# Explainable AI for Chest Disease Diagnosis: A Grad-CAM Enhanced EfficientNetV2S Approach

**Anuranj VV**

# 1. INTRODUCTION

Pulmonary diseases like Pneumonia and COVID-19 continue to affect millions of people worldwide and remain a major public health concern. These diseases mainly impact the lungs, leading to breathing issues and serious health risks if not diagnosed early. Quick and accurate detection is very important for proper treatment and recovery. Among available diagnostic methods, chest X-ray imaging is one of the most used because it is affordable, fast, and easily available, even in low-resource healthcare settings [1].

However, interpreting chest X-rays manually is not always reliable. Different doctors may give different diagnoses for the same image, and some early signs of disease can be very subtle and easy to miss [2]. These challenges have encouraged researchers to use artificial intelligence (AI), especially deep learning, to build systems that can automatically analyse medical images and support faster, more consistent diagnosis.

Convolutional Neural Networks (CNNs) are a popular type of deep learning model that can learn features directly from images. They have been used successfully in general image recognition tasks. Models like VGG16, ResNet, and Inception have shown strong performance. But in the medical field, these models face limitations. They often miss small or unclear features, which are common in chest X-rays, and their decision-making process is hard to understand for medical professionals [3][4].

To overcome these problems, this study proposes an enhanced deep learning pipeline that is specially designed for classifying chest X-rays into three categories: Pneumonia, COVID-19, and Normal. The system uses EfficientNetV2S, an advanced CNN that improves on earlier models by adjusting depth, width, and resolution efficiently. It performs well in terms of speed and accuracy, making it suitable for real-time use in hospitals [5].

Before sending images to the model, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used to improve contrast and reveal fine details in the X-rays. This helps the model better detect patterns related to disease while avoiding the over-enhancement of noise [6].

To make the system more trustworthy and transparent, Gradient-weighted Class Activation Mapping (Grad-CAM) is added. Grad-CAM creates heatmaps on the X-ray that show which areas the model focused on while making its prediction. This allows doctors to see whether the AI is looking at the correct regions, helping them to verify its

decisions [7].

The model is trained on a balanced dataset that contains an equal number of images for all three classes. This prevents the model from favoring any particular category and helps it learn equally from all types. In testing, the model achieved an AUC score of 0.98, showing it can accurately distinguish between the three conditions.

In conclusion, this work presents a reliable, efficient, and explainable AI model that can assist medical professionals in diagnosing chest-related diseases. The combination of advanced neural networks, image enhancement techniques, and explainability tools helps make the system both accurate and practical for real-world clinical use.

## 2. LITERATURE REVIEW

In recent years, the use of deep learning in the field of medical imaging has significantly expanded, particularly for the diagnosis of pulmonary diseases using chest radiographs. Chest X-rays are widely used because they are affordable, fast, and non-invasive. These advantages have made them a common imaging technique for detecting diseases like Pneumonia and COVID-19. Early research in automated chest X-ray classification relied heavily on traditional Convolutional Neural Networks (CNNs) such as VGGNet and ResNet, which showed promising results in image classification tasks [8].

One of the landmark studies in this area was conducted by Rajpurkar et al., who introduced CheXNet, a deep learning model based on a 121-layer DenseNet architecture. Trained on the ChestX-ray14 dataset, CheXNet achieved performance comparable to practicing radiologists in diagnosing pneumonia [9]. Despite its success, the model operated as a "black box," providing little explanation for how predictions were made—limiting its adoption in clinical environments where interpretability is critical.

To overcome the lack of transparency in CNN-based models, researchers introduced explainable AI (XAI) methods, with Gradient-weighted Class Activation Mapping (Grad-CAM) being one of the most widely used. Grad-CAM generates visual heatmaps that indicate which regions of the image influenced the model's prediction the most. This technique brought more trust and usability to AI systems in healthcare by helping clinicians understand the reasoning behind predictions [10].

Another area of focus in the literature has been image preprocessing. Since the quality of input images plays a vital role in model performance, several studies explored enhancement methods to improve feature visibility. One of the most effective techniques is Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE improves the local contrast of medical images, making subtle anatomical features more prominent. Studies have shown that applying CLAHE before model training can significantly boost the performance of deep learning models in detecting lung abnormalities [11].

In terms of model architecture, the field has seen major advancements with the introduction of EfficientNet and its successor EfficientNetV2. These models use a technique called compound scaling, which systematically adjusts the depth, width, and resolution of the network to optimize both accuracy and efficiency. EfficientNetV2 in

particular provides faster training and improved accuracy with fewer parameters, making it well-suited for high-resolution medical imaging tasks [12].

While many past studies have applied CNNs for classifying chest X-ray images, relatively few have integrated preprocessing, high-performance models, and interpretability techniques into a single unified framework. Some attempts have been made to apply explainable AI for COVID-19 detection, but these approaches often faced limitations like inconsistent results across datasets or imbalanced class distributions, which reduced their reliability in real-world scenarios [13].

Building upon these prior works, this study proposes a complete framework that combines CLAHE preprocessing, EfficientNetV2S for feature extraction and classification, and Grad-CAM for visual interpretability. By addressing both accuracy and explainability, the framework aims to not only improve classification performance but also build confidence among healthcare professionals by offering insights into how predictions are made.

## 3. MATERIALS AND METHODS

The proposed system presents a comprehensive and explainable deep learning pipeline designed for the automatic classification of pulmonary diseases—specifically Pneumonia, COVID-19, and Normal chest conditions. The entire framework is built to maximize both classification accuracy and interpretability, which are crucial for deploying AI in real-world clinical practice. As illustrated in Figure 1, the framework is divided into four main components:

1. Image Preprocessing Module
2. Deep Feature Extraction Module
3. Classification Module
4. Explainability Module.

Each component plays a crucial role in transforming raw chest X-ray images into interpretable diagnostic predictions.

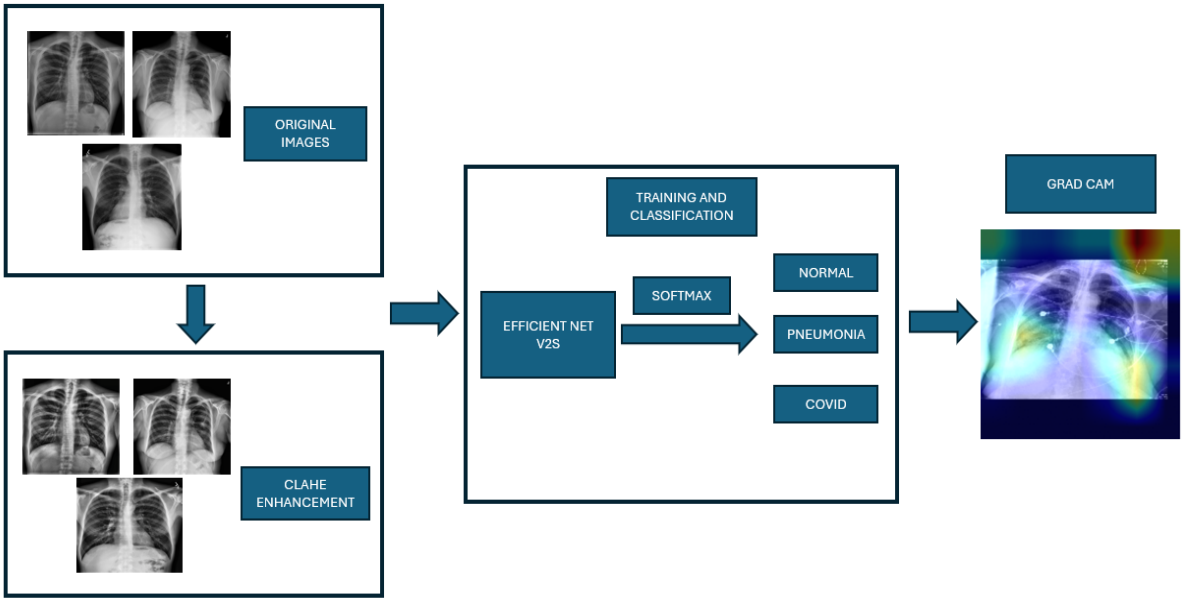



Figure 1 Proposed Framework

In the initial stage of the proposed diagnostic pipeline, raw chest X-ray (CXR) images are collected and subjected to meticulous preprocessing to improve their visual quality and ensure consistency across the dataset. Medical images often vary significantly in terms of brightness, contrast, and noise levels, which can obscure diagnostically important features. To address these challenges, Contrast Limited Adaptive Histogram Equalization (CLAHE) is employed. CLAHE operates by enhancing the local contrast

in regions of the image, particularly in areas where variations in intensity are subtle, thereby accentuating critical pathological structures such as opacities, consolidations, and lesions. This step is essential for ensuring that fine-grained features are preserved and more readily detectable in subsequent stages [14].

Following the enhancement phase, the pre-processed images are passed through a deep feature extraction module, which leverages the capabilities of the EfficientNetV2S architecture. EfficientNetV2S, a lightweight yet powerful convolutional neural network pretrained on the ImageNet dataset, has been selected due to its ability to balance accuracy and computational efficiency. As the images traverse through its layers, the model captures a hierarchical set of features—starting from simple edges and gradients in the shallow layers to more complex structures and semantic patterns in the deeper layers. This enables the model to form discriminative representations of various pulmonary conditions, including Pneumonia, COVID-19, and Normal cases [15].

The extracted feature vectors are then routed to a custom-designed classification module. This module consists of several fully connected (dense) layers, interspersed with batch normalization layers to standardize activations and facilitate faster convergence. Additionally, dropout layers are incorporated to prevent overfitting by randomly deactivating neurons during training. At the final stage, a softmax activation function outputs the class probabilities for the three categories. The model is optimized using the Adam optimizer, well-known for its adaptive learning rate mechanism, and is trained with the categorical cross-entropy loss function, which is suitable for multi-class classification problems. This configuration ensures stable learning dynamics and robust classification performance [16].

To promote interpretability and enhance clinical trust in the AI-driven diagnostic system, an explainability module based on Gradient-weighted Class Activation Mapping (Grad-CAM) is integrated. Grad-CAM generates visual saliency maps by leveraging the gradients flowing into the final convolutional layers. These maps highlight the specific regions within the X-ray images that have the greatest influence on the model's predictions. As a result, clinicians can visually verify that the model is focusing on medically relevant areas, such as regions with visible infiltrates or lung opacities, thereby offering a transparent decision-making process and facilitating clinical adoption [17].

## 4.1. IMAGE PREPROCESSING MODULE

This module is responsible for enhancing image quality and preparing the data for deep learning. To improve the visibility of lung structures and subtle pathological patterns, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied. CLAHE helps in boosting local contrast while suppressing noise amplification. Following enhancement, all images are resized to a fixed resolution suitable for the input dimensions of the neural network and normalized to scale pixel intensities between 0 and 1. This preprocessing step ensures consistent image quality and stable model performance across the dataset.

$$\text{Normalized value} = \frac{\text{rescaled value} - 0.5}{0.5} \qquad (1)$$

| Method | Advantages | Suitability for Xray's | Reason for Selected/Not Selected |
|---|---|---|---|
| CLAHE (Contrast-Limited Adaptive Histogram Equalization) | Prevents over-enhancement, robust for noisy images | High | Enhances local contrast without amplifying noise |
| Histogram Equalization | Enhances global contrast, improves visibility of large-scale patterns | High | Can over-enhance noise, create artifacts |
| Gaussian Filtering | Reduces noise, smoothens images | Medium | Can blur fine details, reduce diagnostic features |

Table 1 Comparison of Image Enhancement Techniques for Medical Imaging.

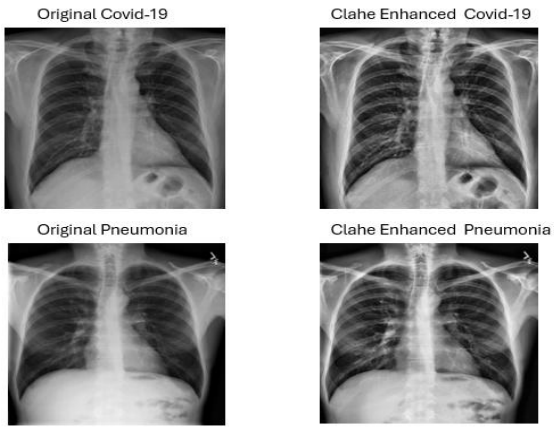



Figure 2 Results of Clahe enhancement on the images.

To improve the visual quality of chest X-ray images and assist the deep learning model in accurate disease detection, various image enhancement techniques were evaluated. These methods were tested after applying initial resizing and normalization steps. The goal was to select the approach that best balances noise reduction, feature visibility, and preservation of critical diagnostic details. The three enhancement methods compared include Contrast-Limited Adaptive Histogram Equalization (CLAHE), Histogram Equalization, and Gaussian Filtering.

CLAHE (Contrast-Limited Adaptive Histogram Equalization) emerged as the most effective method. CLAHE works by dividing the image into smaller regions called tiles and applying histogram equalization to each of them independently. It prevents the over-amplification of noise by applying a limit to the contrast enhancement (clip limit) and then combining the tiles smoothly. This localized contrast adjustment helps in enhancing fine structural details such as opacities, nodules, or mild infiltrations in chest X-rays—details that are often crucial for diagnosing conditions like Pneumonia or COVID-19. Because of its ability to boost important features without introducing excessive artifacts, CLAHE was selected as the main preprocessing technique in this project [18].

Histogram Equalization, in contrast, applies a global transformation that adjusts the entire image histogram for improved contrast. While this can make large-scale patterns more visible, it frequently causes over-enhancement in uniform regions and tends to exaggerate noise. In medical imaging, this over-amplification can result in loss of subtle but diagnostically significant details. As such, while histogram equalization might be suitable for general photography or natural images, it is not ideal for enhancing chest X-rays for medical analysis [19].

Gaussian Filtering is another common image processing technique primarily used for noise suppression and smoothing. This method applies a Gaussian kernel to blur the image, effectively reducing high-frequency noise. However, the trade-off is that it also softens the edges and fine structures, which are precisely the elements needed for accurate medical diagnosis. In the context of pulmonary disease detection, blurring lung boundaries or lesion outlines can significantly impair the model's ability to learn and differentiate patterns. Thus, despite its usefulness in other domains, Gaussian filtering was not adopted in this study [20].

After careful comparison, CLAHE was chosen as the optimal enhancement method. It offers a good compromise between improving image clarity and preserving detailed

anatomical information. Its tile-wise contrast adjustment aligns well with the need for highlighting subtle features in X-ray scans, making it especially valuable for deep learning models focused on medical diagnostics.

## 4.2. DEEP FEATURE EXTRACTION MODULE

The process of deep feature learning is performed using the EfficientNetV2S model, an advanced convolutional neural network (CNN) architecture that has demonstrated strong performance across a wide range of computer vision tasks. EfficientNetV2S is part of the EfficientNetV2 family, which is known for its innovative compound scaling method that jointly optimizes the network's depth, width, and input resolution to achieve an optimal balance between computational efficiency and predictive accuracy. Unlike conventional CNNs that scale one dimension at a time, EfficientNetV2S applies a more holistic scaling approach, allowing it to perform faster and more accurately even on relatively constrained hardware resources.

In the context of medical image analysis, particularly chest X-rays, EfficientNetV2S serves as a powerful feature encoder. When fed with pre-processed radiographic images, the network extracts multi-scale, hierarchical features—ranging from low-level edge and texture details to high-level semantic structures associated with pathological patterns. This deep feature extraction is crucial for identifying nuanced abnormalities such as ground-glass opacities, interstitial markings, or lung consolidations, which are often subtle and easily missed in traditional analysis.

By leveraging transfer learning, the pretrained weights of EfficientNetV2S—originally learned from the extensive and diverse ImageNet dataset—allow the model to generalize well even when applied to domain-specific datasets with limited labelled data. This significantly reduces the computational burden and time typically required for training deep models from scratch, while also enhancing model robustness and performance in real-world clinical scenarios [21].

After preprocessing, the enhanced chest X-ray images are passed through the deep feature extraction module, which is built using the EfficientNetV2S architecture. The complete structure of this module is summarized below:

**Input Layer:**

The input layer of the model takes chest X-ray images that have been resized to a fixed

size of 224x224 pixels with 3 channels (RGB). This resizing ensures that all images match the expected input shape of the EfficientNetV2S model, which was originally trained on ImageNet data of the same size. Standardizing the image dimensions prevents size-related errors during processing and allows the model to efficiently extract features across all samples.

Before the images are passed into the model, their pixel values are normalized to a range between 0 and 1. This is done by dividing the pixel intensity by 255. Normalization helps the model learn more effectively by keeping the input values in a stable range, which leads to faster convergence and improved accuracy. These two preprocessing steps—resizing and normalization—ensure that the model can focus on learning patterns related to diseases rather than dealing with differences in brightness or scale across images [22].

**EfficientNet V2S Base Model:**

The EfficientNetV2S model is used as the core feature extractor in this classification framework. This architecture is a compact and high-performance variant of the EfficientNet family, designed to deliver better accuracy with faster training speed and fewer parameters. In this study, the pretrained version of EfficientNetV2S, trained on the large-scale ImageNet dataset, is used to take advantage of the rich feature representations it has already learned from millions of natural images. Although these images are not medical in nature, the general visual features—such as edge detectors, shape patterns, and texture identifiers—transfer well to medical tasks like X-ray analysis through a process known as transfer learning [23].

As chest X-ray images are passed through the EfficientNetV2S model, the network processes them in several convolutional layers arranged in blocks. These layers capture different levels of information: early layers extract basic patterns such as lines, corners, and textures, while deeper layers detect more complex features like shapes and abnormalities commonly seen in lung infections or diseases. The network is able to learn these features in a hierarchical way, meaning it builds more meaningful representations with each layer. By the end of the EfficientNetV2S base, the model outputs a feature map of size (7, 7, 1280), which effectively condenses all the important visual information in the image into a rich set of feature descriptors. These extracted features are then passed to the next layers of the network for classification.

**Global Average Pooling Layer:**

After the EfficientNetV2S model extracts a detailed set of feature maps with dimensions 7×7×1280, a Global Average Pooling (GAP) layer is applied to reduce these into a simpler and more manageable form. Instead of flattening the full feature map into a long vector, which would greatly increase the number of parameters, the GAP layer calculates the average value of each individual feature channel. This results in a 1-dimensional vector of length 1280, where each value represents the average activation of a specific feature map across its spatial dimensions.

Using GAP helps the model retain the most essential information from each feature channel while significantly reducing computational load. It also minimizes the risk of overfitting, especially in deep networks, by reducing the number of trainable weights in the following layers. Additionally, because GAP preserves only the most global patterns, it enables the model to stay focused on important visual cues related to disease without being distracted by minor or noisy image variations. This makes it highly effective in medical image classification tasks where precision and efficiency are critical [24].

**Batch Normalization:**

After the global average pooling step, the resulting feature vector is passed through a Batch Normalization layer. This layer helps improve the model's learning by normalizing the values in the vector. It does this by adjusting the outputs so that they have a mean close to zero and a standard deviation close to one. This makes the training process more stable and helps the model learn faster.

Batch normalization also helps the model perform better across different batches of data by reducing the effect of changes in data distribution. It allows the model to use higher learning rates and reduces the chances of the model getting stuck during training. Overall, it plays an important role in improving the training efficiency and general performance of deep neural networks [25].

**Dense Layer:**

The output from the batch normalization layer is passed into a Dense (fully connected) layer that contains 128 neurons. This layer plays an important role in learning non-linear combinations of features extracted from the earlier layers. By using the ReLU (Rectified Linear Unit) activation function, the model can better capture complex patterns and

relationships in the data that are necessary to tell the difference between Pneumonia, COVID-19, and Normal chest X-rays.

To make the model more robust and reduce the chance of overfitting, L2 regularization is applied to the weights in this layer. L2 regularization works by adding a small penalty to the loss function whenever the model uses very large weights. This encourages the model to find simpler, more general solutions that work well not just on the training data, but also on new, unseen data [26]. This step helps improve the model's ability to generalize across different patients and image sources, which is especially important in medical diagnosis.

**Batch Normalization (after Dense Layer):**

After the first dense layer with ReLU activation, a second Batch Normalization layer is added. This layer helps keep the output values from the dense layer in a consistent range. By normalizing the activations again, the model remains stable and less sensitive to weight changes during training[27].This stability makes learning faster and helps avoid problems such as vanishing or exploding gradients that can occur in deep networks.

**Dropout Layer (30% rate):**

This Layer is applied with a dropout rate of 30% (0.3). This means that during each training step, 30% of the neurons in the dense layer are randomly turned off. Dropout is a common technique used to prevent overfitting—where the model performs well on training data but poorly on new data. By randomly deactivating neurons, the model is encouraged to learn more diverse and general patterns rather than relying on specific neurons, making it more robust and better suited for unseen images [28].

**Output Layer:**

The model ends with a final Dense output layer consisting of three neurons, each representing one of the target classes: Pneumonia, COVID-19, and Normal. A Softmax activation function is used here, which converts the raw output scores into probabilities. This makes it easy to interpret the model's prediction, as the highest probability indicates the most likely class. Softmax ensures that the output values are between 0 and 1 and sum to 1, which is ideal for multi-class classification tasks like this one [29].

| Layer Type | Parameters | Output Shape | Description |
|---|---|---|---|
| Input Layer | - | (224,224, 3) | Resized and normalized chest X-ray image |
| EfficientNetV2S (Base) | Pretrained on ImageNet | (7,7,1280) | Extracts hierarchical features (edges, textures, shapes) |
| GlobalAveragePooling2D | - | (1280,) | Averages spatial dimensions into a feature vector |
| Batch Normalization | - | (1280,) | Stabilizes and accelerates training by normalizing feature distributions |
| Dense Layer (128 units, ReLU, L2) | 163,968 | (128,) | Learns non-linear high-level feature representations with L2 regularization |
| Batch Normalization | - | (128,) | Further normalizes activations after dense layer |
| Dropout Layer (rate=0.3) | - | (128,) | Randomly drops 30% of units to reduce overfitting |
| Output Layer (3 units, SoftMax) | 387 | (3,) | Produces class-probability scores for: Pneumonia, COVID-19, and Normal |

Table 2 Model Architecture Summary

## 4.3. CLASSIFICATION MODULE

Once the deep features are extracted by the convolutional layers of the model, they are passed into a custom classification module. This part of the model is specifically designed to perform multi-class classification across three categories: Pneumonia, COVID-19, and Normal. The goal is to accurately predict the correct disease class based on the rich feature representations produced by the EfficientNetV2S backbone.

The classification head starts with a Dense (fully connected) layer that takes the pooled and normalized feature vector as input. This layer combines the extracted features using learned weights to discover meaningful patterns and relationships between them. These patterns help the model detect even small differences between disease types, which is important in medical diagnosis. To avoid overfitting, a Dropout layer is included after the Dense layer. This technique randomly turns off a portion of the neurons during training, which prevents the model from becoming too dependent on specific neurons. As a result, the model learns more general features, making it perform better on unseen data [30].

At the final stage of the module, a Softmax activation function is applied to the output layer. This converts the raw scores into probability values, ensuring that all the predicted probabilities add up to 1. The class with the highest probability is selected as the model's prediction. Softmax is ideal for multi-class classification problems like this one, where each image must belong to one of three possible categories.

To train the model, a Categorical Cross-Entropy loss function is used. This loss measures how far the predicted probability distribution is from the actual class label. If the model predicts the wrong class, the loss increases, guiding the optimizer to adjust the weights and improve future predictions. Since each X-ray belongs to exactly one category, this loss function is a suitable choice. For optimization, the model uses the Adam optimizer, which is a powerful method that adapts the learning rate during training. It combines the benefits of momentum and RMSProp, making the training faster and more stable, even when the dataset is noisy or has sparse gradients [31].

In summary, the classification module effectively transforms the deep features into meaningful class predictions. With the help of regularization, probability-based output, and robust training strategies, the model becomes capable of making reliable and accurate decisions—a critical requirement for any clinical decision-support system.

## 4.4. PREDICTION MODULE

This module plays a vital role in evaluating new chest X-ray images and identifying the most probable disease class: Pneumonia, COVID-19, or Normal. It simulates how the trained model would be used in a real-world clinical setting. The process begins by selecting an external chest X-ray image from a local path. This image is passed through a preprocessing function to ensure consistency with the training data.

In preprocessing, the image is first read and resized to 224×224 pixels—the input size expected by the model. Next, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to enhance local contrast, improving the visibility of fine features such as opacities, lung texture, and consolidation patterns. CLAHE is particularly effective for medical images where subtle differences are critical. After contrast enhancement, pixel values are normalized to the [0, 1] range for computational efficiency and consistency. A batch dimension is then added to the image so it can be properly interpreted by the deep learning model during inference [32][33].

Once the image has been pre-processed, it is passed through the trained model, which returns probability scores for each of the three classes. The class with the highest probability is selected as the model's prediction. Alongside the predicted label, the confidence values for all three classes are displayed, allowing clinicians to interpret how certain the model is in its prediction. For example, if the model returns a 0.9976 confidence score for “Pneumonia” and low scores for the other two classes, it strongly suggests the model is confident in its decision [34].

This output is not only informative but also supports transparent decision-making in clinical workflows. By showing class-wise confidence levels, the module allows healthcare professionals to better understand the reasoning behind the prediction. This transparency enhances trust and aids in making well-informed diagnostic decisions. The prediction module, therefore, ensures that the framework is not only technically sound but also clinically practical [35].

## 4.5. EXPLAINABILITY MODULE

To improve the interpretability and transparency of the model's predictions, the proposed framework integrates an Explainability Module based on Gradient-weighted Class Activation Mapping (Grad-CAM). This technique plays a key role in medical AI applications, where it's not only important to predict outcomes but also to explain how those predictions are made. Grad-CAM helps achieve this by generating visual heatmaps over chest X-ray images that show which areas the model focused on while making its prediction. These heatmaps help clinicians and radiologists visually confirm whether the model's decision is supported by medical evidence in the image [36].

Grad-CAM works by computing the gradients of a specific output class (e.g., Pneumonia, COVID-19, or Normal) with respect to the final convolutional feature maps of the model. These gradients are then used to weigh the importance of each feature map, which are combined to create a coarse localization map—or heatmap. This map is then overlaid on the original image, showing the regions that contributed most to the model's classification. The beauty of Grad-CAM is that it can be applied to any CNN-based architecture without needing to change the model's structure. The resulting visualizations are intuitive and human-understandable, making it easier for non-technical medical professionals to engage with AI tools [37].

**Importance of Grad-CAM**

In clinical environments, simply showing a disease prediction is not enough—doctors need to understand the reasoning behind it. Deep learning models are often considered "black boxes" because their internal decision-making process is hard to interpret. Grad-CAM addresses this limitation by providing a visual explanation of what influenced the model's decision. By highlighting disease-specific areas on chest X-rays, it builds confidence among clinicians, helping them validate if the model is looking at medically relevant regions before making a diagnosis [38].

**Need for Grad-CAM**

**Transparency:** Grad-CAM makes the model's decision-making process more understandable. By showing which regions were most influential in a prediction, it reduces the black-box nature of deep learning and increases the interpretability of AI models.

**Clinical Validation**: Medical professionals can visually compare Grad-CAM heatmaps

with known signs of disease. This ensures that the model is not just making accurate predictions, but that it's doing so for the right reasons, improving trust and safety.

**Error Analysis**:Grad-CAM is also helpful when the model makes incorrect predictions. It allows researchers to check whether the model was focusing on misleading areas, which can guide further improvements and data correction.

**Trust Building**: For AI systems to be accepted in healthcare, they must be transparent. Grad-CAM allows clinicians to see and understand how a prediction was made, building trust between human users and AI tools.

**Deployment Readiness**: Regulatory agencies and hospitals increasingly require explainable AI before adopting systems in real-world practice. Grad-CAM makes the model more suitable for such environments by offering visual evidence for its predictions [39].

**Grad-CAM Visualization:**

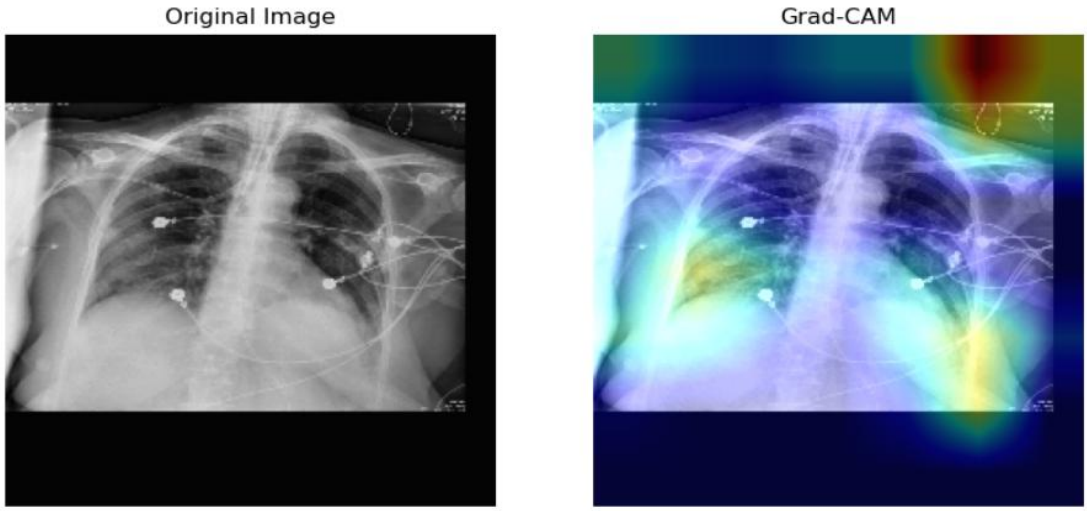



Figure 3 Grad-CAM Visualization

**Figure 3** displays a side-by-side comparison between an original chest X-ray image and its corresponding Grad-CAM visualization, offering insights into how the deep learning model interprets the image for classification purposes. On the left side, the original X-ray reveals the raw radiological structure of the patient's chest, showing anatomical features without any model-inferred annotations.

On the right, the Grad-CAM heatmap overlays coloured regions on the X-ray,

highlighting areas that had the most influence on the model's decision. Warmer colours—such as red, orange, and yellow—represent regions of high importance, where the model detected critical features related to the predicted disease class. In contrast, cooler colours like green and blue represent less relevant areas. As observed in the figure, the model attends to specific zones within the lungs, particularly where radiographic signs such as opacities, consolidations, or infiltrates typically appear. These are clinical indicators commonly associated with Pneumonia or COVID-19[40]. This visualization reinforces the model's reliability by showing that its focus aligns with regions that are clinically significant for disease diagnosis. By using Grad-CAM, medical professionals gain more confidence in the system's decisions, as they can visually interpret and verify whether the prediction is based on medically valid patterns. Thus, Grad-CAM serves as a valuable interpretability tool that bridges the gap between automated predictions and clinical validation, promoting trust and usability in real-world healthcare environments.

## 4. DATASET DESCRIPTION

To build an accurate and reliable model for detecting pulmonary diseases, a new and balanced dataset named CONOP Dataset (COVID-19, Normal, and Pneumonia) was created for this study. Instead of depending on a single data source, images were collected from many trusted public repositories. This helped include a wide variety of X-ray images taken under different conditions and from different patients. Such diversity improves the model's ability to work well on real-world medical data.

Due to the limited availability of large COVID-19 chest X-ray datasets, images were gathered from multiple sources. A total of 1,401 original COVID-19 X-ray images were collected from open-access repositories like GitHub [41], Radiopaedia, the Italian Society of Radiology (SIRM) , and Figshare. To further expand the dataset, 912 more images were taken from Mendeley, which were already augmented. This helped increase the number of samples without needing to manually apply image transformations.

In addition to COVID-19 images, 2,313 images of Pneumonia and 2,313 images of Normal (healthy) cases were collected from well-known Kaggle datasets. All the images used in the dataset were in posteroanterior (PA) view, which is a standard chest X-ray view used by doctors for accurate diagnosis.

The final CONOP Dataset consists of 6,939 chest X-ray images, with exactly 2,313 images for each class—COVID-19, Pneumonia, and Normal. This equal distribution avoids any class imbalance, which is important to train a fair model that doesn't favour one category over another. A balanced dataset ensures that the model learns from all classes equally and performs well across all types of diseases.

By collecting images from multiple trusted sources and maintaining consistent standards like PA view and balanced classes, the CONOP Dataset provides a strong and diverse foundation for training deep learning models. It helps the model to generalize better, making it more useful in real clinical settings.

| Types of Disease | Total Number Of Images/Class | Original Dataset Used Image | Training Set | Test Set |
|---|---|---|---|---|
| Normal | 2313 | 2313 | 1850 | 463 |
| Pneumonia | 2313 | 2313 | 1850 | 463 |
| Covid-19 | 2313 | 2313 | 1850 | 463 |

Table 3 Dataset Configuration

To ensure fair training and evaluation, the CONOP Dataset was organized with a balanced number of images for each disease class: Normal, Pneumonia, and COVID-19. Each category includes exactly 2,313 chest X-ray images, maintaining equal representation across all classes. This uniform distribution is essential for preventing bias during model training, where one class could otherwise dominate and skew the results.

For model development, the dataset was divided into training and testing sets. From each class, 1,850 images were used for training, allowing the model to learn distinguishing features across conditions. The remaining 463 images per class were used for testing, helping to objectively evaluate how well the model performs on unseen data. This clear and consistent split supports robust model validation and helps assess its real-world diagnostic potential more accurately.

## 5. EVALUATION METRICS

To properly assess how well the proposed model performs in classifying pulmonary diseases, several important evaluation metrics were used. These metrics help measure both the general accuracy of the model and how well it handles each disease class individually. By using a combination of these metrics, we can get a complete picture of the model's strengths, such as how often it makes correct predictions, and identify areas where it might need improvement. This thorough evaluation ensures the model is not just accurate overall but also reliable and fair across all types of cases.

**Accuracy**:

Accuracy shows how often the model makes the right prediction. It is calculated by dividing the number of correct predictions by the total number of predictions. In simple terms, it tells us how many times the model got the answer right out of all the times it tried. While high accuracy usually means the model is performing well, it doesn't always give the full picture—especially when some classes have more examples than others.

The formula for accuracy is:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2)$$

In terms of classification:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Where:

TP = True Positives (correctly predicted positives)

TN = True Negatives (correctly predicted negatives)

FP = False Positives (wrongly predicted positives)

FN = False Negatives (wrongly predicted negatives)

**Precision**:

Precision tells us how many of the predictions made as "positive" by the model were actually correct. It only looks at the cases where the model said a disease was present. In medical settings, high precision is very important because it means the model is rarely wrong when it predicts a disease. This helps prevent giving patients unnecessary treatments or causing worry with incorrect results.

The formula for precision is:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (4)$$

Where:

TP: Cases where the model correctly predicted the disease.

FP: Cases where the model wrongly predicted the disease when there was none.

**Recall:**

Recall (also called Sensitivity or True Positive Rate) shows how well the model finds actual positive cases. It tells us how many real disease cases the model was able to correctly identify. In medical diagnosis, having high recall is very important because missing a real case—like not detecting a patient with COVID-19—can delay treatment and put others at risk. So, recall helps measure the model's ability to catch all the true disease cases.

The formula for the Recall is:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (5)$$

**F1-Score**:

This is a helpful metric that combines both Precision and Recall into one value. It is calculated using the harmonic mean of precision and recall, which means it balances the importance of both. This score is especially useful when the number of samples in each class is not equal or when it is important to reduce both false positives and false

negatives.

In medical diagnosis, the F1-Score plays an important role because both types of mistakes—missing a real disease (false negative) and predicting a disease that isn't there (false positive)—can be harmful. A high F1-Score means the model is good at correctly identifying disease cases while also avoiding wrong predictions. This helps make the model more reliable and safer for use in real-world healthcare situations.

The formula for the F1-Score is:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (6)$$

**AUC and AUC-ROC Curve:**

Area Under the Curve (AUC) and the Receiver Operating Characteristic (ROC) Curve are useful tools to check how well a classification model works, especially in medical cases where it's important to correctly tell apart different conditions.

The ROC Curve is a graph that shows how good the model is at separating the classes. It does this by plotting the True Positive Rate (Recall) against the False Positive Rate at different thresholds. This helps us see how the model's performance changes when we adjust the decision boundary. A model with a curve closer to the top-left corner is generally better at making correct predictions.

The False Positive Rate (FPR) is calculated as:

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \qquad (7)$$

The ROC curve shows the trade-off between sensitivity (Recall) and specificity (1 - FPR).

The Area Under the Curve (AUC) is a single scalar value that summarizes the overall ability of the model to discriminate between the classes. An AUC of 1.0 indicates perfect classification, while an AUC of 0.5 suggests no better performance than random guessing.

Interpretation of AUC values:

- AUC = 1.0 → Perfect classifier
- 0.9 ≤ AUC < 1.0 → Excellent classification
- 0.8 ≤ AUC < 0.9 → Good classification
- 0.7 ≤ AUC < 0.8 → Fair classification
- 0.6 ≤ AUC < 0.7 → Poor classification
- 0.5 ≤ AUC < 0.6 → Very poor classification

The formula to compute AUC is typically based on the trapezoidal rule, integrating the ROC curve:

$$\text{AUC} = \int_0^1 TPR(x)dx \qquad (8)$$

where TPR(x) is the True Positive Rate as a function of the False Positive Rate.

**Confusion Matrix**

A confusion matrix is a useful tool to evaluate how well a classification model is performing. Instead of just showing the overall accuracy, it gives a detailed view of the model's correct and incorrect predictions. This helps us understand not only how many predictions were right, but also where the model made mistakes.

A confusion matrix for a binary classification typically has four components:

- True Positives (TP): The model correctly predicts a positive case.
- True Negatives (TN): The model correctly predicts a negative case.
- False Positives (FP): The model wrongly predicts a positive case when it is actually negative (also known as a Type I error).
- False Negatives (FN): The model wrongly predicts a negative case when it is actually positive (also called a Type II error).

The structure of a confusion matrix

| | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | True Positive (TP) | False Negative (FN) |
| Actual Negative | False Positive (FP) | True Negative (TN) |

Table 4 Structure Of Confusion Matrix

By combining all these metrics, the evaluation not only captures the model's overall predictive strength but also highlights its precision, reliability, and robustness across different disease categories. This ensures that the system is suitable for practical deployment in real-world clinical environments where both accuracy and

trustworthiness are critical.

# 6. RESULT AND DISCUSSION

## 7.1. OVERVIEW

This chapter presents a clear and detailed explanation of the experimental results obtained during the study. The main goal is to evaluate how well the deep learning model performs in identifying different chest conditions and to compare its performance with results from previous research in the same field. The model's performance is measured using common evaluation metrics such as accuracy (how often the model is correct), precision (how many predicted positive cases are actually correct), recall (how many actual positive cases are correctly identified), F1-score (the balance between precision and recall), and AUC (Area Under the Curve, which shows how well the model separates the classes).

Along with the numerical results, various visual tools are used to better understand how the model makes its predictions. These include graphs, confusion matrices, and Grad-CAM heatmaps, which visually highlight the parts of the X-ray images that the model focused on to make its decision. These tools help explain how the model distinguishes between diseases like Pneumonia, COVID-19, and Normal cases.

By carefully analysing both the numbers and the visual results, this section also discusses what the model does well and where it may need improvement. This gives useful insight into how effective the model can be when used in real-world medical settings, especially for supporting doctors in diagnosing lung diseases.

## 7.2. MODEL PERFORMANCE AND EVALUATION

The deep learning model developed in this study showed excellent results in identifying and classifying chest X-ray images into three major categories: Pneumonia, COVID-19, and Normal. During testing on the validation dataset, the model achieved an impressive validation accuracy of 95.68%, which means that nearly 96 out of every 100 predictions made by the model were correct.

In addition to accuracy, the model also scored highly on the AUC (Area Under the Curve) metric, with a value of 0.9884. This indicates that the model is very good at distinguishing between the three classes, with very little confusion between them. A high AUC means the model can correctly tell the difference between sick and healthy

cases in most situations.

| | Precision | Recall | F1-Score |
|---|---|---|---|
| Pneumonia | 0.96 | 0.94 | 0.95 |
| Normal | 0.93 | 0.95 | 0.94 |
| Covid-19 | 0.99 | 0.98 | 0.98 |

Table 5 Classification Report Of Proposed Model

A closer look at the classification report shows how well the model performs for each category

- For the Pneumonia class, the model achieved a precision of 0.96, recall of 0.94, and an F1-score of 0.95. This means that most pneumonia cases were correctly identified, and only a few were either missing or wrongly classified as something else.
- In the Normal class the model reached a precision of 0.93 and recall of 0.95, with an F1-score of 0.94. These results show that the model did a good job of recognizing healthy X-ray images, with very few mistakes.
- For COVID-19 detection, the model showed excellent performance, with a precision of 0.99, recall of 0.98, and an F1-score of 0.98. These high scores are especially important in a healthcare setting, where accurately detecting COVID-19 is critical for proper treatment and control.

The model's **overall accuracy** across all classes was **95.68%**, which proves it can generalize well to new, unseen data. Additionally, the **macro average** of precision, recall, and F1-score was around **0.96**, showing that the model performs consistently across all categories, without favoring one class more than others.

Lastly, the **AUC score of 0.9884** highlights the model's strong ability to separate the different classes at various threshold levels. This is particularly useful in medical applications where thresholds might need to be adjusted depending on how cautious a diagnosis must be.

## 7.3. CONFUSION MATRIX INTERPRETATION

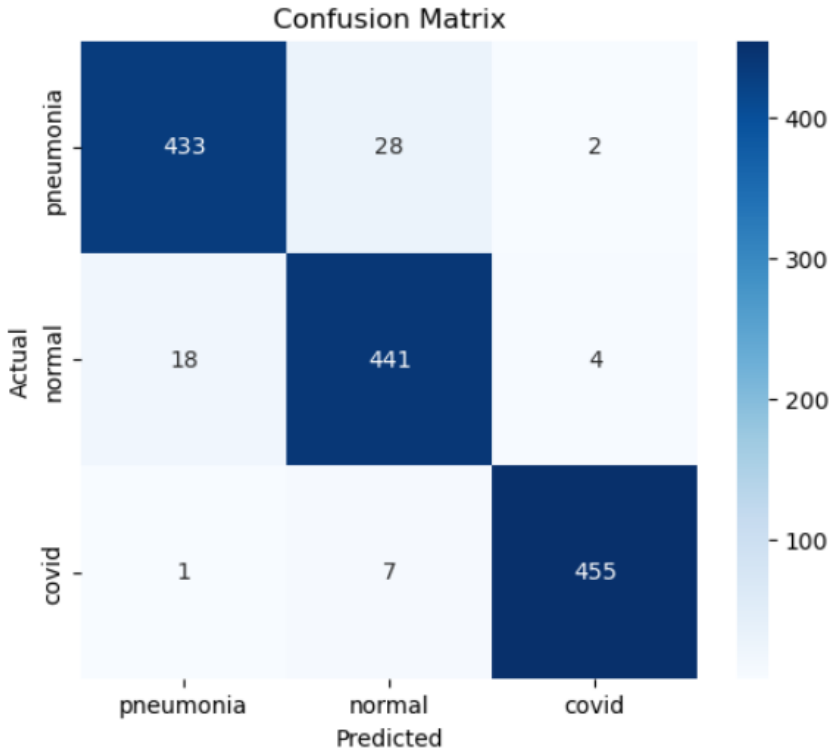



Figure 4 Confusion Matrix For Proposed Model

The confusion matrix shows how well the model predicted three classes: Pneumonia, Normal, and COVID-19. The rows represent the actual labels, while the columns show the predicted labels.

- For **Pneumonia** cases, the model correctly predicted **433** images. However, **28** cases were wrongly predicted as **Normal**, and **2** were incorrectly classified as **COVID-19**. This means the model mostly gets pneumonia cases right but sometimes confuses them with normal ones.
- In the case of **Normal** images, **441** were correctly identified. But **18** were misclassified as **Pneumonia**, and **4** as **COVID-19**. This shows that the model is fairly accurate in detecting normal cases, with only a few errors.
- For **COVID-19**, the model did very well, correctly predicting **455** cases. Only **1** was mistaken for pneumonia, and **7** for normal. This shows that the model is highly reliable in detecting COVID-19 cases.

The colour intensity of each cell correlates with the number of samples: darker shades represent higher counts, emphasizing the dominance of correct predictions along the diagonal of the matrix. Lighter shades on the off-diagonal elements highlight the relatively lower number of misclassifications.

## 7.4. ROC CURVE AND AUC PERFORMANCE ANALYSIS

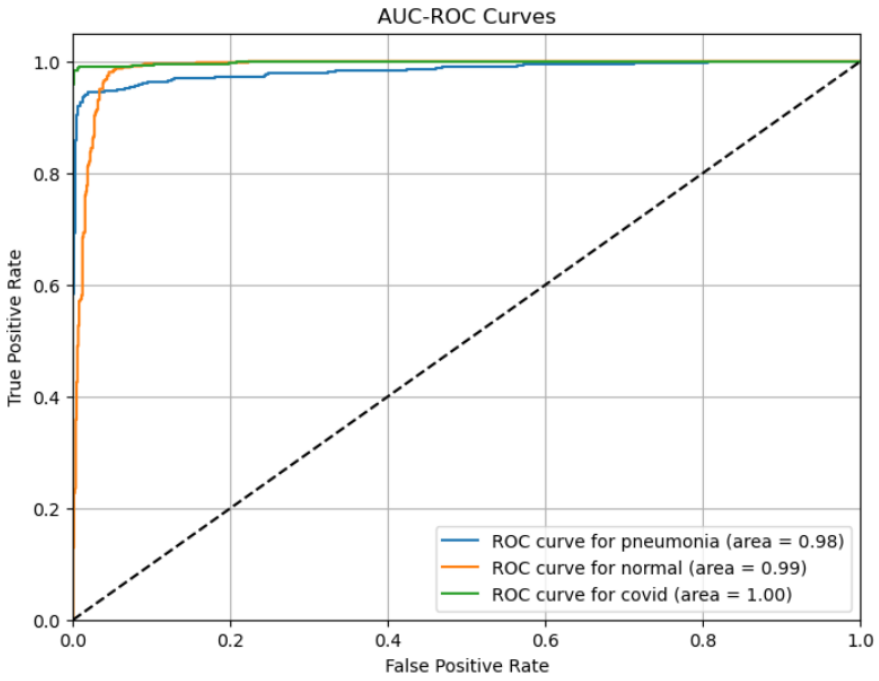



Figure 5 AUC-ROCCurves For Proposed Model

The AUC-ROC curve shows how well the model can separate the three classes: Pneumonia,
Normal, and COVID-19. It compares the True Positive Rate with the False Positive Rate at different thresholds.
The curves are all close to the top-left corner, which means the model is doing a great job in classifying the images correctly. Here's a quick summary:

- For Pneumonia, the AUC score is 0.98, showing that the model is very good at identifying pneumonia cases, though not perfect.
- For Normal cases, the AUC is 0.99, which is even better, with very few wrong predictions.
- For COVID-19, the model scored a perfect 1.00, meaning it classified all

COVID cases correctly without any errors.

The high AUC values across all categories underscore the robustness and effectiveness of the model. The almost vertical and horizontal trajectories of the ROC curves, especially for the COVID category, signify a low false positive rate and a high true positive rate, critical for medical diagnosis tasks where accuracy is paramount.
The dotted diagonal line in Figure 4 represents random chance (AUC = 0.5), and the significant deviation of the ROC curves from this line further highlights the model's superiority over a random classifier.

## 7.5. GRAD-CAM VISUALIZATION FOR INTEPRETABILITY

To better understand how the model makes its predictions, Grad-CAM (Gradient-weighted Class Activation Mapping) was used. Grad-CAM helps us visualize the areas in the chest X-ray images that the model pays attention to when making a decision. It creates a heatmap that highlights the important regions influencing the output.
The heatmaps generated by Grad-CAM clearly show that the model is focusing on the chest regions that are medically significant — such as areas showing opacities, inflammation, or
other abnormalities. These regions are commonly linked to conditions like pneumonia or COVID-19, which indicates that the model is learning useful and relevant features.
Although there were minor warnings due to changes in software libraries during the Grad-CAM implementation, these did not affect the results. The visual outputs were accurate and showed that the model is not simply memorizing the training data or focusing on background noise. Instead, it is recognizing actual patterns that match clinical signs of disease.
This visual interpretability adds another layer of trust, especially for use in the medical field. It helps doctors understand why the model is predicting a certain disease and ensures that the decision is based on the correct parts of the image.
In combination with other evaluation results:

- The model has shown high accuracy, precision, and recall, particularly for detecting COVID-19, where it achieved a perfect AUC score.
- The confusion matrix shows that most predictions are correct, with only a few errors — mainly between pneumonia and normal cases, which might improve

with more training data or better tuning.

- The ROC-AUC curves further confirm the model's strong performance, with excellent sensitivity (true positive rate) and specificity (true negative rate).

Overall, the Grad-CAM visualizations prove that the model is learning meaningful medical patterns. This makes it not only accurate but also interpretable and trustworthy, which is essential when using AI tools in healthcare. It has strong potential to assist healthcare professionals in diagnosing pulmonary diseases from chest X-rays in a reliable and explainable way.

## 7. CONCLUSION

This study presents a well-structured deep learning system designed to classify chest X-ray images into three categories: Pneumonia, COVID-19, and Normal. The model was built using a step-by-step process, starting with image enhancement using CLAHE, followed by feature extraction using a powerful deep learning model called EfficientNetV2S, and ending with a custom classification network. To improve transparency and build trust in the model's decisions, Grad-CAM visualizations were included to show which parts of the X-ray the model focused on while making predictions.

A new dataset named the CONOP dataset was created for this purpose by collecting chest X-ray images from various trusted sources. Special care was taken to make sure all classes (COVID-19, Pneumonia, Normal) had a balanced number of images, and the quality of the images was consistent. This helped the model to learn patterns that are common across many patients, rather than just memorizing specific examples.

The model performed very well, with a validation accuracy of 95.68% and a high AUC score of 0.9884, showing that it can correctly identify both positive and negative cases. It also achieved strong precision, recall, and F1-scores for each class, proving that the model is reliable in recognizing different types of pulmonary conditions. This means it is both sensitive (good at detecting disease) and specific (good at identifying healthy cases).

The Grad-CAM heatmaps added an important layer of interpretability by visually showing which regions of the lungs were most important for the model's decisions. This makes the system more understandable for doctors, and gives them confidence in using AI as a supportive tool for diagnosis.

Several performance metrics such as accuracy, precision, recall, F1-score, and the ROC-AUC curve were used to evaluate the model. The consistent results across all classes suggest that the model has learned to make fair and balanced predictions, which is crucial in real-world medical settings to avoid both missed diagnoses and false alarms.

In summary, the proposed framework offers a strong foundation for automated, accurate, and explainable diagnosis of pulmonary diseases using chest X-ray images. With further testing on larger and more diverse datasets, this model could be a valuable addition to clinical decision-making systems, helping doctors diagnose lung conditions

more efficiently and improving outcomes for patients.

## 8. CHALLENGES AND FUTURE WORKS

During this project, several challenges came up that provided valuable learning experiences related to both deep learning techniques and medical image analysis. One major challenge was collecting a balanced and diverse dataset. Since no single source provided a complete set of COVID-19 chest X-rays, we had to gather images from different public databases. These images varied in quality, resolution, and format, which made standardizing and preprocessing an essential task. Another layer of complexity was making sure only posteroanterior (PA) view X-rays were used, as incorrect views could mislead the model. Many images had to be manually checked and filtered.

Another significant issue was class imbalance. The number of COVID-19 images was far less compared to Normal and Pneumonia images. To deal with this, data augmentation techniques were applied to generate more COVID-19 samples. However, we had to be careful that the augmented images still looked realistic and reflected actual clinical conditions.

Choosing the best deep learning model was another challenge. After testing different models, we selected EfficientNetV2S for its good accuracy and efficient performance. Even then, tuning hyperparameters such as learning rate, dropout, and batch size took multiple experiments. Overfitting was a concern due to the limited dataset size. To handle this, we used methods like dropout layers, early stopping, and batch normalization to improve generalization.

Another important challenge was making the model's predictions interpretable. In clinical settings, it is not enough for an AI model to just give a result; it must also show why it made that prediction. To address this, we used Grad-CAM to create heatmaps that showed which areas of the X-ray the model focused on. However, understanding and validating these highlighted areas required some medical background and careful observation.

We also faced difficulties in ensuring the model would work well with new data from different hospitals or imaging devices. While the model performed well on our curated dataset, in real-world scenarios, X-ray images may come from various sources with different standards, patient profiles, and devices. Building a system that can generalize across such differences remains a challenge. Additionally, working with high-resolution medical images needed powerful computing resources. Limited access to high-end

GPUs sometimes slowed down training and experimentation, especially while applying advanced image augmentation.

Despite these challenges, our system was able to classify chest X-rays into three categories—COVID-19, Pneumonia, and Normal—with good accuracy. However, there is room for future improvement and exploration:

1. Better Preprocessing Methods: While CLAHE worked well for contrast enhancement, other preprocessing techniques like Gaussian filtering, median filtering, or different forms of histogram equalization could be explored to better highlight disease areas. Adding image normalization or denoising algorithms may also help improve clarity and model performance.
2. Trying Different Models: Although EfficientNetV2S gave good results, we could try other models like ResNet50, DenseNet121, ConvNeXt, or transformer-based models like Vision Transformers (ViT). A hybrid or ensemble model could also be created by combining the strengths of multiple models to improve accuracy.
3. Improving the Dataset: Expanding the CONOP dataset by including chest X-rays from various hospitals or countries could help the model generalize better. Including extra clinical data such as patient age, symptoms, and medical history could make the predictions more informative and personalized.
4. Explainability Tools: Grad-CAM is useful, but we can also try other methods such as LIME (Local Interpretable Model-Agnostic Explanations), SHAP (SHapley Additive Explanations), or attention-based visualizations to provide deeper insights into how the model is making decisions.
5. Real-World Application: The model can be turned into a web or mobile app that allows real-time X-ray classification. Testing the app in a clinical setting and getting feedback from doctors could help refine the tool. Integration with hospital systems and close collaboration with healthcare.

In conclusion, while our model has shown promising results, future efforts should aim to improve robustness, transparency, and real-world usability. With more advanced techniques, better datasets, and stronger interpretability tools, AI-based diagnostic systems can play a major role in supporting medical decision-making.

## 9. CODE IMPLEMENTATION

```
import os
import numpy as np
import cv2
import random
import tensorflow as tf
from tensorflow.keras import layers, models, callbacks, regularizers
from tensorflow.keras.applications import EfficientNetV2S
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.cm as cm
```

```
# CLAHE Preprocessing
# -------------------------
def apply_clahe(img):
    lab = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=2.0)
    cl = clahe.apply(l)
    merged = cv2.merge((cl, a, b))
    return cv2.cvtColor(merged, cv2.COLOR_LAB2BGR)
```

```
def build_model():
    base_model = EfficientNetV2S(include_top=False, input_shape=(IMG_SIZE, IMG_SIZE, 3), weights="imagenet")
    base_model.trainable = True
    model = models.Sequential([
        base_model,
        layers.GlobalAveragePooling2D(),
        layers.BatchNormalization(),
        layers.Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.001)),
        layers.BatchNormalization(),
        layers.Dropout(0.3),
        layers.Dense(3, activation='softmax')
    ])
    return model
```

```
model = build_model()
from tensorflow.keras.metrics import AUC

model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=[
        'accuracy',
        AUC(name='auc', multi_label=False)
    ]
)
```

```
history = model.fit(X_train, y_train,
                    validation_data=(X_val, y_val),
                    epochs=EPOCHS,
                    batch_size=BATCH_SIZE,
                    callbacks=[early_stop, reduce_lr])
```

```
Epoch 1/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1188s 6s/step - accuracy: 0.7666 - auc: 0.9003 - loss: 0.8909 - val_accuracy: 0.9194 - val_auc: 0.9827 - val_loss: 0.5169 - learning_rate: 1.0000e-04
Epoch 2/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1086s 6s/step - accuracy: 0.9327 - auc: 0.9875 - loss: 0.4299 - val_accuracy: 0.9374 - val_auc: 0.9891 - val_loss: 0.4158 - learning_rate: 1.0000e-04
Epoch 3/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1041s 6s/step - accuracy: 0.9498 - auc: 0.9933 - loss: 0.3739 - val_accuracy: 0.8992 - val_auc: 0.9747 - val_loss: 0.5437 - learning_rate: 1.0000e-04
Epoch 4/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1047s 6s/step - accuracy: 0.9704 - auc: 0.9970 - loss: 0.3181 - val_accuracy: 0.9518 - val_auc: 0.9871 - val_loss: 0.4226 - learning_rate: 1.0000e-04
Epoch 5/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9660 - auc: 0.9970 - loss: 0.3135
Epoch 5: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-05.
174/174 ━━━━━━━━━━━━━━━━━━━━ 1037s 6s/step - accuracy: 0.9661 - auc: 0.9970 - loss: 0.3135 - val_accuracy: 0.9482 - val_auc: 0.9875 - val_loss: 0.4157 - learning_rate: 1.0000e-04
Epoch 6/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1051s 6s/step - accuracy: 0.9833 - auc: 0.9982 - loss: 0.2743 - val_accuracy: 0.9561 - val_auc: 0.9866 - val_loss: 0.4165 - learning_rate: 5.0000e-05
Epoch 7/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 992s 6s/step - accuracy: 0.9870 - auc: 0.9991 - loss: 0.2587 - val_accuracy: 0.9597 - val_auc: 0.9860 - val_loss: 0.4152 - learning_rate: 5.0000e-05
Epoch 8/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 992s 6s/step - accuracy: 0.9970 - auc: 0.9999 - loss: 0.2316 - val_accuracy: 0.9568 - val_auc: 0.9884 - val_loss: 0.3969 - learning_rate: 5.0000e-05
Epoch 9/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1021s 6s/step - accuracy: 0.9919 - auc: 0.9999 - loss: 0.2325 - val_accuracy: 0.9618 - val_auc: 0.9847 - val_loss: 0.4329 - learning_rate: 5.0000e-05
Epoch 10/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1052s 6s/step - accuracy: 0.9947 - auc: 0.9999 - loss: 0.2262 - val_accuracy: 0.9597 - val_auc: 0.9862 - val_loss: 0.4341 - learning_rate: 5.0000e-05
Epoch 11/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9938 - auc: 0.9998 - loss: 0.2265
Epoch 11: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
174/174 ━━━━━━━━━━━━━━━━━━━━ 1048s 6s/step - accuracy: 0.9938 - auc: 0.9998 - loss: 0.2265 - val_accuracy: 0.9561 - val_auc: 0.9832 - val_loss: 0.4609 - learning_rate: 5.0000e-05
Epoch 12/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1030s 6s/step - accuracy: 0.9949 - auc: 0.9999 - loss: 0.2191 - val_accuracy: 0.9568 - val_auc: 0.9845 - val_loss: 0.4380 - learning_rate: 2.5000e-05
Epoch 13/20
174/174 ━━━━━━━━━━━━━━━━━━━━ 1063s 6s/step - accuracy: 0.9956 - auc: 0.9999 - loss: 0.2165 - val_accuracy: 0.9568 - val_auc: 0.9831 - val_loss: 0.4430 - learning_rate: 2.5000e-05
```

```python
def preprocess_external_image(img_path):
    img = cv2.imread(img_path)
    if img is None:
        raise ValueError("Image not found or not readable.")
    img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
    img = apply_clahe(img)
    img = img / 255.0
    return np.expand_dims(img, axis=0)  # Add batch dimension

# Path to your external image
external_img_path = r"C:\Users\anuvv\FINAL PROJECT\CONOP\pneumonia\00000165_001.png"

# Preprocess and predict
try:
    processed_img = preprocess_external_image(external_img_path)
    prediction = model.predict(processed_img)
    predicted_idx = np.argmax(prediction)
    predicted_label = CLASS_NAMES[predicted_idx]

    print(f"\nPredicted Disease: {predicted_label}")
    print("Confidence Scores:")
    for i, cls_name in enumerate(CLASS_NAMES):
        print(f"{cls_name}: {prediction[0][i]:.4f}")
except Exception as e:
    print("Error:", e)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 273ms/step

Predicted Disease: pneumonia
Confidence Scores:
pneumonia: 0.9976
normal: 0.0023
covid: 0.0001
```

```
y_pred = model.predict(X_val)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_val, axis=1)

print("\nClassification Report:\n")
print(classification_report(y_true, y_pred_classes, target_names=CLASS_NAMES))
```

```
44/44 ━━━━━━━━━━━━━━━━━━━━ 62s 1s/step

Classification Report:

              precision    recall  f1-score   support

   pneumonia       0.96      0.94      0.95       463
      normal       0.93      0.95      0.94       463
       covid       0.99      0.98      0.98       463

    accuracy                           0.96      1389
   macro avg       0.96      0.96      0.96      1389
weighted avg       0.96      0.96      0.96      1389
```

```
def make_gradcam_heatmap(img_array, model, last_conv_layer_name, pred_index=None):
    grad_model = tf.keras.models.Model(
        [model.inputs], [model.get_layer(last_conv_layer_name).output, model.output]
    )
    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(img_array)
        if pred_index is None:
            pred_index = tf.argmax(predictions[0])
        class_channel = predictions[:, pred_index]
    grads = tape.gradient(class_channel, conv_outputs)
    pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))
    conv_outputs = conv_outputs[0]
    heatmap = conv_outputs @ pooled_grads[..., tf.newaxis]
    heatmap = tf.squeeze(heatmap)
    heatmap = tf.maximum(heatmap, 0) / tf.math.reduce_max(heatmap)
    return heatmap.numpy()
```

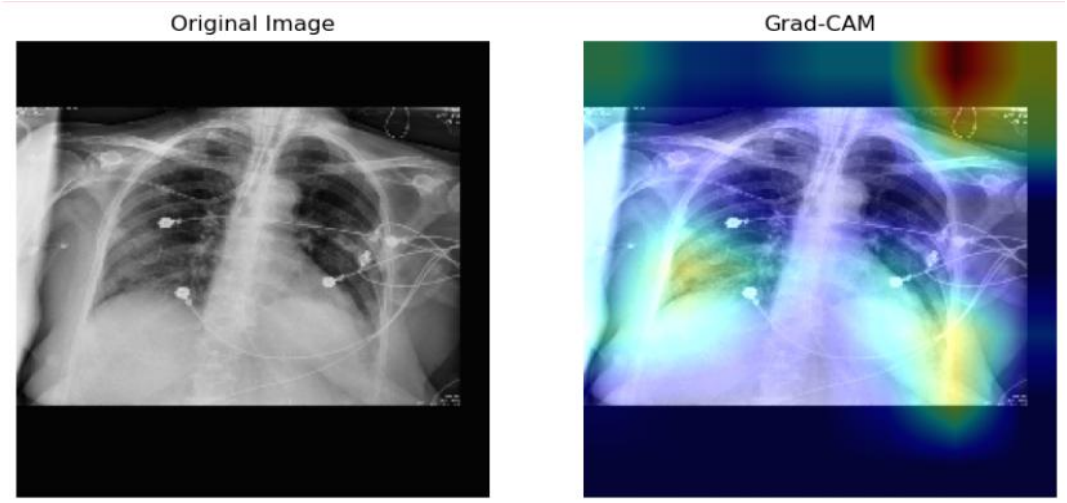
```
# Pick a random validation image to visualize
idx = random.randint(0, len(X_val) - 1)
img = X_val[idx:idx+1]

heatmap = make_gradcam_heatmap(img, model, last_conv_layer_name="block6e_add")
```

```
# Overlay
img_original = (img[0] * 255).astype("uint8")
heatmap_resized = cv2.resize(heatmap, (IMG_SIZE, IMG_SIZE))
heatmap_rescaled = np.uint8(255 * heatmap_resized)
# colored_heatmap = cm.get_cmap("jet")(heatmap_rescaled / 255.0)
colored_heatmap = mpl_cm.get_cmap("jet")(heatmap_rescaled / 255.0)
colored_heatmap = np.delete(colored_heatmap, 3, 2)  # Remove alpha
superimposed_img = colored_heatmap * 0.4 + img_original / 255.0

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(img_original)
plt.title("Original Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(superimposed_img)
plt.title("Grad-CAM")
plt.axis('off')
plt.show()
```



## 10. REFERENCES


[1] World Health Organization. (2020). Global surveillance for human infection with coronavirus disease (COVID-19). WHO.

[2] Rajpurkar, P., et al. (2017). CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv:1711.05225.

[3] Litjens, G., et al. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88.

[4] Irvin, J., et al. (2019). CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. AAAI.

[5] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. arXiv:2104.00298

[6] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. Graphics Gems IV.

[7] Selvaraju, R. R., et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. ICCV.

[8] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

[9] Rajpurkar, P., Irvin, J., Zhu, K., et al. (2017). CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv:1711.05225.

[10] Selvaraju, R. R., et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. ICCV.

[11] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. In Graphics Gems IV.

[12] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. arXiv:2104.00298.

[13] Ardakani, A. A., et al. (2020). Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. Computers in Biology and Medicine, 121, 103795.

[14] Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In Graphics gems IV (pp. 474-485). Academic Press Professional, Inc.

[15] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. Proceedings of the 38th International Conference on Machine Learning (ICML).

[16] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR).

[17] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[18] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. In P. S. Heckbert (Ed.), Graphics Gems IV. Academic Press.

[19] Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Pearson.

[20] Jain, A. K. (1989). Fundamentals of Digital Image Processing. Prentice-Hall.

[21] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning (ICML).

[22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. https://www.deeplearningbook.org

[23] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. arXiv preprint arXiv:2104.00298.

[24] Lin, M., Chen, Q., & Yan, S. (2014). Network in network. arXiv preprint arXiv:1312.4400.

[25] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

[26] 26. Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. Proceedings of the twenty-first international conference on Machine learning (ICML).

[27] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[29] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[30] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[31] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[32] Pizer, S. M., Amburn, E. P., Austin, J. D., et al. (1987). Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing, 39(3), 355–368.

[33] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[34] Suresh, H., & Guttag, J. V. (2019). A Framework for Understanding Unintended Consequences of Machine Learning. Communications of the ACM, 63(5), 62–71.

[35] Tjoa, E., & Guan, C. (2020). A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. IEEE Transactions on Neural Networks and Learning Systems.

[36] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, 618–626.

[37] Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 839–847.

[38] Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(4), e1312.

[39] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.

[40] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2921–2929.

[41] [41] Cohen, J.P., Morrison, P., & Dao, L. (2020). COVID-19 image data collection. arXiv:2003.11597