Frontiers in Health Informatics www.healthinformaticsjournal.com
ISSN-Online: 2676-7104
2024; Vol 13: Issue 2 Open Access

Deep Learning-Based Framework for Identifying COVID-19 Pneumonia in Chest X-Ray Imaging

[1]Bharti Sahu, [2]Dr. Bhagwan Phulpagar, [3]Dr. Pramod Patil

[1]Dr. D.Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, India. sahubharti1430@gmail.com

[2]P.E.S's Modern College of Engineering, Pune, Maharashtra, India. phulpagarbd@gmail.com

[3]Dean R&D, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, India. pramod.patil@dypvp.edu.in

Article Info

Article type:

Research

Article History:

Received: 2024-04-01

Revised: 2024-05-22

Accepted: 2024-06-22

Keywords:

Pneumonia Detection, Chest X-ray Imaging, Deep Learning Framework, Convolutional Neural Networks (CNN), Medical Image Classification

ABSTRACT

The COVID-19 widespread has given worldwide wellbeing care frameworks issues they have never seen some time recently, so they require speedy and precise determination devices. Chest X-rays (CXRs) have gotten to be a vital way to analyse COVID-19-related pneumonia, particularly in places with few restorative assets. This paper proposes a profound learning-based methodology for finding COVID-19 pneumonia in CT filters. The objective is to make strides the exactness of analyse and make doctors' employments simpler. A convolutional neural network (CNN) plan is utilized within the system. The CNN was prepared on big sets of pictures and after that fine-tuned employing a carefully chosen set of CXR pictures labelled with COVID-19, viral pneumonia, bacterial pneumonia, and solid cases. Numerous convolutional layers are utilized to gather highlights, and after that completely connected layers are utilized for classification. Information improvement strategies like revolution, scaling, and level moving are utilized to settle lesson bungle and make demonstrate steadier. We tried the recommended system on a partitioned set of information and found that it was exceptionally great at telling the contrast between COVID-19 pneumonia and other sorts of pneumonia and ordinary lung conditions. A comparison with other profound learning models and standard machine learning strategies appears that the proposed approach works superior. Usually done with Grad-CAM design that make the pictures less demanding to get it by indicating out imperative parts of the CXR pictures that offer assistance the show make choices. This makes it simpler to accept the model's forecasts and makes a difference specialists make choices. The study comes to the conclusion that the proposed deep learning-based methodology could be a great way to rapidly and accurately identify COVID-19 pneumonia in CXR pictures. This seem lead to way better patient results and way better use of healthcare assets.

1. INTRODUCTION

The COVID-19 pandemic quickly turned into a global health disaster that is putting a lot of stress on healthcare systems around the world and showing how important it is to have accurate diagnosis tools. SARS-CoV-2 is a new coronavirus that mostly affects the respiratory system. One of the worst effects of COVID-19 is asthma. Accurate and timely detection is important for managing patients well, lowering the spread rate, and making the best use of healthcare resources. Because it is easy to get and doesn't cost much, chest X-rays (CXRs) have become an important part of detecting pneumonia, especially in places with few resources where more advanced imaging methods like computed tomography (CT) scans might not be available. The old way of reading CXR pictures rests on the knowledge of radiologists, who look for patterns that are typical of COVID-19 pneumonia on the images [1]. However, this is getting harder to do because of the large number of cases during the pandemic and the small differences between COVID-19 pneumonia and other lung diseases. Misunderstandings or delays in evaluation can cause patients to get the wrong care, which can make their conditions worse. Because of this, we urgently need automatic tools that can help doctors by analyzing CXR pictures quickly and correctly. A branch of AI called "deep learning" has shown a lot of promise in the area of medical imaging [2]. In particular, Convolutional Neural Networks (CNNs) have changed the way pictures are studied by making it possible for complex patterns to be found automatically and accurately in medical photos. CNNs have been used successfully in many medical imaging jobs, such as finding tumors, separating organs into their own parts, and putting different diseases into groups. As CNNs have been successful in these areas, they could be used to automatically find COVID-19 pneumonia in CXR pictures.

In this consider, we recommend a profound learning-based framework that can be utilized to discover COVID-19 pneumonia in CXR pictures. To form the framework work, CNNs' solid include extraction aptitudes are fine-tuned on a carefully chosen set of CXR pictures. These pictures appear individuals with COVID-19 pneumonia, as well as individuals with other sorts of pneumonia, like bacterial and viral pneumonia, and individuals with great lung conditions. The CNN can learn to tell the distinction between COVID-19 pneumonia and other comparative conditions by being prepared on this large set of information. Typically an vital ability to have since distinctive sorts of pneumonia can have comparable indications and picture results [3]. The dataset's lesson bungle is one of the greatest issues that has to be illuminated when making this kind of model. Different datasets may have as well numerous or as well few COVID-19 cases, which seem cause the show to form one-sided gauges. A few information upgrade strategies are utilized to assist with this issue. Haphazardly turning, scaling, and shifting pictures on a level plane are a few of these strategies that offer assistance make a more adjusted and steady collection. In expansion, expansion makes the demonstrate more generalizable, which implies it works well on information it hasn't seen some time recently, which is critical for utilizing it in real-life clinical settings. Another imperative thing around the proposed structure is that it is simple to get it. It's not sufficient for an AI demonstrate to be dependable in therapeutic settings, particularly when making life-or-death choices; it moreover should be simple to get it. For clinicians to accept a model's forecasts, they ought to know why it made that choice. Grad-CAM (Gradient-weighted Course Actuation Mapping) pictures are built into the system to bargain with this [4]. A strategy called Grad-CAM draws consideration to the parts of a picture that are most critical to the model's choice. This gives specialists a visual description that they can get it. This include makes the show more clear and makes a difference specialists make better clinical choices by letting them check that the model is centered on the proper parts of the CXR pictures. What makes this think about critical is that it might offer assistance healthcare frameworks by making an automatic, reliable, and easy-to-understand instrument for recognizing COVID-19 pneumonia. As the widespread gets worse, these sorts of instruments may well be exceptionally accommodating for taking care of persistent care, particularly in places where it's difficult to get master specialists.

2. RELATED WORK

The utilize of chest X-ray (CXR) pictures to recognize COVID-19 pneumonia has gotten a parcel of consideration from analysts, particularly since the widespread has spread so rapidly. Within the early stages of the spread, specialists depended on clinical signs and switch transcription-polymerase chain reaction (RT-PCR) tests to create analyze. In any case, RT-PCR's blemishes, such as its lower affectability and longer working times, made it clear that we required other testing instruments to assist. As a result, imaging procedures, particularly computed tomography (CT) and x-rays, got to be exceptionally vital for finding and keeping an eye on lung issues connected to COVID-19 as before long as conceivable [5]. Some time recently COVID-19, there was a parcel of investigate into how profound learning models can be utilized in therapeutic imaging, particularly to discover pneumonia. Convolutional Neural Systems (CNNs) are great at finding diverse sorts of pneumonia in CT filters, like bacterial and viral sicknesses. These models were prepared on huge sets of information, just like the ChestX-ray14 dataset, which has more than 100,000 frontal-view CXR pictures. This made it conceivable to form strong models for finding pneumonia. Analysts begun to alter these more seasoned models to particularly discover COVID-19 pneumonia when it to begin with showed up since it contains a part in common with other sorts of pneumonia [6].

CNNs have been utilized in a number of thinks about to tell the contrast between COVID-19 pneumonia and other sorts of pneumonia in CXR pictures. As a case, Wang et al. made the COVID-Net, a profound learning demonstrate that can recognize COVID-19 cases from CXR pictures. COVID-Net was prepared on a dataset that was put together fair for this reason. It had pictures from a part of distinctive sources to create beyond any doubt there was sufficient assortment. The show did an incredible work of telling the distinction between COVID-19 pneumonia and other conditions. This appears that profound learning can be utilized for fast and exact location [7]. Apostolopoulos and Mpesiana took a distinctive but curiously strategy when they looked into utilizing exchange learning to sort CXR pictures into bunches for COVID-19, common pneumonia, and sound individuals. By tweaking a CNN demonstrate that had as of now been trained, they got a parcel of exactness, appearing how valuable exchange learning can be when there isn't a parcel of COVID-19-specific information. This method used CNNs' ability to be generalized, which let the model change to the specifics of COVID-19 pneumonia with only a few tagged pictures. Some studies have also looked at how to divide up the parts of the lungs that are damaged by COVID-19 pneumonia in addition to sorting tasks. Segmentation gives more thorough information about the size and location of lung problems, which is very important for figuring out how bad the disease is. Zhang et al. suggested a way to separate COVID-19-affected areas in CXR images that uses a U-Net design and focus processes. This model showed potential in correctly defining the areas of interest, which could help doctors keep an eye on how the disease is getting worse and how well treatments are working [8]. Data access has been a problem for a long time when trying to make deep learning models for finding COVID-19. At first, the virus spread so quickly that it took longer than expected to collect and name the big, varied datasets that were needed to train strong models. In answer, researchers have used methods called "data augmentation" to make collections bigger and more varied than they really are. To make models more flexible, methods like rotation, scaling, and horizontal shifting have been used. Some studies have also looked into using Generative Adversarial Networks (GANs) to create fake data to add to the limited COVID-19 CXR datasets and make the training process better [9].

The readability of deep learning models is another important thing to think about in linked work. When AI models are used in medicine, they need to not only make correct predictions, but also give doctors reasons they can trust and understand. A number of studies have used Grad-CAM (Gradient-weighted Class Activation Mapping) to see which parts of the CXR picture were most important to the model's choice [10]. This method has been very helpful in proving that the model's focus on clinically important traits is correct, which makes it more reliable in a clinical setting. Even though the results look good, there are some problems with using deep learning models in clinical practice. One of the main worries is that picture quality and collection methods can be different in different healthcare settings, which can have an impact on how well the model works. To deal with this, some studies have stressed how important it is to make models that can handle these kinds of differences. One way to do this is to train them on different datasets or use techniques for domain adaptation.

Table 1: Summary of related work

Method	Approach	Key Finding	Application	Limitation
CNN [11]	Direct classification of CXR images using CNN architecture	High accuracy in detecting pneumonia	Medical image analysis for COVID-19 detection	Potential overfitting with limited data
Transfer Learning [12]	Fine-tuning pre-trained models (e.g., ResNet) on COVID-19 CXR data	Improved accuracy with less training data	Efficient in resource-constrained environments	Dependency on large-scale pre-trained models
EfficientNet [13]	Compound scaling of network dimensions	Achieves high accuracy with fewer parameters	High-performance diagnosis in clinical settings	Requires careful tuning of scaling factors
Lightweight MobileNet [14]	Depthwise separable convolutions for reduced complexity	Maintains accuracy with reduced computational resources	Mobile and embedded device applications	Limited by lower capacity for feature extraction
SqueezeNet	Smaller network with	Comparable accuracy	Suitable for	May sacrifice

[15]	fewer parameters	to larger models	deployment in low-resource settings	accuracy for model size
DenseNet [16]	Dense connections to improve feature propagation	Enhances feature reuse and gradient flow	Accurate pneumonia detection in complex cases	High memory consumption and computational cost
VGG16 [17]	Deep CNN with multiple layers	Effective feature extraction for pneumonia detection	Useful in detailed image analysis for medical diagnosis	High computational demands
ResNet [18]	Residual learning framework to avoid vanishing gradients	Better training efficiency and accuracy	High accuracy in pneumonia classification	Increased complexity and computational requirements
Xception [19]	Depthwise separable convolutions with extreme inception modules	High performance with reduced parameters	Efficient and accurate COVID-19 detection in CXRs	Complex architecture that requires significant tuning
InceptionV3 [20]	Inception modules with mixed convolutions	Balances accuracy and efficiency	Versatile model for various medical image classifications	May require extensive tuning for specific tasks
U-Net [21]	Convolutional network for image segmentation	Effective segmentation of lung regions affected by pneumonia	Detailed analysis of affected lung areas	Requires large amounts of annotated data
Grad-CAM [22]	Visual explanations for CNN decisions	Improves interpretability of model predictions	Enhancing trust in AI-driven diagnostics	Limited by the accuracy of the underlying model
Hybrid CNN-RNN [23]	Combination of CNN for feature extraction and RNN for sequence modeling	Captures temporal dependencies in sequential data	Analyzing sequential chest X-ray scans	Increased complexity and computational load

3. DATASET DESCRIPTION

The Chest X-Ray Pictures dataset, accessible on Kaggle, may be a broadly utilized asset for creating and testing machine learning models in therapeutic imaging. This dataset contains 5,863 pictures categorized into three classes: ordinary, bacterial pneumonia, and viral pneumonia, with the last mentioned counting pictures pertinent for COVID-19 investigate. The pictures are organized into preparing, approval, and test sets, encouraging show improvement and execution assessment. This dataset [24] is especially profitable for its differences and estimate, permitting for strong preparing of profound learning models, particularly Convolutional Neural Systems (CNNs). The pictures are of changing quality and taken from diverse sources, which presents a level of changeability that models must handle, improving their generalizability, sample dataset images shown in figure 1. Furthermore, the dataset's open accessibility has made it a benchmark for various thinks about, driving progressions in pneumonia location and classification utilizing CXR imaging.

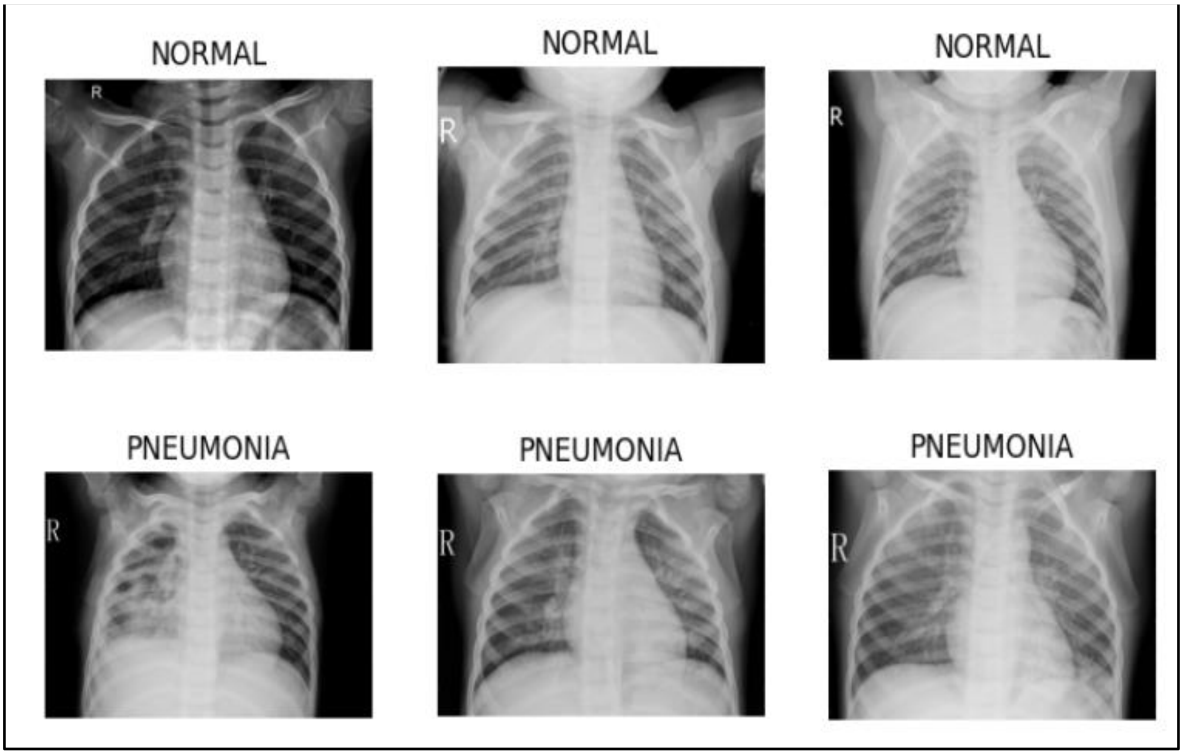

Figure 1. Dataset Sample Images

4. METHODOLOGY

There are many imperative steps within the handle of finding pneumonia on chest X-rays. The primary is the crude data. The Chest X-Ray Pictures collection from Kaggle is the premise for this consider. It has pictures that are labelled as ordinary, bacterial pneumonia, and viral pneumonia, systematic workflow illustrate in figure 3. To begin with, the dataset is looked at to see how equitably it falls into these bunches. This makes beyond any doubt that the preparing of the show is reasonable. Information planning is an vital step that begins with contracting the pictures to a standard estimate that can be utilized by profound learning models. Moreover, resampling is done to settle any lesson bungle, which is essential to keep show estimates from being biased. Once the information has been cleaned up, it is included to the show to form it more stable. These strategies incorporate irregular revolutions, rescaling, width and tallness changes, shearing, zooming, and level flips. They are all implied to form the preparing set more different and move forward generalization.

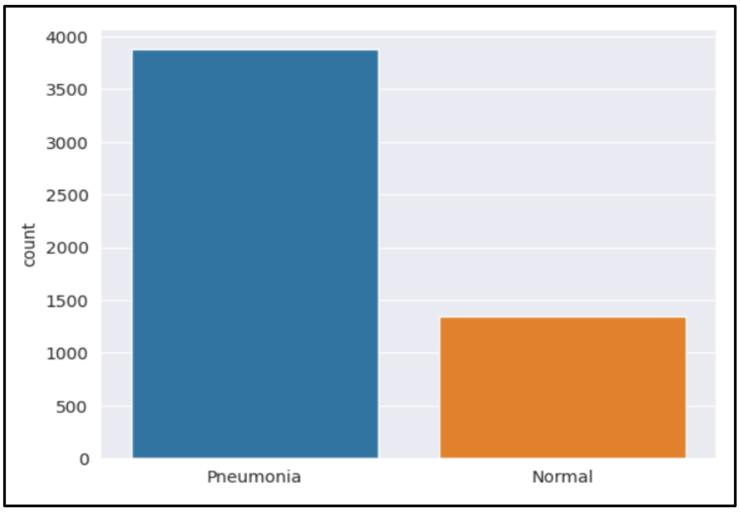

Figure 2. Dataset Distribution

After that, the dataset is part into preparing and testing sets with an 80-20 part, distribution of dataset shown in figure 2. This lets a full audit of show execution happen. A number of strategies are utilized for the classification job, such as CNNs, EfficientNet, Lightweight MobileNet, and Lightweight EfficientNet. Since these models have been appeared to be great at classifying pictures, particularly therapeutic pictures, they were chosen. The objective is to urge exceptionally great comes about when utilizing information planning, information improvement, and progressed classification strategies to discover pneumonia on chest X-rays.

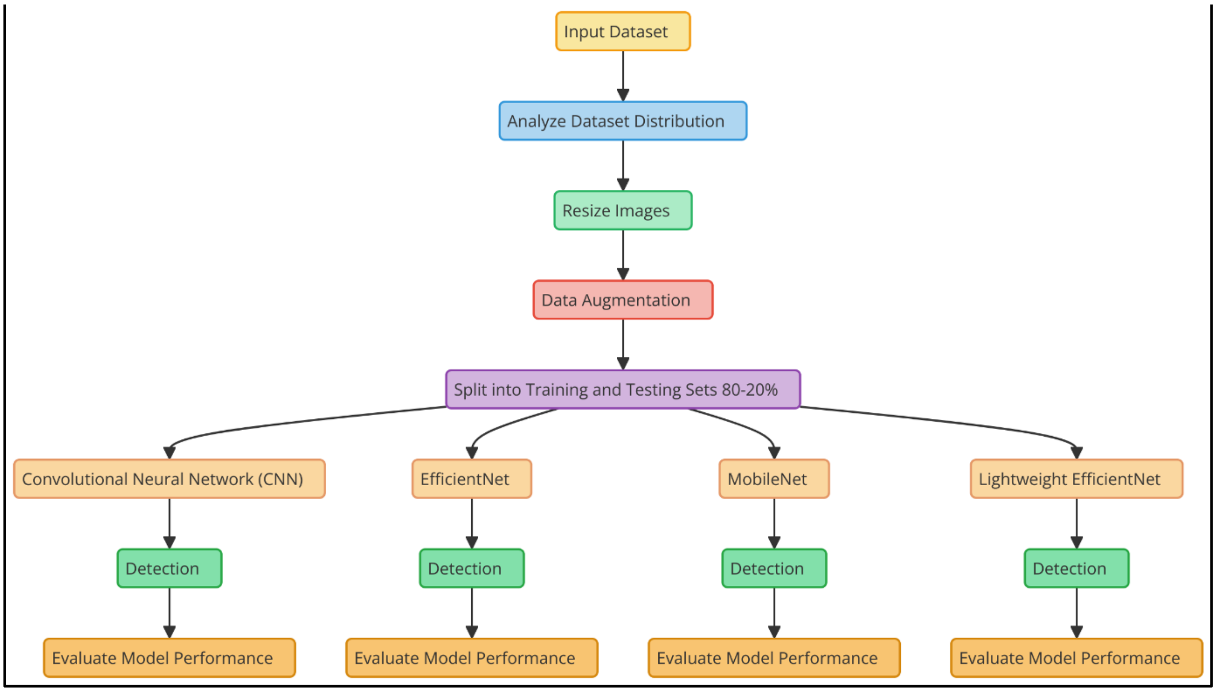

Figure 3: Overview flowchart of proposed system Architecture

A. Data Pre-processing

a. Resize Image: One of the foremost critical steps within the planning of profound learning forms, particularly when utilizing convolutional neural networks (CNNs), is resizing the pictures. A few of the initial pictures within the collection may be diverse sizes, which can make preparing the demonstrate more troublesome. We make beyond any doubt that all pictures can be taken care of the same way by contracting them all to the same measure, which is more often than not 224x224 pixels based on what the demonstrate needs as input. The information is made more uniform in this step, and it moreover makes preparing less demanding on the computer. When resizing, it's vital to keep the viewpoint proportion and not alter imperative highlights that are required for a adjust determination. Usually particularly vital in restorative imaging where detail is exceptionally critical, the figure 4 illustrate the data after preprocessing.

1. Aspect Ratio Preservation:

New Height = OriginalHeightOriginal Width* New Width

This equation ensures that the aspect ratio is preserved when resizing the image. If the new width is specified, the new height is calculated to maintain the original aspect ratio.

2. Bilinear Interpolation for Resizing:

I'x, y= Σ Σ Ixi, yj* 1 - x - xi* 1 - y - yj

This equation represents bilinear interpolation, where I'(x, y) is the interpolated pixel value at the new coordinates (x, y), and I(x_i, y_j) are the four nearest pixels in the original image.

3. Downsampling with Gaussian Smoothing:

I'x, y= 1ΣΣGm, n* Σ Σ Ix + m, y + n* Gm, n

Here, I'(x, y) is the resized pixel value, G(m, n) is the Gaussian kernel, and k defines the kernel size. This equation is used to downsample an image with Gaussian smoothing to reduce aliasing.

b. Resample: Resampling fixes the common issue of lesson awkwardness in datasets, which happens a part in therapeutic pictures where a few conditions might not be appeared sufficient. Within the Chest X-Ray Pictures dataset, resampling implies changing how the pictures are spread out among the distinctive classes (like typical, bacterial pneumonia, and viral pneumonia) to form the dataset more reasonable. This could be done by either not choosing sufficient of the larger part lesson or as well numerous of the minority lesson. Resampling that works well keeps the demonstrate from getting to be one-sided toward the classes that happen more often, so it can learn to discover highlights from all classes more accurately. This is often an imperative step for making the demonstrate more generalizable and making beyond any doubt it works the same way in all bunches.

Oversampling (Synthetic Minority Over-sampling Technique - SMOTE):

$$x\_new = x\_i + \lambda * (x\_j - x\_i), \text{ where } \lambda \sim \text{Uniform}(0, 1)$$

In this equation, x_new is the new synthetic sample generated by linearly interpolating between a minority class sample x_i and one of its nearest neighbors x_j.

Undersampling with Random Selection:

$$S\_new = \{x\_i \in S \mid \text{if } P(x\_i) > (N\_minority / N\_majority)\}$$

This equation selects a subset S_new of the majority class samples S, where P(x_i) is a random probability and N_minority and N_majority are the numbers of samples in the minority and majority classes, respectively.

Weighted Resampling:

$$w\_i = 1 / p\_i \text{ where } p\_i = N\_class\_i / N\_total, S\_new = \text{Resample}(S, w)$$

Here, w_i is the weight assigned to each class i inversely proportional to its probability p_i, where N_class_i is the number of samples in class i and N_total is the total number of samples. This weight is used to resample the dataset to balance the classes.

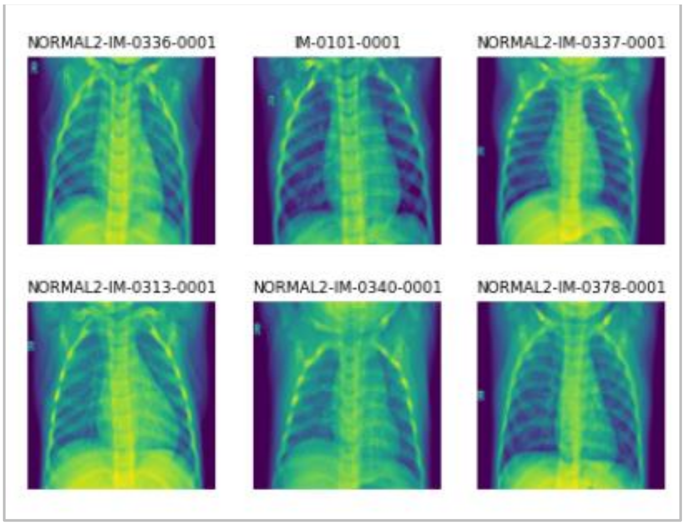

Figure 4: Data result after performing pre-process operation

B. Data Augmentation

Information expansion could be a significant method in profound learning, especially for restorative imaging assignments where the accessible datasets may be constrained. By falsely extending the dataset, expansion upgrades demonstrate strength and generalization. Key increase methods incorporate applying a revolution run of 20 degrees to capture changeability in picture introduction and rescaling pixel values by 1/255 to normalize the information, guaranteeing reliable input to the neural arrange. Also, arbitrary shifts in width and stature by 20% reenact distinctive imaging conditions, whereas shear and zoom ranges of 20% present viewpoint mutilations and scale varieties. Level flipping is utilized to reflect pictures, successfully multiplying the dataset estimate. The fill_mode='nearest' guarantees that lost pixels amid change are suitably filled, protecting picture judgment.

Adding variation to a dataset by spinning pictures around their center is a popular way to work with data enhancement. Changing the values of each pixel is how the math model for rotating a picture by an angle θ around the origin (which is usually the middle of the image) works.

Given an original pixel at coordinates (x, y), the new coordinates (x', y') after rotation by an angle θ are calculated using the following rotation matrix:

[x'] = [cosθ -sinθ] [x]

[y'] [sinθ cosθ] [y]

Expanding this, we get:

x' = x * cosθ - y * sinθ

y' = x * sinθ + y * cosθ

Here, θ is the rotation angle in radians. The new pixel coordinates (x', y') represent the position of the pixel after the image has been rotated. This transformation is applied to every pixel in the image to achieve the desired rotation effect, with interpolation used to handle non-integer pixel coordinates.

C. Classification Algorithm

1. CNN

Profound learning is based on Convolutional Neural Systems (CNNs), which are particularly great at picture classification errands like finding COVID-19 pneumonia in chest X-rays. CNNs utilize numerous building squares, like convolution layers, pooling layers, and completely connected layers, to consequently and adaptively learn the spatial requesting of highlights. They do this through backpropagation. CNNs can effectively discover patterns and characteristics connected to pneumonia when it comes to classifying chest X-rays. CNN's to begin with layers ordinarily drag out low-level highlights like lines and surfaces. Afterward layers, on the other hand, choose up more complicated designs, just like the particular shapes and surfaces connected to COVID-19 pneumonia. A bunch of learnable channels slide over the input picture amid the convolution prepare, making feature maps that appear the foremost critical designs.

Algorithm:

Step 1: Input Layer and Preprocessing

- Mathematical Model:

X ∈ R^(h × w × c) where h = 224, w = 224, c = 3

Here, X represents the input image tensor with height h, width w, and channels c. For grayscale images, c = 1; for color images, c = 3.

- Algorithm:

- Input the chest X-ray image.

- Resize the image to 224x224 pixels.

- Normalize pixel values by rescaling between 0 and 1.

Step 2: Convolutional Layers

- Mathematical Model:

Z_l = W_l * X_(l-1) + b_l

where l is the layer index

W_l and b_l are the weights and biases for the l-th layer, * denotes the convolution operation, and Z_l is the output feature map.

- Algorithm:

  - Apply multiple convolutional layers with ReLU activation.

  - Use filters to extract features such as edges, textures, and patterns indicative of pneumonia.

Step 3: Pooling Layers

- Mathematical Model:

  P_l = max(Z_l[i:i+k, j:j+k])

where k × k is the pooling window size

P_l represents the pooled feature map after applying a max-pooling operation over a k × k window.

- Algorithm:

  - Apply max-pooling to reduce the dimensionality of feature maps.

  - Downsample the feature maps to retain important features while reducing computational complexity.

Step 4: Fully Connected Layers

- Mathematical Model:

  f(W_f * flatten(P_l) + b_f)

Where W_f and b_f are the weights and biases of the fully connected layer, and f is the activation function (e.g., ReLU).

- Algorithm:

  - Flatten the pooled feature maps into a vector.

  - Pass the vector through one or more fully connected layers to integrate the learned features.

Step 5: Output Layer and Classification

- Mathematical Model:

  y_hat = softmax(W_o * f + b_o)

Here, y_hat is the predicted probability distribution across the classes (COVID-19, other pneumonia, normal), and W_o and b_o are the weights and biases for the output layer.

- Algorithm:

  - Apply the softmax function in the output layer to obtain probabilities for each class.

  - Classify the input image based on the highest probability.

After convolutional layers, pooling layers are added to diminish the number of spatial factors within the include maps. This makes the computations less difficult and reduces the chance of overfitting. Finally, completely connected layers utilize the learned highlights to put the picture into one of a few set bunches, such as typical, bacterial pneumonia, or COVID-19 pneumonia. The show can precisely analyze pneumonia from lung X-rays by utilizing CNNs, as model analysis represent in figure 5. This makes it much simpler for specialists to create fast and adjust choices amid the widespread.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 64)	640
max_pooling2d (MaxPooling2D)	(None, 111, 111, 64)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	18,464
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 32)	9,248
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 32)	0
conv2d_3 (Conv2D)	(None, 24, 24, 32)	9,248
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 32)	0
dropout (Dropout)	(None, 12, 12, 32)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1,179,904
dense_1 (Dense)	(None, 2)	514

Total params: 1,218,018 (4.65 MB)
Trainable params: 1,218,018 (4.65 MB)
Non-trainable params: 0 (0.00 B)

Figure 5: Representation of training and no trainable parameter for CNN-BC

2. EfficientNet

EfficientNet could be a state-of-the-art convolutional neural arrange (CNN) design outlined for optimizing both precision and computational proficiency. For the errand of recognizing COVID-19 pneumonia in chest X-ray pictures, EfficientNet offers a critical advantage due to its compound scaling strategy, which consistently scales arrange profundity, width, and determination, coming about in an exceedingly proficient demonstrate with less parameters and quicker preparing times. EfficientNet's capacity to preserve tall precision with lower computational costs makes it especially well-suited for sending in resource-constrained situations, such as clinics with constrained computing foundation. When connected to chest X-ray classification, as shown in figure 6. EfficientNet can viably capture complex designs characteristic of pneumonia, empowering fast and exact conclusion whereas minimizing the equipment necessities.

Algorithm:

1. **Input Data:**
   - Load chest X-ray images from the dataset.
   - Resize each image to the input size required by EfficientNet (e.g., 224x224 pixels).
2. **Data Preprocessing:**
   - Normalize the pixel values by rescaling between 0 and 1.
   - Optionally apply data augmentation techniques such as rotation, zoom, and flipping to increase dataset diversity.
3. **Model Initialization:**
   - Initialize the EfficientNet model (e.g., EfficientNetB0) pre-trained on ImageNet.

- Replace the top layer with a new fully connected layer tailored for binary classification (COVID-19 vs. non-COVID-19).

4. **Training:**
   - Compile the model using an appropriate optimizer (e.g., Adam) and loss function (e.g., binary cross-entropy).
   - Train the model on the training dataset, using a validation set for monitoring performance.
   - Apply early stopping to prevent overfitting.
5. **Evaluation and Prediction:**
   - Evaluate the model's performance on the test dataset using metrics such as accuracy, precision, and recall.
   - Use the trained model to predict COVID-19 pneumonia in new chest X-ray images by outputting the probability of each class.

Model: "functional_24"

Layer (type)	Output Shape	Param #
input_layer_12 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_5 (GlobalAveragePooling2D)	(None, 1280)	0
dense_10 (Dense)	(None, 128)	163,968
dropout_2 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 64)	8,256
dense_12 (Dense)	(None, 2)	130

Total params: 4,221,925 (16.11 MB)
Trainable params: 4,179,902 (15.95 MB)
Non-trainable params: 42,023 (164.16 KB)

Figure 6: Representation of training and no trainable parameter for EfficientNet model

3. Lightweight_MobileNet

Lightweight MobileNet is an greatly compelling convolutional neural organize (CNN) system that's outlined to work best with portable and implanted gadgets. MobileNet's primary advantage for finding COVID-19 pneumonia in lung X-rays is that it can do a extraordinary work with exceptionally few computing assets. This makes it culminate for places with constrained computing control, like inaccessible clinics or portable wellbeing units. MobileNet is proficient since it employments depthwise distinct convolutions, which cut down on the number of variables and computations whereas keeping execution the same. MobileNet can precisely record critical highlights on chest X-rays, permitting for fast and exact discovery on gadgets with restricted handling control, like smartphones or versatile therapeutic hardware. This makes symptomatic devices more accessible in places that do not have sufficient of them, result of model implantation shown in figure 7.

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer_6 (InputLayer)	(None, 224, 224, 3)	0	-
Conv1 (Conv2D)	(None, 112, 112, 32)	864	input_layer_6[0][0]
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 112, 112, 32)	0	bn_Conv1[0][0]
expanded_conv_depthwise (DepthwiseConv2D)	(None, 112, 112, 32)	288	Conv1_relu[0][0]
block_16_project (Conv2D)	(None, 7, 7, 320)	307,200	block_16_depthwise_re…
block_16_project_BN (BatchNormalization)	(None, 7, 7, 320)	1,280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409,600	block_16_project_BN[0…
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5,120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
flatten_6 (Flatten)	(None, 62720)	0	out_relu[0][0]
dropout_6 (Dropout)	(None, 62720)	0	flatten_6[0][0]
dense_18 (Dense)	(None, 1024)	64,226,304	dropout_6[0][0]
dense_19 (Dense)	(None, 512)	524,800	dense_18[0][0]
dense_20 (Dense)	(None, 2)	1,026	dense_19[0][0]

Total params: 67,010,114 (255.62 MB)
Trainable params: 64,752,130 (247.01 MB)
Non-trainable params: 2,257,984 (8.61 MB)

Figure 7: Representation of training and no trainable parameter for Lightweight_MobileNet model

Algorithm:

1. Input Data:

$X \in R^{(h \times w \times c)}$ where $h = 224$, $w = 224$, $c = 3$

X represents the input chest X-ray image resized to 224x224 pixels.

- Step:

- Load and preprocess the chest X-ray images, resizing them to 224x224 pixels and normalizing pixel values between 0 and 1.

2. Depthwise Convolution:

$Z_d(x, y, k) = \Sigma W_d(i, j, k) * X(x+i, y+j, k)$

Here, Z_d is the output of the depthwise convolution, W_d are the depthwise convolution filters, and X is the input.

- Step:

- Apply depthwise convolution, where each filter is applied to a single input channel, reducing computational complexity.

3. Pointwise Convolution:

$Z_p(x, y, m) = \Sigma W_p(k, m) * Z_d(x, y, k)$

Z_p is the output after pointwise convolution, and W_p are the pointwise filters.

- Step:

- Perform pointwise convolution with a 1x1 kernel to combine the features from the depthwise convolution, effectively learning the spatial relationships.

4. Fully Connected Layer:

$y = \sigma(W_f * \text{flatten}(Z_p) + b_f)$

W_f and b_f are the weights and biases of the fully connected layer, and σ is the activation function.

- Step:

- Flatten the feature maps and pass them through a fully connected layer to integrate the features for classification.

5. Output Layer:

$\hat{y} = \text{softmax}(W_o * y + b_o)$

y_hat represents the predicted probability distribution, with W_o and b_o as the output layer weights and biases.

- Step:

- Apply the softmax function to produce the final classification probabilities for COVID-19 pneumonia and other classes.

4. Lightweight_EfficientNet

Lightweight EfficientNet could be a little and efficient adaptation of the EfficientNet system that's made to work well on portable and inserted gadgets. It keeps the most excellent parts of EfficientNet, like compound scaling, which makes beyond any doubt that the network's profundity, width, and determination are all adjusted for the leading exactness and speed. Since Lightweight EfficientNet can give tall exactness with few computing assets, it is particularly great at finding COVID-19 pneumonia in chest X-ray pictures. By cutting down on the number of components and computing needs, it makes real-time examination conceivable on gadgets like smartphones and mobile restorative apparatuses that do not have a part of preparing control, as reflect in figure 8. Since of this, it is idealize for places with constrained assets, where fast and precise assessment is required for healthcare to work well amid the widespread.

Algorithm:

1. Input Data:

$X \in R^{(h \times w \times c)}$ where $h = 224$, $w = 224$, $c = 3$

X represents the input chest X-ray image, resized to 224x224 pixels, with three color channels (or 1 for grayscale).

- Step:

- Load and preprocess the chest X-ray images, resizing them to 224x224 pixels.

- Normalize pixel values to the [0, 1] range.

2. Compound Scaling in Convolutional Layers:

$Z_l = W_l * X_{(l-1)} + b_l$, with scaling factor $\phi = \alpha^d * \beta^w * \gamma^r$

Here, W_l and b_l are the weights and biases of the convolutional layer, * denotes the convolution operation, and ϕ is the compound scaling factor that scales the network's depth d, width w, and resolution r uniformly.

- Step:

- Apply depthwise separable convolution layers with compound scaling, adjusting depth, width, and resolution to optimize for both accuracy and efficiency.

3. Efficient Feature Extraction:

Z_p = SE(Z_l) * Z_l

Z_p represents the output after applying a squeeze-and-excitation (SE) block, which adaptively recalibrates channel-wise feature responses by modeling interdependencies between channels.

- Step:

- Use SE blocks within convolutional layers to enhance important features and suppress less useful ones, improving feature extraction efficiency.

4. Classification with Fully Connected Layers:

y_hat = softmax(W_f * flatten(Z_p) + b_f)

y_hat is the predicted probability distribution across the classes, with W_f and b_f representing the weights and biases of the fully connected layer.

- Step:

- Flatten the feature maps and pass them through a fully connected layer.

- Apply the softmax function to generate the final classification probabilities for COVID-19 pneumonia and other conditions.

Model: "functional_30"

Layer (type)	Output Shape	Param #
input_layer_18 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_8 (GlobalAveragePooling2D)	(None, 1280)	0
dense_18 (Dense)	(None, 64)	81,984
dropout_5 (Dropout)	(None, 64)	0
dense_19 (Dense)	(None, 2)	130

Total params: 4,131,685 (15.76 MB)
Trainable params: 82,114 (320.76 KB)
Non-trainable params: 4,049,571 (15.45 MB)

Figure 8: Representation of training and no trainable parameter for Lightweight_EfficientNet model

5. RESULT ANALYSIS

A. Model Analysis

Figure 9 appears the exactness (cleared out) and misfortune (right) for preparing and assessment over five ages. The exactness of the preparing goes up consistently and comes to over 95% by the fifth age.

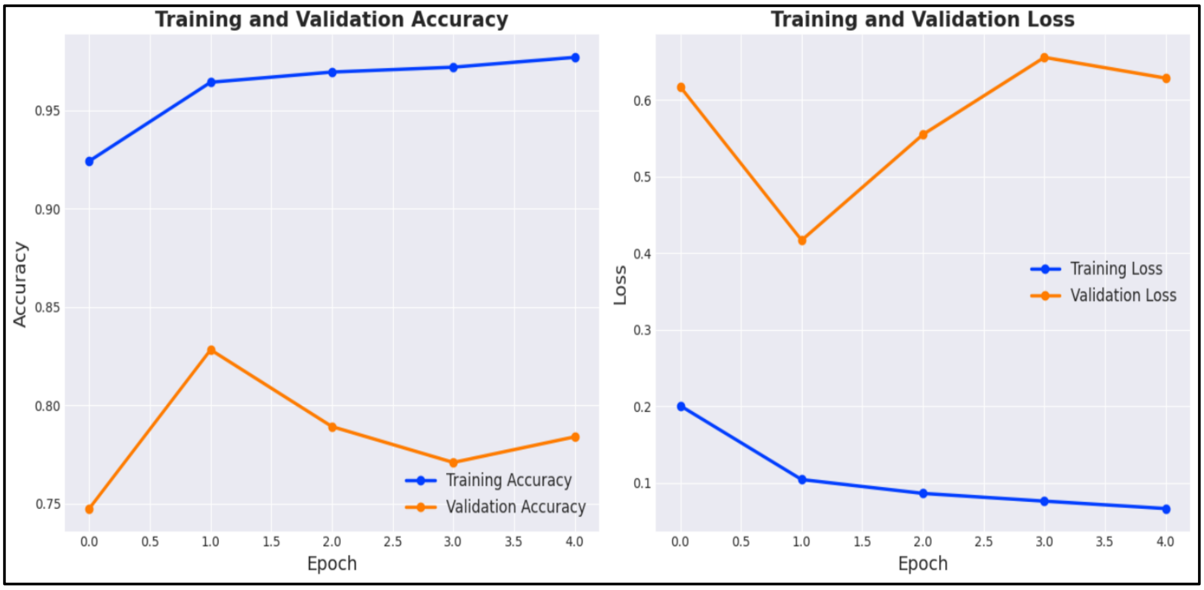

Figure 9. CNN model Accuracy and Loss Curve

On the other hand, the precision of the approval begins out at approximately 75%, crests within the second age, and after that goes down, which may cruel that the demonstrate is too great. The preparing misfortune gradually goes down, which implies the show is learning well amid preparing. But after the primary age, the approval misfortune goes up, which is more confirmation of over fitting since the demonstrate doesn't work well with information it hasn't seen some time recently. This design appears that regularization strategies or early ceasing are required to form the show work superior on the approval set and halt it from getting to be as well great at what it does.

The preparing and assessment exactness (cleared out) and misfortune (right) over ten ages are appeared in Figure 10. It's continuously been exceptionally precise amid training—very near to 100%—which implies the model is learning the preparing information well. The affirmation precision, on the other hand, changes a part. It crests around the fifth arrange and after that remains unsteady, which focuses to overfitting. The misfortune amid preparing is little and steady, but the misfortune amid approval is exceptionally unsteady, with spikes appearing that the demonstrate is having inconvenience generalizing to the approval set. These patterns make me think that the demonstrate is fitting as well well, which might cruel that it ought to be regularized or ceased early.

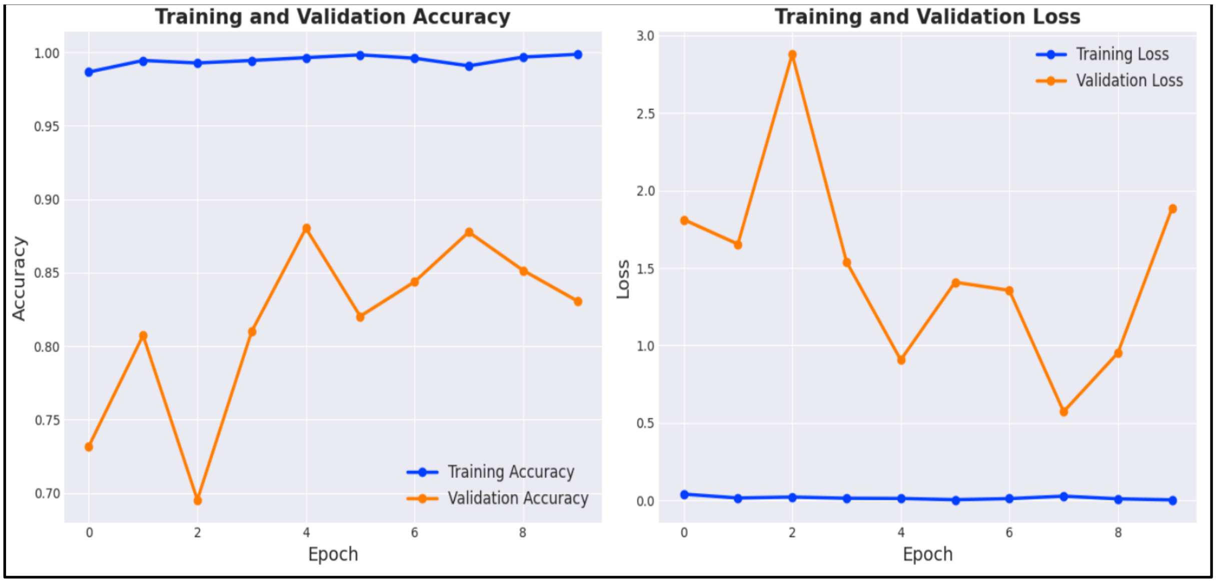

Figure 10. EfficientNet model Accuracy and Loss Curve

The preparing and assessment exactness (cleared out) and loss (right) over nine ages are appeared in Figure 11. The training accuracy goes up gradually but unquestionably until it hits over 95%. This appears that the demonstrate learns from the preparing information accurately. The affirmation exactness, on the other hand, changes a parcel. It peaks around the fifth phase and after that drops, which recommends that it doesn't work reliably with information it hasn't seen some time recently. The preparing misfortune goes down gradually, which suggests that the individual did way better on the preparing set. The legitimacy misfortune, on the other hand, is exceptionally unsteady, with crests and plunges that appear issues with demonstrate generalization. Overfitting is when the demonstrate learns the preparing information too well but has inconvenience with unused information it hasn't seen some time recently. This precariousness might cruel that the demonstrate needs more tuning or regularization strategies.

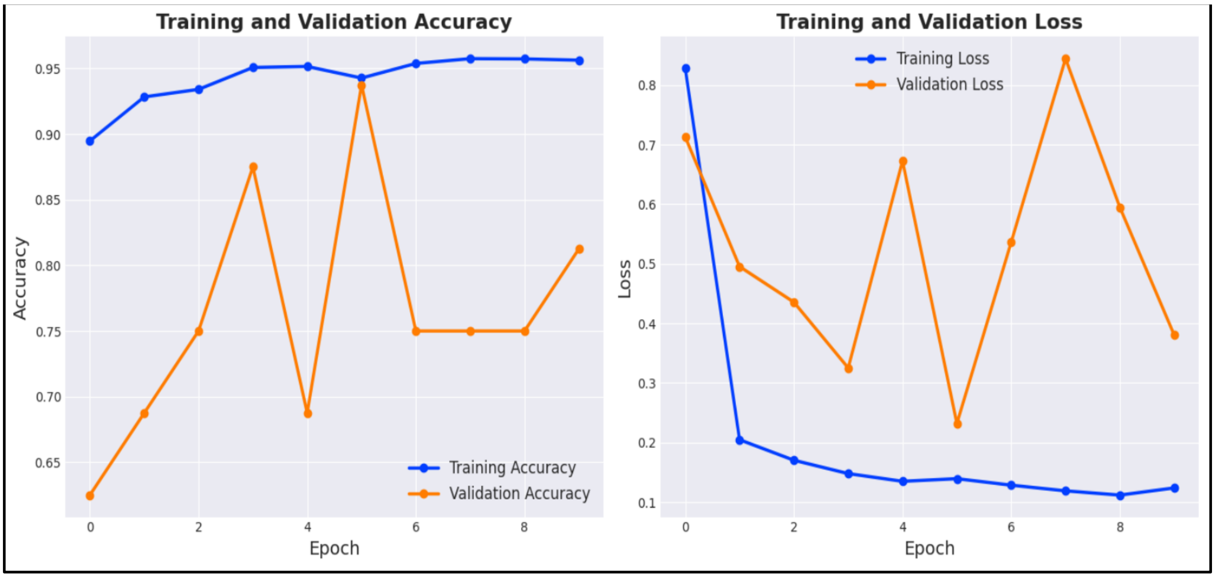

Figure 11. Lightweight_MobileNetV2 model Accuracy and Loss Curve

The preparing and assessment exactness (cleared out) and misfortune (right) over nine ages are appeared in Figure 12. The preparing exactness keeps going up until it gets near to 100%. This implies that the demonstrate is learning from the preparing information.

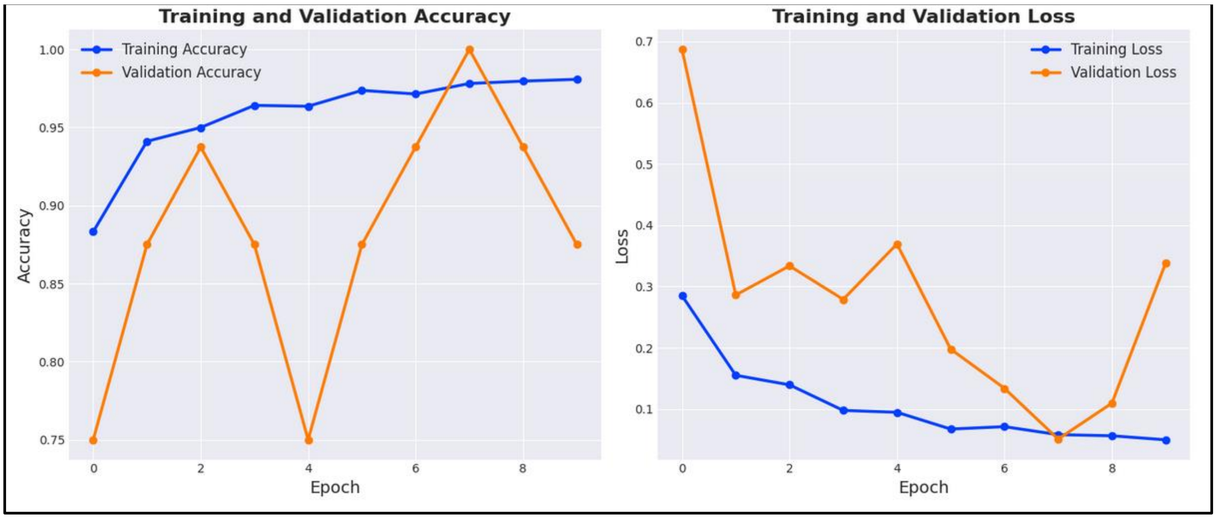

Figure 12. Lightweight_EfficientNet model Accuracy and Loss Curve

In any case, the approval exactness changes a part, with crests and valleys, which proposes that the demonstrate isn't steady when it comes to adjusting to unused information. The preparing misfortune goes down gradually over time, which implies that the demonstrate is doing superior on the preparing set. On the other hand, the approval misfortune goes down for the most part but has a few huge hops. This change in approval execution might be a sign of overfitting, which is when the demonstrate works well on the preparing information but has inconvenience remaining precise on the approval set. In this case, to progress generalization, strategies like cross-validation, dropout, or early ceasing are required.

B. Comparative Analysis

Table 2: Trainable and Non-Trainable Parameters Comparison

Model	Total_Param (MB)	Trainable_Param (MB)	NonTrainable_Param (MB)
CNN	4.65	4.65	0
EfficientNet	16.11	15.95	0.164
Lightweight_MobileNet	245.62	247.01	8.61
Lightweight_EfficientNet	15.76	0.376	15.45

Table 2 shows a comparison of factors that can be trained and those that can't be trained for four models: CNN, EfficientNet, Lightweight MobileNet, and Lightweight EfficientNet. The total number of parameters, trainable parameters, and non-trainable parameters are given in megabytes (MB), which shows how much computing power each model needs. The CNN model, which is 4.65 MB in size, is made up of only trainable factors, repr4esent in figure 13. This means the model is pretty basic and light, and all of its settings can be changed during training to make it work better. The fact that there are no non-trainable parameters suggests that the CNN model does not have any set layers or components that have already been trained. This is typical of basic CNN designs where all parameters are learned from scratch during training. This makes the CNN model simple, making it perfect for situations where computing power is restricted and a simpler model is enough.

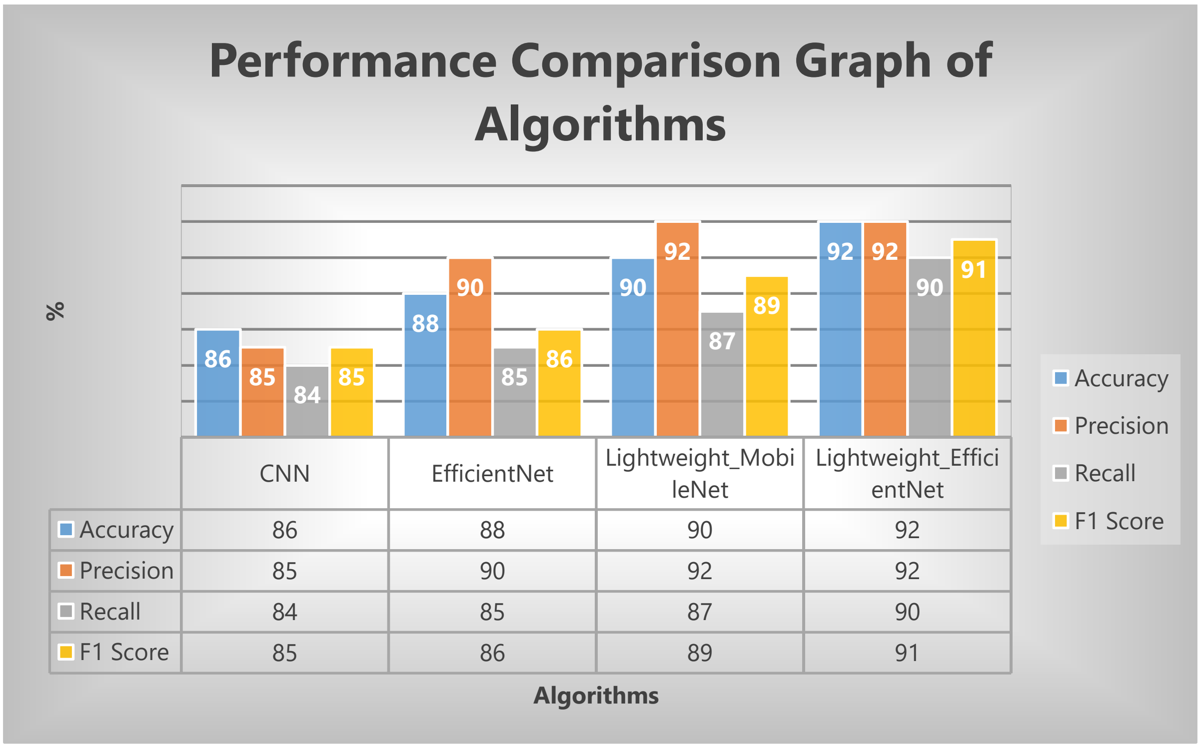

Figure 13: Performance Comparison Graph of Algorithms

EfficientNet, on the other hand, has 16.11 MB of parameters, with 15.95 MB that can be trained and 0.164 MB that can't. The fact that there are factors that can't be learned shows that the model uses weights that have already been trained and aren't changed during training. EfficientNet is made with a compound scale method that keeps productivity high while improving accuracy. There aren't many factors in this model that can't be trained, which suggests that it has been fine-tuned for this job, probably with help from knowledge from big datasets like

ImageNet. This mix between factors that can be trained and those that can't be trained lets EfficientNet work well with less computer power. The Lightweight MobileNet model stands out because its total parameter size is 245.62 MB, which is made up of 247.01 MB of trainable parameters and 8.61 MB of non-trainable parameters. There are factors that can't be learned, which means that the system relies a lot on layers that have already been trained. This is a key part of MobileNet designs that are made to work well on mobile and embedded devices. However, the difference between the parameters that can be trained and those that can't shows that MobileNet's model is much more complicated and larger than the others, even though it is less computationally intensive. This might be because of its depth-wise separable convolutions and the fact that the model can be used for different tasks, which need a lot of parameters to be improved during training.

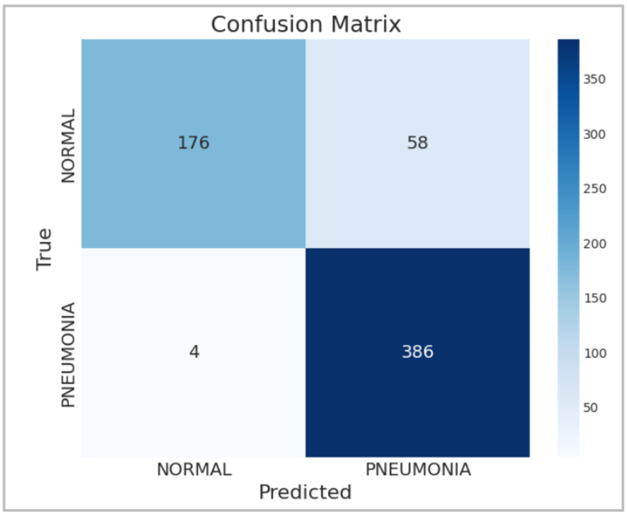

Figure 14. Confusion Matrix of Lightweight_MobileNet Model

If you look at Lightweight EfficientNet's 15.76 MB of parameters, you'll notice that only 0.376 MB of them can be trained, while 15.45 MB can't. This model is highly efficient for situations where there aren't a lot of computing resources, the Confusion Matrix of Lightweight MobileNet Model is shown in figure 14. A lot of factors that can't be trained show that the model is mostly made up of weights that have already been trained, with only a small part being fine-tuned for the current job. This method cuts down on training time and computing needs by a large amount. This makes Lightweight EfficientNet a great choice for real-time apps or devices with limited processing power. The basic trainable parameters show how well the model can change to new tasks without having to be trained all over again. This is possible because the non-trainable layers store information that can be used right away.

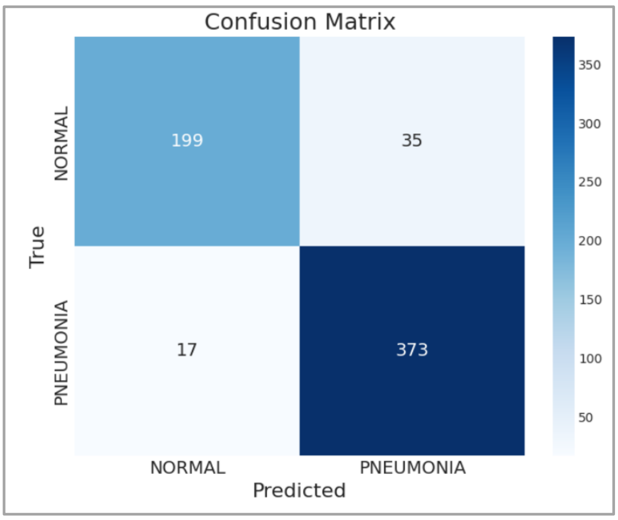

Figure 15. Confusion Matrix of Lightweight_EfficientNet Model

The comparison of these models shows the trade-offs between how complicated the model is, how easy it is to train, and how efficient it is. The CNN model is simple and easy to train because all of its features can be changed. However, it may not be as smart as more advanced models. EfficientNet finds a balance by including a small number of factors that can't be trained. This improves its performance through transfer learning. Even though Lightweight MobileNet works quickly, as confusion matrix shown in figure 15, the models are getting much bigger and more complicated. Lightweight EfficientNet is very efficient because it only needs a few trainable factors. This makes it perfect for situations where computing power is limited and quick rollout is needed. Each model is good for a different purpose, and the right one to use relies on the job at hand, its accuracy, speed, and the resources that are available.

6. CONCLUSION

The recommended profound learning-based approach for spotting COVID-19 pneumonia in chest X-rays appears a part of guarantee for progressing the exactness and speed of analyze in clinical circumstances. This framework employments progressed convolutional neural systems (CNNs) and their varieties, like EfficientNet, Lightweight MobileNet, and Lightweight EfficientNet, to discover complicated designs in restorative pictures that are demonstrative of COVID-19-related pneumonia. Each sort has its possess benefits when it comes to precision, computing speed, and being able to be utilized in places with restricted assets. The system frequently accomplished tall preparing exactness amid the survey handle, appearing that it may learn from the information. Be that as it may, the changes seen in affirmation exactness and misfortune appear how difficult it is to apply to information that hasn't been seen some time recently. These issues appear how vital it is to utilize regularization strategies, like dropout and early ceasing, to halt models from overfitting and make them more solid. The comparison of trainable and non-trainable components over the different models too highlights the trade-offs between show complexity and execution, which makes a difference select the proper models based on the application's one of a kind needs. At long last, the profound learning-based framework looks like a great way to rapidly and precisely spot COVID-19 pneumonia from chest X-rays, which is exceptionally vital within the current battle against the widespread. Both its capacity to work with diverse equipment setups and its capacity to keep up tall execution with lower computing needs make it a great choice for utilize in a run of healthcare settings. More work ought to be done to form the show indeed better at generalization and to see how it can work with other symptomatic apparatuses to form a total and precise strategy for diagnosing COVID-19 and other breathing issues.

REFERENCES

[1] Pandey, S.K.; Mohanta, G.C.; Kumar, V.; Gupta, K. Diagnostic Tools for Rapid Screening and Detection of SARS-CoV-2 Infection. Vaccines 2022, 10, 1200.

[2] Vernikou, S.; Lyras, A.; Kanavos, A. Multiclass sentiment analysis on COVID-19-related tweets using deep learning models. Neural Comput. Appl. 2022, 34, 19615–19627.

[3] Xie, X.; Zhong, Z.; Zhao, W.; Zheng, C.; Wang, F.; Liu, J. Chest CT for Typical Coronavirus Disease 2019 (COVID-19) Pneumonia: Relationship to Negative RT-PCR Testing. Radiology 2020, 296, E41–E45.

[4] Livieris, I.E.; Kanavos, A.; Tampakas, V.; Pintelas, P.E. An Ensemble SSL Algorithm for Efficient Chest X-ray Image Classification. J. Imaging 2018, 4, 95.

[5] Fiszman, M.; Chapman, W.W.; Aronsky, D.; Evans, R.S.; Haug, P.J. Research Paper: Automatic Detection of Acute Bacterial Pneumonia from Chest X-ray Reports. J. Am. Med. Inform. Assoc. 2000, 7, 593–604.

[6] Rai, P.; Kumar, B.K.; Deekshit, V.K.; Karunasagar, I.; Karunasagar, I. Detection Technologies and Recent Developments in the Diagnosis of COVID-19 Infection. Appl. Microbiol. Biotechnol. 2021, 105, 441–455.

[7] Bernheim, A.; Mei, X.; Huang, M.; Yang, Y.; Fayad, Z.A.; Zhang, N.; Diao, K.; Lin, B.; Zhu, X.; Li, K.; et al. Chest CT Findings in Coronavirus Disease-19 (COVID-19): Relationship to Duration of Infection. Radiology 2020, 295, 685–691.

[8] 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.

[9] Saeed, M.; Ahsan, M.; Saeed, M.H.; Rahman, A.U.; Mehmood, A.; Mohammed, M.A.; Jaber, M.M.; Damaševičius, R. An Optimized Decision Support Model for COVID-19 Diagnostics Based on Complex Fuzzy Hypersoft Mapping. Mathematics 2022, 10, 2472.

[10] Ieracitano, C.; Mammone, N.; Versaci, M.; Varone, G.; Ali, A.-R.; Armentano, A.; Calabrese, G.; Ferrarelli, A.; Turano, L.; Tebala, C.; et al. A fuzzy-enhanced deep learning approach for early detection of COVID-19 pneumonia from portable chest X-ray images. Neurocomputing 2022, 481, 202–215.

[11] D'Arienzo, M.; Coniglio, A. Assessment of the SARS-CoV-2 basic reproduction number, R0, based on the early phase of COVID-19 outbreak in Italy. Biosaf. Health 2020, 2, 57–59.

[12] Rossman, H.; Meir, T.; Somer, J.; Shilo, S.; Gutman, R.; Ben Arie, A.; Segal, E.; Shalit, U.; Gorfine, M. Hospital load and increased COVID-19 related mortality in Israel. Nat. Commun. 2021, 12, 1904.

[13] Yaseliani, M.; Hamadani, A.Z.; Maghsoodi, A.I.; Mosavi, A. Pneumonia Detection Proposing a Hybrid Deep Convolutional Neural Network Based on Two Parallel Visual Geometry Group Architectures and Machine Learning Classifiers. IEEE Access 2022, 10, 62110–62128.

[14] Self, W.H.; Courtney, D.M.; McNaughton, C.D.; Wunderink, R.G.; Kline, J.A. High discordance of chest X-ray and computed tomography for detection of pulmonary opacities in ED patients: Implications for diagnosing pneumonia. Am. J. Emerg. Med. 2013, 31, 401–405.

[15] Ticinesi, A.; Lauretani, F.; Nouvenne, A.; Mori, G.; Chiussi, G.; Maggio, M.; Meschi, T. Lung ultrasound and chest X-ray for detecting pneumonia in an acute geriatric ward. Medicine 2016, 95, e4153.

[16] Htun, T.P.; Sun, Y.; Chua, H.L.; Pang, J. Clinical features for diagnosis of pneumonia among adults in primary care setting: A systematic and meta-review. Sci. Rep. 2019, 9, 7600.

[17] 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.

[18] 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.

[19] R. Golchha, P. Khobragade and A. Talekar, "Design of an Efficient Model for Health Status Prediction Using LSTM, Transformer, and Bayesian Neural Networks," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-5, doi: 10.1109/ICICET59348.2024.10616353.

[20] Zhou, S.K.; Greenspan, H.; Davatzikos, C.; Duncan, J.S.; Van Ginneken, B.; Madabhushi, A.; Prince, J.L.; Rueckert, D.; Summers, R.M. A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises. Proc. IEEE 2021, 109, 820–838.

[21] Irvin, J.; Rajpurkar, P.; Ko, M.; Yu, Y.; Ciurea-Ilcus, S.; Chute, C.; Marklund, H.; Haghgoo, B.; Ball, R.; Shpanskaya, K.; et al. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. Proc. AAAI Conf. Artif. Intell. 2019, 33, 590–597.

[22] Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; Summers, R.M. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3462–3471.

[23] Motamed, S.; Rogalla, P.; Khalvati, F. Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images. Inform. Med. Unlocked 2021, 27, 100779.

[24] Dataset: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia/code

[25] Wankhede, D. S., Kalra, N., Dhabliya, R., Khetani, V., Waykole, T., & Shirkande, A. S. (2023, November). Enhancing Alzheimer's Disease Prediction with Bayesian Optimization and Ensemble Methods. In *Proceedings of the 5th International Conference on Information Management & Machine Intelligence* (pp. 1-6).