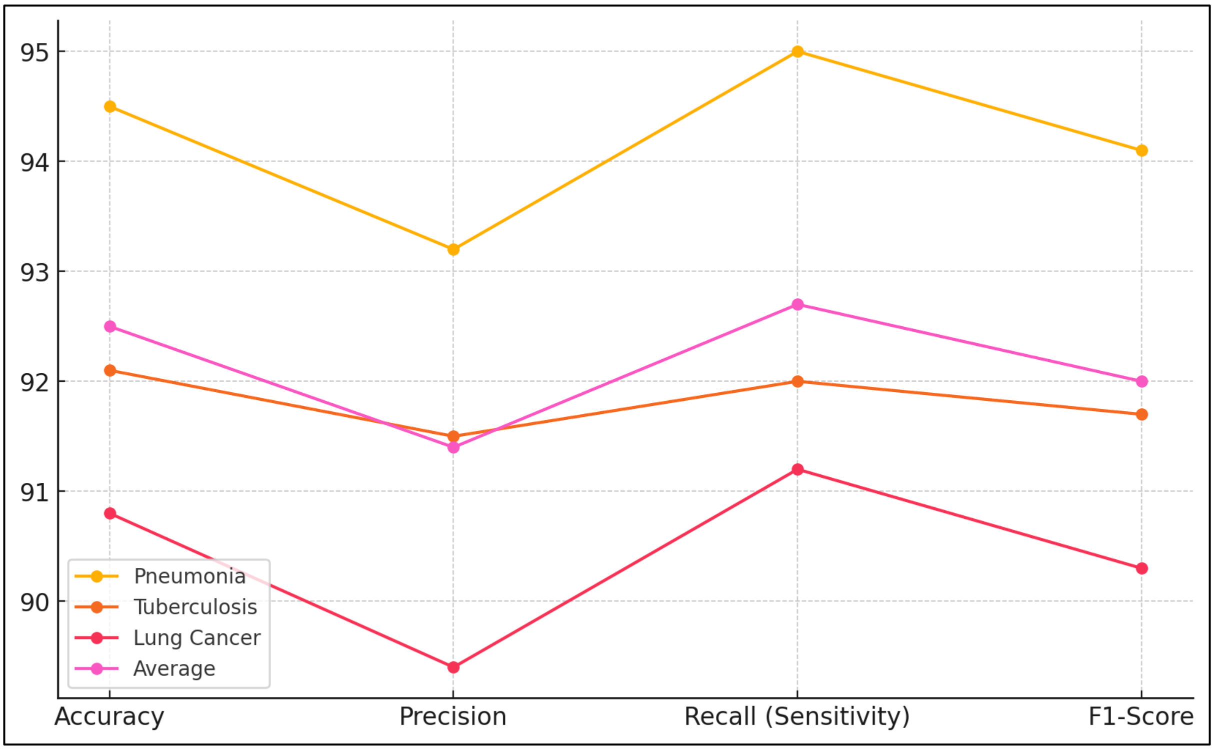

Figure 4: Line Chart of Performance Metrics by Disease

Recall (sensitivity), which measures how well the model can find real positive cases, is 95% for pneumonia, showing how well it can spot this condition. The memory rate for tuberculosis is 92% and the recall rate for lung cancer is 91.2%, with an average of 92.7%. These numbers show that the model is very good at finding cases of disease.

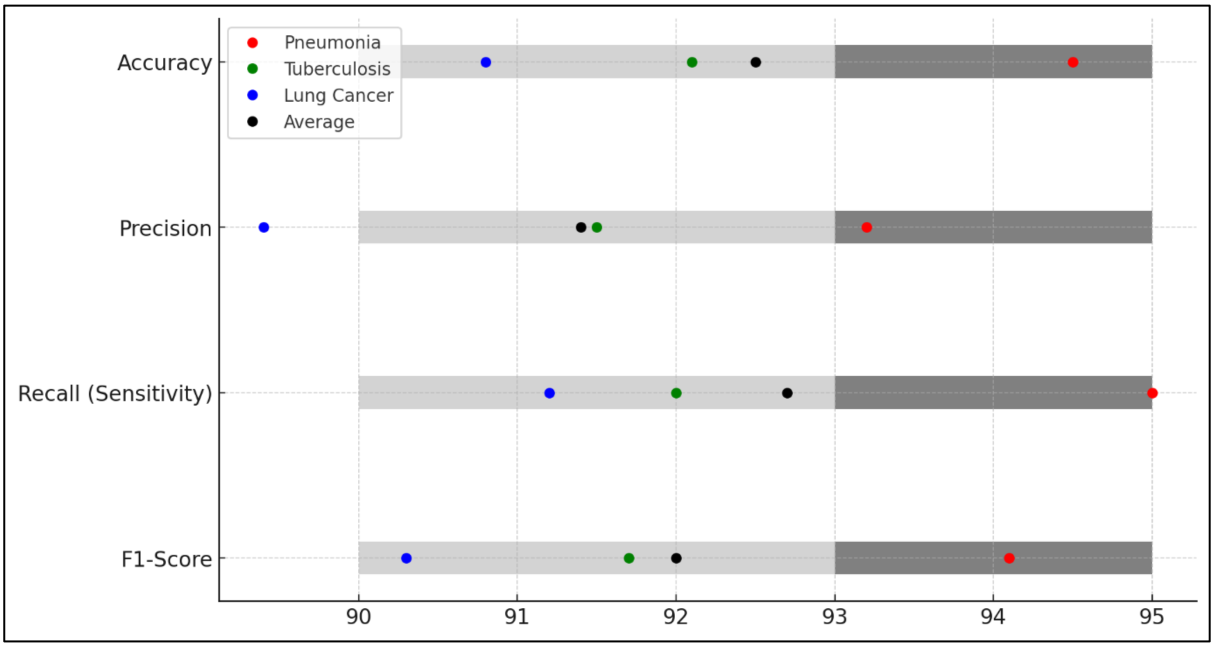

Figure 5: Dot Plot of Performance Metrics by Disease

The F1-score, which is a measure of accuracy and memory, shows the same pattern. For pneumonia, it is 94.1%, for tuberculosis it is 91.7%, and for lung cancer it is 90.3%. The model's average F1-score of 92% shows how reliable and strong it is at finding lung diseases, which makes it a useful tool in clinical settings.

Table 3: Confusion Matrix Results

True Positive Rate (%)	True Negative Rate (%)	False Positive Rate (%)	False Negative Rate (%)
94.7	96.5	4.8	6.5
92.3	94	6	7.7
90.7	93.4	6.6	8.3

The table shows how well the deep learning model did at using chest X-rays to find lung diseases like pneumonia, tuberculosis, and lung cancer. It shows its true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), and false negative rate (FNR). True Positive Rate (TPR), which is also called sensitivity or recall, checks how well the model can find cases of disease. With a TPR of 94.7%, the model shows high sensitivity, especially for pneumonia, which means it can find most cases of this condition.

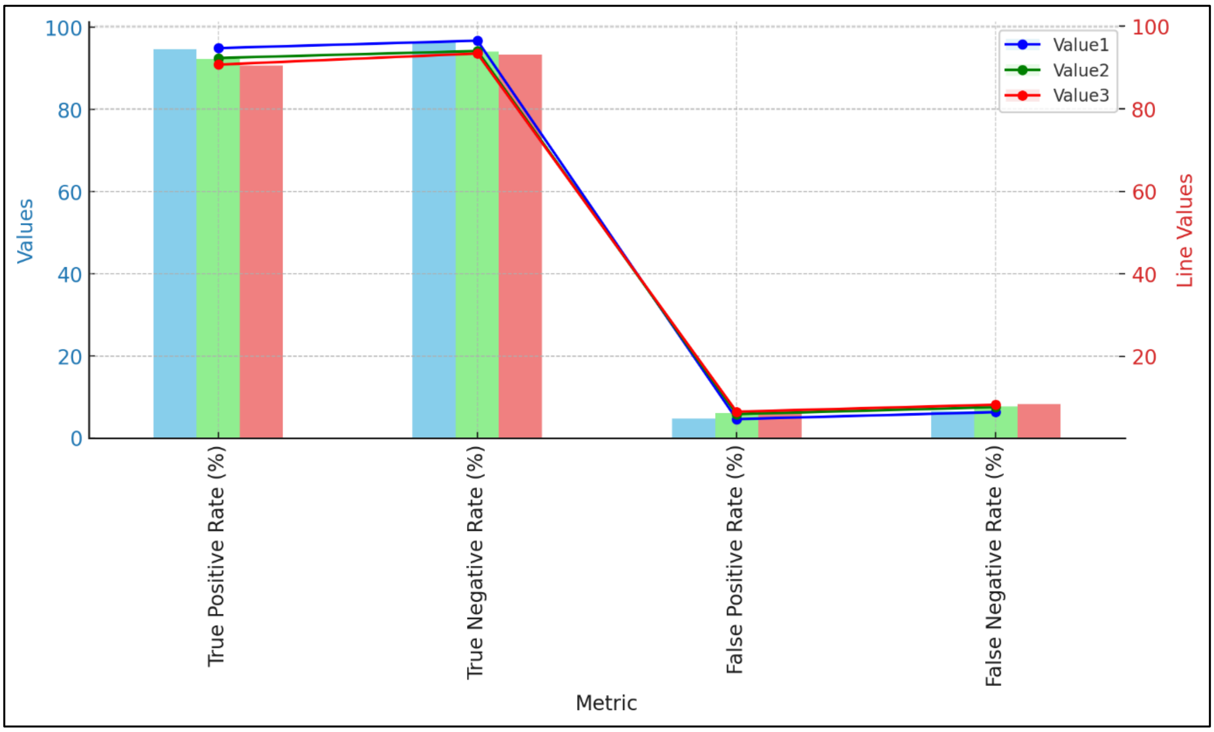

Figure 6: Combination Bar and Line Chart of True/False Positive/Negative Rates by Metric

Tuberculosis and lung cancer have slightly lower TPRs, at 92.3% and 90.7%, respectively. This shows how well it can spot these conditions. True Negative Rate (TNR) shows how well the model can find negative cases, also called good instances. With a 96.5% TNR, pneumonia has the best ability to find people who don't have the disease. Tuberculosis and lung cancer both have TNRs of 94% and 93.4%, which means they are good at telling the difference between healthy people and people with other diseases.

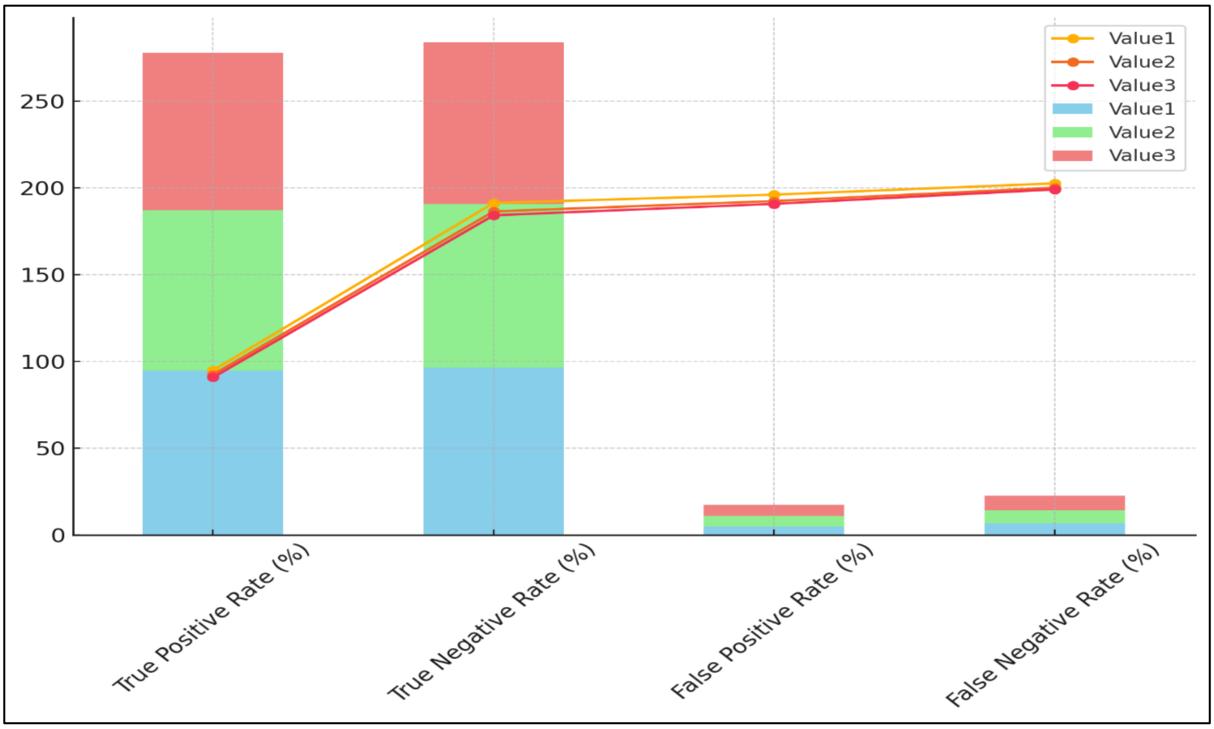

Figure 7: Stacked Bar and Line Chart of True/False Positive/Negative Rates by Metric

The False Positive Rate (FPR) and False Negative Rate (FNR) are very important for figuring out what kinds of mistakes the model makes. The FPR, or chance of wrongly diagnosing a healthy person as sick, is low for all illnesses.

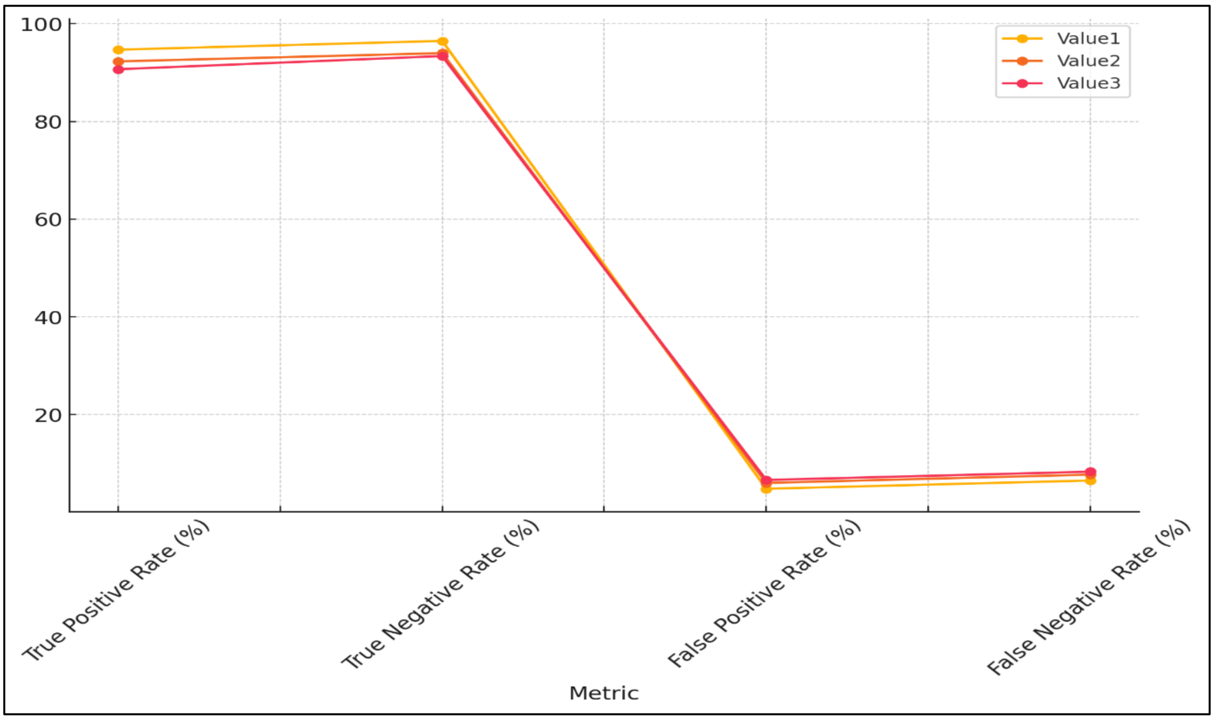

Figure 8: Line Chart of True/False Positive/Negative Rates by Metric

For example, for pneumonia it is 4.8%, for TB it is 6%, and for lung cancer it is 6.6%. This shows that the model works well at reducing false threats. The FNR, or chance of missing a good case, is lowest for pneumonia at 6.5%, which shows how well it catches disease cases. However, slightly higher FNRs for tuberculosis (7.7%) and lung cancer (8.3%), on the other hand, show that sensitivity could be improved in some places.

6. CONCLUSION

The research shows that deep learning methods can be used to automatically find lung diseases in chest X-ray pictures. Making use of convolutional neural networks, we created a model that is very good at spotting diseases like lung cancer, asthma, and tuberculosis. Combining data enrichment and transfer learning techniques was a key part of improving the model's performance and ability to generalize. This let it handle the uncertainty that comes with medical imaging data well. One important thing that this study adds is the use of explainable AI methods, which show how the model made its choices visually. This feature is very important for getting healthcare workers to believe the model because it lets them understand and check the model's results, which makes it easier to use in clinical settings. By drawing attention to important parts of the X-ray pictures, our method not only helps doctors make correct diagnoses but also teaches them. The results show that deep learning has the ability to change how lung diseases are diagnosed. It is a flexible and efficient approach that can be especially useful in places with limited resources. Radiologists can focus on more difficult cases and provide better healthcare generally when automated tools make their jobs a lot easier. In the future, researchers should work on adding more lung diseases and demographic groups to the dataset to make sure that the model is stable and can be used with a wide range of people. Adding different types of data, like information from a patient's medical history and lab tests, could also help doctors make more accurate diagnoses and get a fuller picture of their situation.

REFERENCES

[1] Çallı, E.; Sogancioglu, E.; van Ginneken, B.; van Leeuwen, K.G.; Murphy, K. Deep learning for chest X-ray analysis: A survey. Med. Image Anal. 2021, 72, 102125.

[2] Wang, L.; Lin, Z.Q.; Wong, A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci. Rep. 2020, 10, 19549.

[3] Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; Xia, L. Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. Radiology 2020, 296, E32–E40.

[4] Fang, Y.; Zhang, H.; Xie, J.; Lin, M.; Ying, L.; Pang, P.; Ji, W. Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR. Radiology 2020, 296, E115–E117.

[5] Jacobi, A.; Chung, M.; Bernheim, A.; Eber, C. Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review. Clin. Imaging 2020, 64, 35–42.

[6] Pal, B.; Gupta, D.; Rashed-Al-Mahfuz, M.; Alyami, S.A.; Moni, M.A. Vulnerability in Deep Transfer Learning Models to Adversarial Fast Gradient Sign Attack for COVID-19 Prediction from Chest Radiography Images. Appl. Sci. 2021, 11, 4233.

[7] Rajaraman, S.; Guo, P.; Xue, Z.; Antani, S.K. A Deep Modality-Specific Ensemble for Improving Pneumonia Detection in Chest X-rays. Diagnostics 2022, 12, 1442.

[8] Kundu, R.; Das, R.; Geem, Z.W.; Han, G.-T.; Sarkar, R. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. PLoS ONE 2021, 16, e0256630.

[9] Mousavi, Z.; Shahini, N.; Sheykhivand, S.; Mojtahedi, S.; Arshadi, A. COVID-19 detection using chest X-ray images based on a developed deep neural network. SLAS Technol. 2022, 27, 63–75.

[10] Cha, S.-M.; Lee, S.-S.; Ko, B. Attention-Based Transfer Learning for Efficient Pneumonia Detection in Chest X-ray Images. Appl. Sci. 2021, 11, 1242.

[11] Siddiqi, R. Fruit-classification model resilience under adversarial attack. SN Appl. Sci. 2021, 4, 31.

[12] Chlap, P.; Min, H.; Vandenberg, N.; Dowling, J.; Holloway, L.; Haworth, A. A review of medical image data augmentation techniques for deep learning applications. J. Med. Imaging Radiat. Oncol. 2021, 65, 545–563.

[13] Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.-Y. Generative adversarial networks: Introduction and outlook. IEEECAA J. Autom. Sin. 2017, 4, 588–598.

[14] Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. IEEE Trans. Knowl. Data Eng. 2021, 35, 3313–3332.

[15] Gulakala, R.; Markert, B.; Stoffel, M. Rapid diagnosis of Covid-19 infections by a progressively growing GAN and CNN optimisation. Comput. Methods Programs Biomed. 2023, 229, 107262.

[16] Gulakala, R.; Markert, B.; Stoffel, M. Generative adversarial network based data augmentation for CNN based detection of COVID-19. Sci. Rep. 2022, 12, 19186.

[17] Loey, M.; Smarandache, F.; Khalifa, N.E.M. Within the Lack of Chest COVID-19 X-ray Dataset: A Novel Detection Model Based on GAN and Deep Transfer Learning. Symmetry 2020, 12, 651.

[18] Jabbar, A.; Li, X.; Omar, B. A Survey on Generative Adversarial Networks: Variants, Applications, and Training. ACM Comput. Surv. 2022, 54, 1–49.

[19] Aggarwal, P.; Mishra, N.K.; Fatimah, B.; Singh, P.; Gupta, A.; Joshi, S.D. COVID-19 image classification using deep learning: Advances, challenges and opportunities. Comput. Biol. Med. 2022, 144, 105350.

[20] Rajawat, N.; Hada, B.S.; Meghawat, M.; Lalwani, S.; Kumar, R. C-COVIDNet: A CNN Model for COVID-19 Detection Using Image Processing. Arab. J. Sci. Eng. 2022, 47, 10811–10822.

[21] Sedik, A.; Hammad, M.; Abd El-Samie, F.E.; Gupta, B.B.; Abd El-Latif, A.A. Efficient deep learning approach for augmented detection of Coronavirus disease. Neural Comput. Appl. 2022, 34, 11423–11440.

[22] Subramanian, N.; Elharrouss, O.; Al-Maadeed, S.; Chowdhury, M. A review of deep learning-based detection methods for COVID-19. Comput. Biol. Med. 2022, 143, 105233.