INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

# An Efficient Hypertuned DNN Based Approach for Pneumonia Detection

**Sayali Abhijeet Salkade**[1]*, **Sheetal Rathi**[1]

[1] Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai, 400101, Maharashtra, India

OPEN ACCESS

**Received:** 12-04-2024

**Accepted:** 04-06-2024

**Published:** 25-06-2024

**Citation:** Salkade SA, Rathi S (2024) An Efficient Hypertuned DNN Based Approach for Pneumonia Detection. Indian Journal of Science and Technology 17(25): 2667-2678. https://doi.org/10.17485/IJST/v17i25.1230

* **Corresponding author**.

salkadesayali6@gmail.com

**Funding:** None

**Competing Interests:** None

Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

## Abstract

**Objectives:** To create a deep learning system for pneumonia detection that is both effective and gradually optimized. **Methods:** A customized CNN is used with an incremental approach for pneumonia classification and detection. Starting with a baseline model, hypertuning parameters such as four convolution layers with filters of 16, 32, 64, and 128 sizes, a dropout layer with values of 0.3, 0.5, and 0.7, four batch normalization layers, and an Adam optimizer are added. A total images of 5,863 for training, 624 for testing, and 16 for validation from the Paul Mooney dataset were used to test the suggested model. **Findings:** The study recorded a test accuracy of 94% for the customized CNN followed by ResNet50 at 79.9%, VGG16 at 90.14%, VGG19 at 82.21%, InceptionV3 at 74.51%, and EfficientNetB1 at 83.17%. Recall of 98.20%, accuracy of 85.55%, AUC of 93.52%, and F1_score of 92.45% obtained were all fairly excellent for the customized CNN. 15 epochs, a learning rate of 0.0001, callbacks with a patience of 3, and an early stopping feature were applied to the training model. **Novelty**: Five convolution blocks, two separable convolution layers, one batch normalization layer, one maxpooling layer, and a fully connected layer with an Adam optimizer were all included in the customized CNN that was developed to identify and categorize pneumonia. With Explainable AI's GradCAM technology, pneumonia-infected areas on chest X-rays were highlighted and the sickness was seen.

**Keywords:** Customized CNN; VGG16; VGG19; ResNet50; Explainable AI

## 1 Introduction

Pneumonia, a common respiratory infection, poses a significant health risk, especially due to its high mortality rate. Early detection and accurate diagnosis are vital for successful treatment and can significantly improve patient outcomes. Chest X-ray imaging is one of the primary diagnostic tools used to identify pneumonia, but its interpretation can be challenging and subjective, often requiring expert radiologists. The primary aim of this research is to evaluate the effectiveness of Convolutional Neural Networks (CNNs) in identifying pneumonia from chest X-ray images. By leveraging deep learning and image classification capabilities, this study seeks to enhance the accuracy and efficiency of pneumonia diagnosis, thereby contributing to the development of automated systems that assist healthcare professionals in quickly and accurately diagnosing patients. Recent advances in machine learning, particularly deep learning, have revolutionized various fields, including medical imaging. Deep learning models, such as CNNs, have demonstrated remarkable success in image classification tasks. These advancements provide an opportunity to develop more accurate and efficient pneumonia detection methods, potentially transforming clinical practices and improving patient care. Explainable AI techniques have been employed to evaluate the extent of disease-affected areas in chest X-ray images. Ibrahim et al.[1] developed a method for three-class pneumonia recognition using VGG and ResNet152v2 architectures, achieving recall values between 93% and 97% on their dataset of X-ray images. Ongoing research aims to improve the performance of these models by increasing the dataset size, extending training epochs, and incorporating additional deep learning techniques, such as Generative Adversarial Networks (GANs) for both classification and augmentation. Enhanced Restricted Boltzmann Machine (RBM) was created in[2] to diagnose pneumonia, overcoming the limitations of the traditional Boltzmann machine. Despite the development of numerous deep learning methods for pneumonia detection, there remains a need to enhance the precision and accuracy of these approaches. Most existing models have successfully identified both abnormal and normal cases of pneumonia but have struggled to differentiate between viral and bacterial pneumonia. Quan and colleagues[3] combined DenseNet and CapsNet architectures to achieve a 96% recall rate in a smaller COVID-19 dataset. However, their model's accuracy dropped to 90.7% when applied to a larger set of pneumonia-related X-ray images, indicating a need for further improvement in handling larger datasets. Alhudhaif et al.[4] utilized a DenseNet-201-based architecture, achieving recall and accuracy levels of up to 95% and approximately 90% precision. However, the results were not significant enough due to the need for greater precision and the issue of lossy image detection necessitating feature changes. Singh and Yow[5] introduced an interpretable deep learning neural network technique based on ProtoPNet and NP-ProtoPNet, achieving an accuracy of 87.27%. The convex optimization of the final layer, however, prolonged model training times. Horry MJ et al.[6] contributed the concept of data fusion, allowing the use of multiple data modalities to improve model classification performance. The study's limitation was the small dataset size. Wang et al.[7] created a prior attention residual learning block and integrated it with two 3D-ResNets to achieve 93.3% accuracy using 4,697 X-ray images. Wang Z and colleagues[8] introduced a data fusion method to improve COVID-19 classification performance by integrating multiple data modalities. One limitation was the lack of classification for images related to health, viruses, germs, or COVID-19, necessitating further risk assessment and detailed examination of tissue infected with COVID-19. Quan et al.[3] combined DenseNet and CapsNet designs, achieving 90.7% accuracy on a larger collection of pneumonia X-ray images but only 96% recall on a small COVID-19 dataset. Brunese et al.[9] developed a three-class pneumonia diagnosis framework using the GradCAM technique for visual debugging and VGG-16 architecture, achieving 99% accuracy. However, the model was limited to considering only one type of lung disease. Suryaa S. V[10] developed an ensemble learning technique using VGG-16, ResNet-50, ResNet-101, ResNet-152, and VGG-19 for pneumonia diagnosis. Hashmi F. M. et al. built a weighted classifier using ResNet-18, Xception, InceptionV3, DenseNet121, and MobileNetV3 to identify pneumonia. The model's shortcomings included inadequate dataset annotation, a lack of datasets, and the need for a more in-depth explanation of the model's predictions[11]. Studies[12–15] investigated lung GAN, R-CNN, ensemble learning, and modified VGG16 approaches for pneumonia identification. Farukh H. M. employed a weighted classifier to identify pneumonia[11]. Studies[11,16–18] highlighted the limitations imposed by the scarcity of datasets. A weakly supervised method was used in[19] to identify pneumonia cases, and lightweight DenseNet-121 was utilized in[20] to extract features. With limited datasets, Xception and DenseNet121 achieved the highest accuracy for pneumonia diagnosis[13,21]. Multidilation convolution networks and ensemble learning approaches were applied in[22,23] to identify COVID-19 and pneumonia, respectively.

## To address the limitations of current state-of-the-art methods, we propose the following contributions:

1. ***Customized CNN Architecture:***
   - We developed a customized Convolutional Neural Network (CNN) featuring five distinct convolutional blocks, a flattening layer, and four fully connected layers with dropout.
   - Each layer employs a relu activation function, and the final layer uses a sigmoid activation function.
   - This architecture achieved a high accuracy of 93.75% in pneumonia detection.

2. ***Hypertuned Incremental Approach:***
   - We implemented a hypertuned incremental approach to classify pneumonia and normal images, enhancing the models performance and precision.

3. ***Activation Maps:***
   - Utilizing CNN activation maps, we highlighted disease-infected regions in the chest X-ray images, providing better visualization and interpretability of the affected areas.

4. ***Data Augmentation:***
   - To mitigate the issues related to small datasets, we employed data augmentation techniques, thereby increasing the diversity and robustness of the training data.

**Section I**: Provides an overview of the methods used for pneumonia detection. **Section II**: Discusses various approaches, including the hypertuning of CNNs and the customized CNN architecture, along with different preprocessing techniques. **Section III**: Presents the experimental results and discusses the findings. **Section IV**: Concludes the study. **Section V**: Lists the references used throughout the research.

## 2 Methodology

The project successfully identified individuals as having pneumonia or being normal using a Convolutional Neural Network (CNN) model. The model was effectively utilized for training, testing, and validating chest X-ray scans sourced from Kaggle. The process involved several key steps like data pre-processing, model training and testing, validation, incremental hypertuning.

### 2.1 Proposed System

#### *Dataset Acquisition and Setup*

The dataset used for this project, provided by Paul Mooney, was obtained from Kaggle. The necessary library files and dependencies were retrieved to process and analyze the data. The dataset includes several images of both pneumonia and normal chest X-rays, divided into three sections:

1. **Training Dataset**: Used to train the CNN model.
2. **Validation Dataset**: Used to tune the model and validate its performance during training.
3. **Test Dataset**: Used to evaluate the final performance of the model on unseen data.

The study used dataset which includes training (5216 images), testing (624 images), and validation (16 images) classes with pneumonia and normal images. There are three folders—train, test, and val, each containing a subfolder for an image category (Pneumonia/Normal) in the dataset. Anterior-posterior chest X-ray images were selected from a retrospective cohort of pediatric patients from Guangzhou Women and Children's Medical Center in Guangzhou, ranging in age from one to five. Every chest X-ray was taken as a standard component of the patient's medical care. All chest radiographs were checked for quality control by eliminating any poor-quality or unreadable scans prior to the analysis of the chest x-ray images.

#### *Data Augmentation*

To increase the volume and diversity of the data, a data augmentation function was employed. Data augmentation synthesizes new data points from the existing dataset, enhancing the model's ability to generalize and perform well on new, unseen data. There are two types of data generators used:

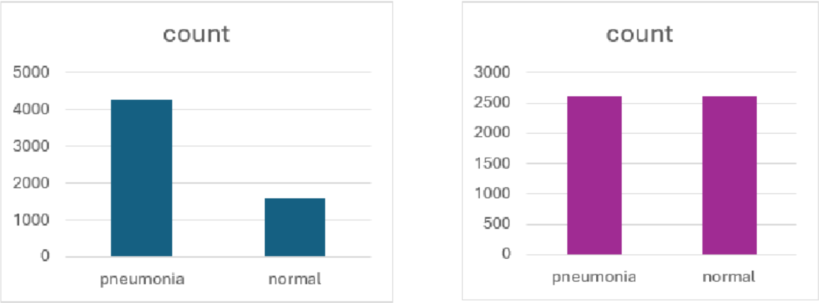
1. **Training Data Generator**.
   - Applied various transformations to the training images, such as rotation, scaling, flipping, and shifting.
   - These modifications help the model learn from a broader range of image variations.

2. **Validation and Test Data Generator**.
   - Ensured the validation and test images were processed in a consistent manner without altering their inherent characteristics.
   - Provided a stable benchmark for model evaluation.

#### *Data pre-processing techniques used:*

Following the loading of the dataset, a few pre-processing steps are required to enhance the visual information and quality of the input chest X-rays (e.g., removing noise, enhancing contrast, removing high and low frequencies from the input image etc.). In this paper, we examined two popular pre-processing methods: intensity normalization and CLAHE. One of the first uses of image processing is intensity normalization. To provide the model with the image input in a normalized format, the minmax pre-processing technique is employed. The mathematical form of minmax is given by Equation (1).

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$



(a)

(b)

**Fig 1. a. Images without augmentation b. Images with augmentation**

Also, we have used augmentation for normalizing datasets and avoiding overfitting. The Figure 1 shows how data imbalance and overfitting are reduced using the data augmentation techniques mentioned below. The method of filtering involves removing extraneous noise from an image. The process involves using a filter to modify the pixel values in the image. We can be applied in a variety of ways, depending on the kind of filter. It helps to eliminate particular kinds of noise, including speckle, salt-and-pepper, or Gaussian noise. The median, mean, and Gaussian filters are among the tools that aid in eliminating the noises listed above. One procedure that can improve image quality is enhancement. It is accomplished by adjusting the image contrast or brightness. These methods are as basic as using a histogram to change the brightness and contrast in an image, or they can be more sophisticated, like employing algorithms to improve an image's edges and textures. The technique of noise reduction was done using Gaussian and median filtering, as shown in Figure 2. Canny Edge Detection: based on the sobel filter, it is mostly used for edge detection. In essence, it computes the gradient of image intensity at each pixel, with the gradient reaching its maximum near edges where colour shifts rapidly. It was used for feature extraction. Contrast enhancement was done using histogram equalization. A mathematical method for increasing the histogram's dynamic range is called **histogram equalization**. The histogram may occasionally span a small range. Histogram equalization is an excellent method for improving images. It provides higher-quality images without sacrificing any information. Global histogram equalization is applied using this technique.

Data augmentation makes it possible to improve the model's performance and reduce the operating expenses related to gathering data. A 64-image batch size was employed.

**The following are the steps for the proposed CNN:**

- Setting up the environment, importing required libraries was done.
- Dataset was loaded into the Kaggle notebook.
- Different pneumonia and normal images from dataset were visualized.
- We perform a grayscale normalization to reduce the effect of illumination's differences. Since, the CNN converges faster on [0...1] data than on [0..255].
- **Data Augmentation:** In order to avoid overfitting problem, we need to expand artificially our dataset. We have made our existing dataset even larger. The idea is to alter the training data with small transformations to reproduce the variations. Approaches that alter the training data in ways that change the array representation while keeping the label the same are known as **data augmentation techniques.**

Data augmentation increases the dataset size by creating modified versions of the original data points. This can involve:

- **Slight Modifications**: Rotating, scaling, and flipping the original images to generate new training samples.
- **Latent Space Generation**: Using machine learning models to synthesize entirely new data points within the latent space of the original dataset.

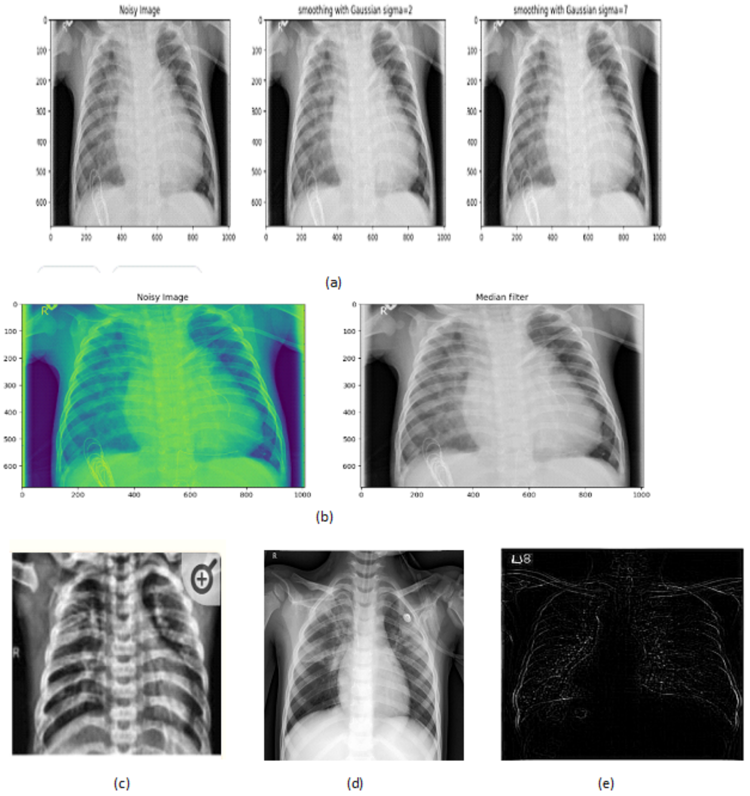

Noisy Image
smoothing with Gaussian sigma=2
smoothing with Gaussian sigma=7
(a)
Noisy Image
Median filter
(b)
(c)
(d)
(e)

**Fig 2. a. Gaussian filter b. Median Filter c. Histogram Equalization d. Sharpening image e. Edge detection using Canny Edge Detector**

Some popular augmentations people use are grayscales, horizontal flips, vertical flips, random crops, color jitters, translations, rotations, and much more. By applying just, a couple of these transformations to our training data, we can easily double or triple the number of training examples and create a very robust model.

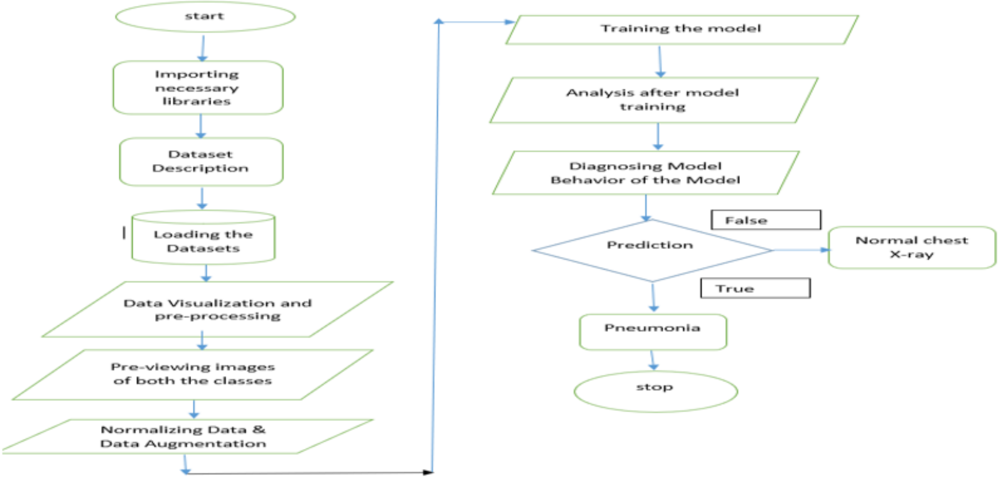

start
Importing necessary libraries
Dataset Description
Loading the Datasets
Data Visualization and pre-processing
Pre-viewing images of both the classes
Normalizing Data & Data Augmentation
Training the model
Analysis after model training
Diagnosing Model Behavior of the Model
Prediction
False
Normal chest X-ray
True
Pneumonia
stop

**Fig 3. Proposed model for pneumonia detection and classification**

Following are the data augmentation techniques used.

**Table 1. Data augmentation techniques settings**

| Data Augmentation | Hyperparameters Values |
|---|---|
| Horizontal split | True |
| Validation split | 0.2 |
| Colour mode | Grayscale |
| Zoom limit | 0.2 |
| Shear limit | 0.2 |
| Rescaling | 1./255 |

1. **Rescale** - also referred to as ”normalizing input.” If every image is scaled to the same range [0,1], the loss will be spread more fairly. Pixels in the range [0,255] will become [0,1] when rescale=1/.255 is applied.

2. **Shearing** - An axis-based visual distortion commonly employed to create or modify perception angles is called shearing. It is typically used to enhance images so that computers may observe objects from various perspectives that are comparable to those of humans.

3. **Zoom limit** - The image is zoomed using the zoom augmentation technique. This technique adds pixels to the image at random locations in order to enlarge or zoom it in. This function makes advantage of the zoom range argument from the image data generator class.

4. **Horizontal split** - The image dimensions stay the same, but its pixels move horizontally with an augmentation of the horizontal shift. It follows that certain picture pixels must be cropped off in order to maintain a consistent image dimension, and new image pixels can be inserted to a replacement zone. To generate new data for the CNN Model, we divide the image horizontally to two in this section.

5. **Color mode** - The black and white color mode sometimes referred to as grayscale color mode, consists of just two colors: black and white. It is employed to replicate images with various grayscale tones.

6. **Validation split** - After first rearranging the data, the model divides it into train and validation sets. The train and validation datasets were pre-established in the first epoch, thus they are used in the subsequent epoch rather than being shuffled and partitioned once more.

7. Training of different CNN models were done (VGG16, VGG19, EfficientNetB1, ResNet50, Customized CNN model).

8. Testing of model and prediction of pneumonia was done with different parameters evaluation like accuracy, precision, recall, sensitivity, AUC curve and Confusion matrix, testing loss.

**The model was built in 5 steps below :**

- Max-pooling, batch-normalization, convolutional layer, and other convolutional blocks were the five convolutional blocks we used.
- We have applied four totally connected layers after using a flattened layer on top of it.
- We have also employed dropouts in between to reduce over-fitting.
- Except the final layer, where the sigmoid was used because this was a binary classification task, the activation function relu was used throughout.
- Adam has been my optimizer, and my loss function is cross-entropy.

$$Z^{[l]} = W^{[l]}.A^{(l-1]} + b^{(l]} \qquad A^{[l]} = g^{[l]}\left(Z^{[l]}\right) \tag{2}$$

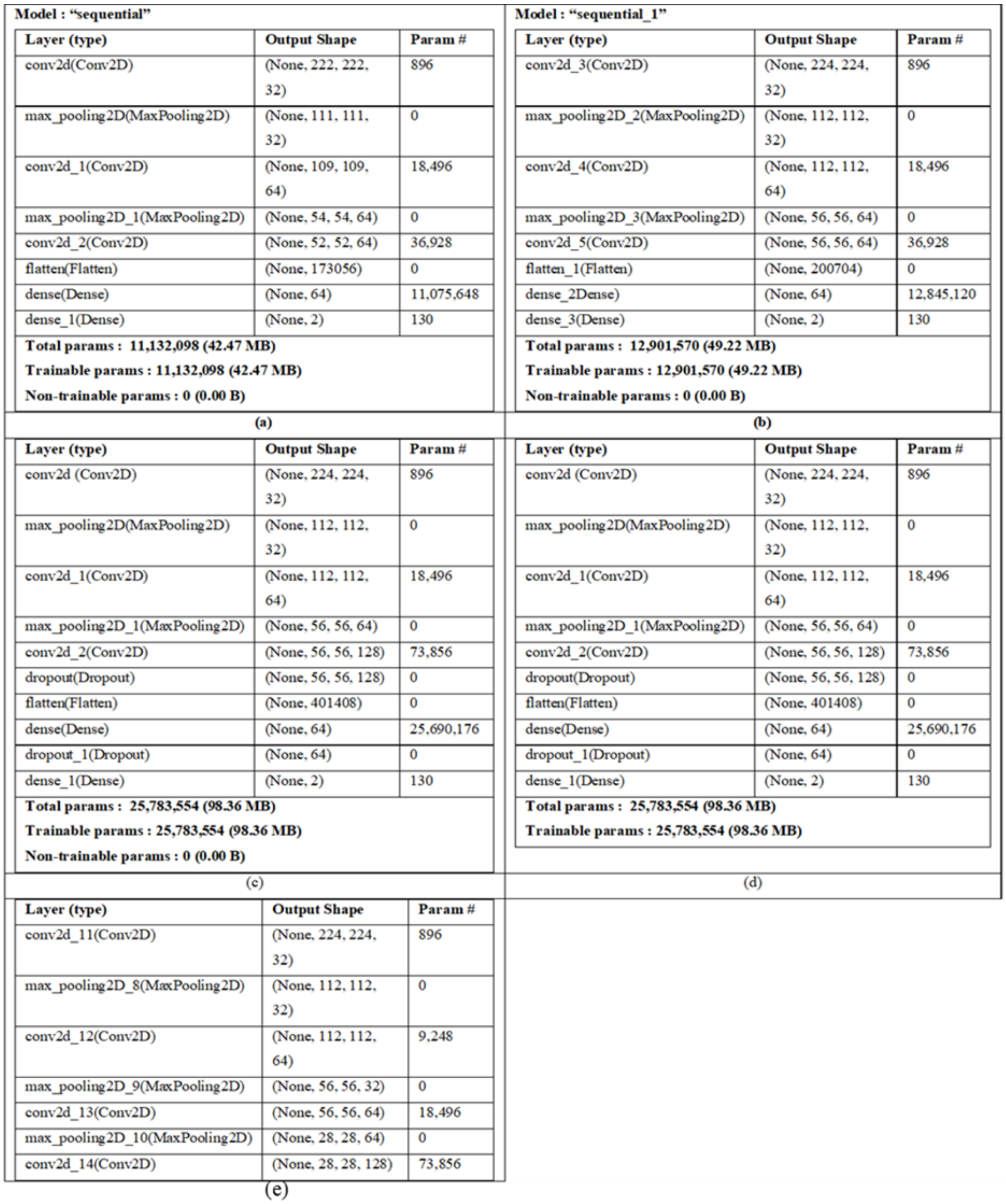
The equation above describes overall working of customized CNN module designed above.

First, calculate the intermediate value Z which is obtained from the result of convolution of input data obtained from a previous layer with tensor W and finally add a bias b to it.

It is helpful to define one or more callbacks prior to training the model. Two that are quite useful are early stopping and model checkpoint.

**Model Checkpoint:** Several iterations are frequently needed when training takes a long time to provide satisfactory results. Here, it is better to save a copy of the best-performing model till the end of an epoch that aids in improving metrics.

**Early Stopping:** Occasionally, we see that the generalization gap that is, the difference between training and validation error begins to increase rather than decrease over training. There are numerous strategies to address this overfitting symptom



Model : “sequential”

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d(Conv2D) | (None, 222, 222, 32) | 896 |
| max_pooling2D(MaxPooling2D) | (None, 111, 111, 32) | 0 |
| conv2d_1(Conv2D) | (None, 109, 109, 64) | 18,496 |
| max_pooling2D_1(MaxPooling2D) | (None, 54, 54, 64) | 0 |
| conv2d_2(Conv2D) | (None, 52, 52, 64) | 36,928 |
| flatten(Flatten) | (None, 173056) | 0 |
| dense(Dense) | (None, 64) | 11,075,648 |
| dense_1(Dense) | (None, 2) | 130 |

**Total params : 11,132,098 (42.47 MB)**
**Trainable params : 11,132,098 (42.47 MB)**
**Non-trainable params : 0 (0.00 B)**

(a)

Model : “sequential_1”

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_3(Conv2D) | (None, 224, 224, 32) | 896 |
| max_pooling2D_2(MaxPooling2D) | (None, 112, 112, 32) | 0 |
| conv2d_4(Conv2D) | (None, 112, 112, 64) | 18,496 |
| max_pooling2D_3(MaxPooling2D) | (None, 56, 56, 64) | 0 |
| conv2d_5(Conv2D) | (None, 56, 56, 64) | 36,928 |
| flatten_1(Flatten) | (None, 200704) | 0 |
| dense_2Dense) | (None, 64) | 12,845,120 |
| dense_3(Dense) | (None, 2) | 130 |

**Total params : 12,901,570 (49.22 MB)**
**Trainable params : 12,901,570 (49.22 MB)**
**Non-trainable params : 0 (0.00 B)**

(b)

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 224, 224, 32) | 896 |
| max_pooling2D(MaxPooling2D) | (None, 112, 112, 32) | 0 |
| conv2d_1(Conv2D) | (None, 112, 112, 64) | 18,496 |
| max_pooling2D_1(MaxPooling2D) | (None, 56, 56, 64) | 0 |
| conv2d_2(Conv2D) | (None, 56, 56, 128) | 73,856 |
| dropout(Dropout) | (None, 56, 56, 128) | 0 |
| flatten(Flatten) | (None, 401408) | 0 |
| dense(Dense) | (None, 64) | 25,690,176 |
| dropout_1(Dropout) | (None, 64) | 0 |
| dense_1(Dense) | (None, 2) | 130 |

**Total params : 25,783,554 (98.36 MB)**
**Trainable params : 25,783,554 (98.36 MB)**
**Non-trainable params : 0 (0.00 B)**

(c)

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 224, 224, 32) | 896 |
| max_pooling2D(MaxPooling2D) | (None, 112, 112, 32) | 0 |
| conv2d_1(Conv2D) | (None, 112, 112, 64) | 18,496 |
| max_pooling2D_1(MaxPooling2D) | (None, 56, 56, 64) | 0 |
| conv2d_2(Conv2D) | (None, 56, 56, 128) | 73,856 |
| dropout(Dropout) | (None, 56, 56, 128) | 0 |
| flatten(Flatten) | (None, 401408) | 0 |
| dense(Dense) | (None, 64) | 25,690,176 |
| dropout_1(Dropout) | (None, 64) | 0 |
| dense_1(Dense) | (None, 2) | 130 |

**Total params : 25,783,554 (98.36 MB)**
**Trainable params : 25,783,554 (98.36 MB)**

(d)

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_11(Conv2D) | (None, 224, 224, 32) | 896 |
| max_pooling2D_8(MaxPooling2D) | (None, 112, 112, 32) | 0 |
| conv2d_12(Conv2D) | (None, 112, 112, 64) | 9,248 |
| max_pooling2D_9(MaxPooling2D) | (None, 56, 56, 32) | 0 |
| conv2d_13(Conv2D) | (None, 56, 56, 64) | 18,496 |
| max_pooling2D_10(MaxPooling2D) | (None, 28, 28, 64) | 0 |
| conv2d_14(Conv2D) | (None, 28, 28, 128) | 73,856 |

(e)

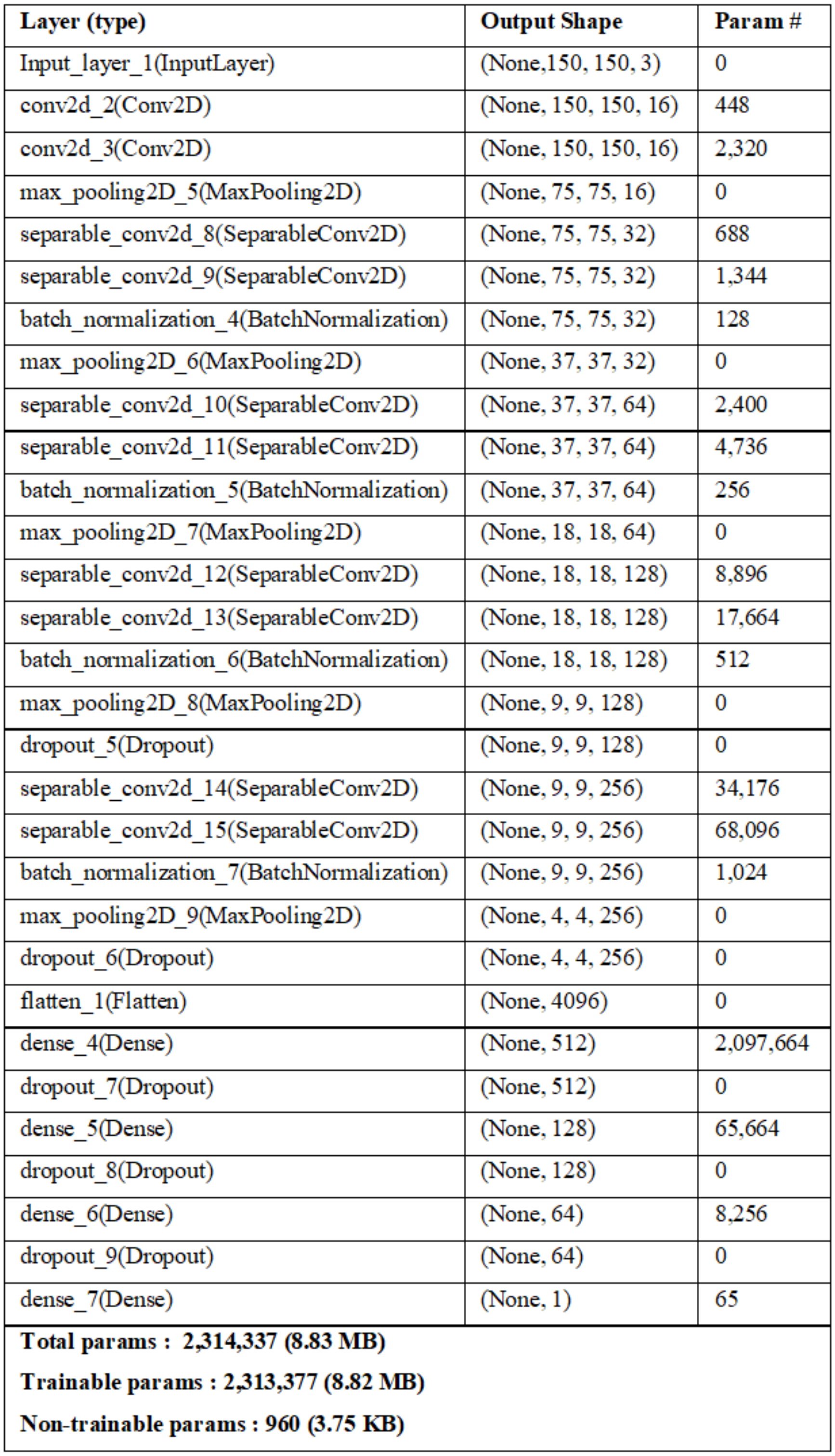
**Fig 4. a. Baseline CNN b. CNN with l2 kernel regularizer c.CNN with dropout layer d.CNN with softmax activation function e.cnn with rmsprop optimizer**

including decreasing model capacity, boosting training data, data augmentation, regularization, dropout, etc. When the generalization gap is reached, stopping training is frequently a useful and effective approach.

The Figure 4 shows different CNNs with an incremental approach of baseline to l2 kernel regularizer, dropout layer, softmax activation function and rmsprop optimizer.[13–15]

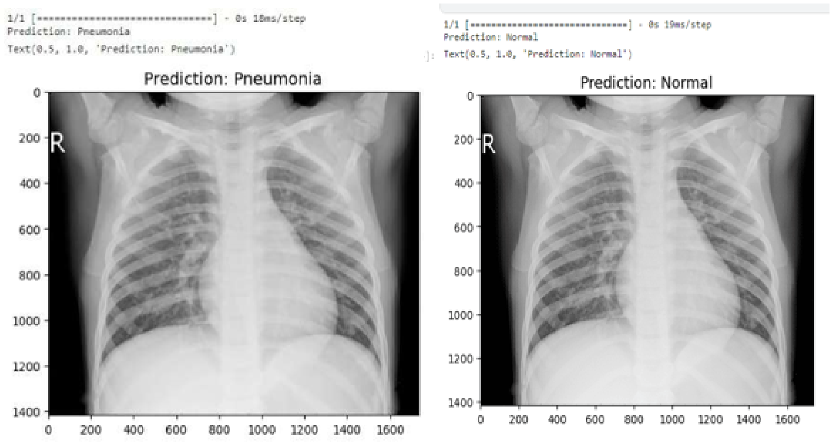
Finally, customized CNN is shown in Figure 5 with 5 convolution blocks, maxpooling, batch-normalization, flattened layer, fully connected layer with dropout layers in the middle of them, adam as optimizer and relu activation function except the last layer as sigmoid with loss of cross-entropy gave best results among previously developed CNNs.

Prediction of pneumonia disease infected regions is shown in the Figure 6. If the probability is <0.6, its pneumonia else a normal image.



| Layer (type) | Output Shape | Param # |
|---|---|---|
| Input_layer_1(InputLayer) | (None,150, 150, 3) | 0 |
| conv2d_2(Conv2D) | (None, 150, 150, 16) | 448 |
| conv2d_3(Conv2D) | (None, 150, 150, 16) | 2,320 |
| max_pooling2D_5(MaxPooling2D) | (None, 75, 75, 16) | 0 |
| separable_conv2d_8(SeparableConv2D) | (None, 75, 75, 32) | 688 |
| separable_conv2d_9(SeparableConv2D) | (None, 75, 75, 32) | 1,344 |
| batch_normalization_4(BatchNormalization) | (None, 75, 75, 32) | 128 |
| max_pooling2D_6(MaxPooling2D) | (None, 37, 37, 32) | 0 |
| separable_conv2d_10(SeparableConv2D) | (None, 37, 37, 64) | 2,400 |
| separable_conv2d_11(SeparableConv2D) | (None, 37, 37, 64) | 4,736 |
| batch_normalization_5(BatchNormalization) | (None, 37, 37, 64) | 256 |
| max_pooling2D_7(MaxPooling2D) | (None, 18, 18, 64) | 0 |
| separable_conv2d_12(SeparableConv2D) | (None, 18, 18, 128) | 8,896 |
| separable_conv2d_13(SeparableConv2D) | (None, 18, 18, 128) | 17,664 |
| batch_normalization_6(BatchNormalization) | (None, 18, 18, 128) | 512 |
| max_pooling2D_8(MaxPooling2D) | (None, 9, 9, 128) | 0 |
| dropout_5(Dropout) | (None, 9, 9, 128) | 0 |
| separable_conv2d_14(SeparableConv2D) | (None, 9, 9, 256) | 34,176 |
| separable_conv2d_15(SeparableConv2D) | (None, 9, 9, 256) | 68,096 |
| batch_normalization_7(BatchNormalization) | (None, 9, 9, 256) | 1,024 |
| max_pooling2D_9(MaxPooling2D) | (None, 4, 4, 256) | 0 |
| dropout_6(Dropout) | (None, 4, 4, 256) | 0 |
| flatten_1(Flatten) | (None, 4096) | 0 |
| dense_4(Dense) | (None, 512) | 2,097,664 |
| dropout_7(Dropout) | (None, 512) | 0 |
| dense_5(Dense) | (None, 128) | 65,664 |
| dropout_8(Dropout) | (None, 128) | 0 |
| dense_6(Dense) | (None, 64) | 8,256 |
| dropout_9(Dropout) | (None, 64) | 0 |
| dense_7(Dense) | (None, 1) | 65 |
| **Total params : 2,314,337 (8.83 MB)**<br>**Trainable params : 2,313,377 (8.82 MB)**<br>**Non-trainable params : 960 (3.75 KB)** | | |

**Fig 5. Customized CNN model Architecture**



```
1/1 [==============================] - 0s 18ms/step
Prediction: Pneumonia
Text(0.5, 1.0, 'Prediction: Pneumonia')
```

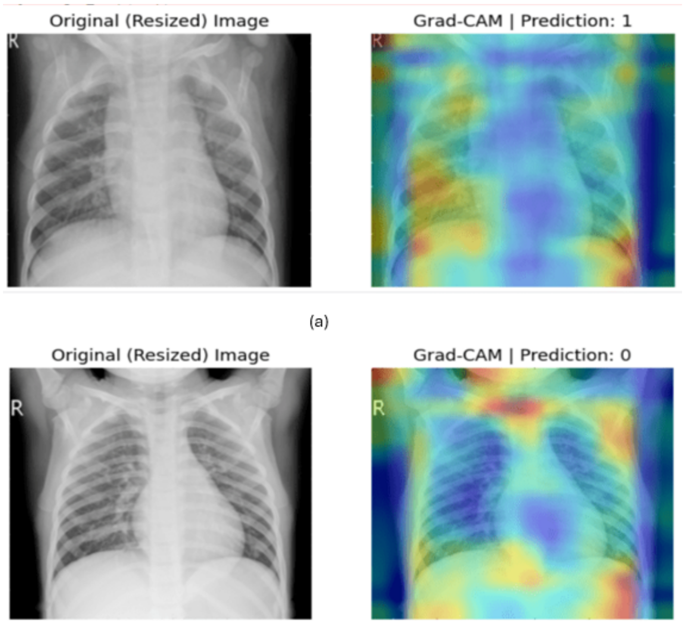
```
1/1 [==============================] - 0s 19ms/step
Prediction: Normal
]: Text(0.5, 1.0, 'Prediction: Normal')
```

(a)

(b)

**Fig 6. a. Prediction of pneumonia b. Prediction of normal image**

The Figure 7 shows pneumonia visualization using Grad CAM. Activation maps can be used to localize disease infected regions[9,11,24].



(a)

(b)

**Fig 7. Visualization of pneumonia using Explainable AI Grad CAM approach**

## 3 Results and Discussion

With an accuracy of 98.84%, the customized CNN outperforms the weighted classifier CNN at 98.43%, the VGG19+CNN at 98.05%, followed by ensemble learning using (GoogleNet, DenseNet121, ResNet-18), CovXNet, ensemble learning (VGG16, ResNet-50, ResNet-101, ResNet152, VGG-19, DenseNet 201, MobileNet-V2), EfficientNetB0 and DenseNet121, DenseNet201, DenseNet121. The Figure 9 shows the confusion matrix for the following: baseline CNN, l2 kernel regularizer, dropout layer, softmax activation function, and rmsprop optimizer.

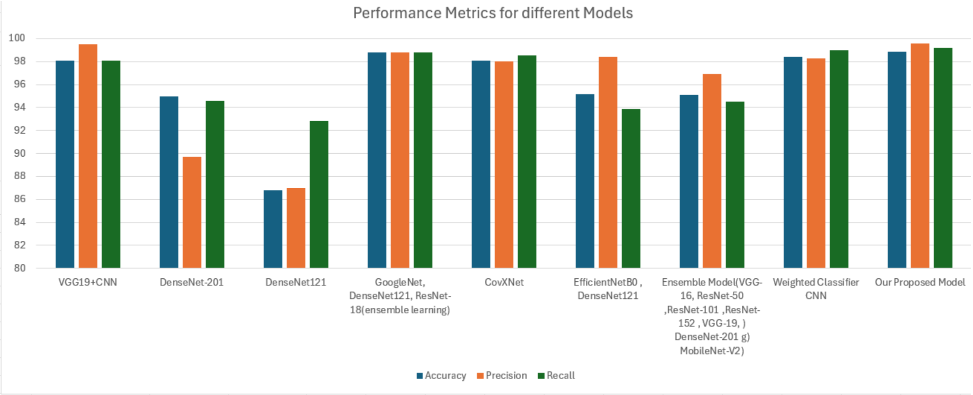


**Fig 8. Comparison of customized CNN with state of art methods**

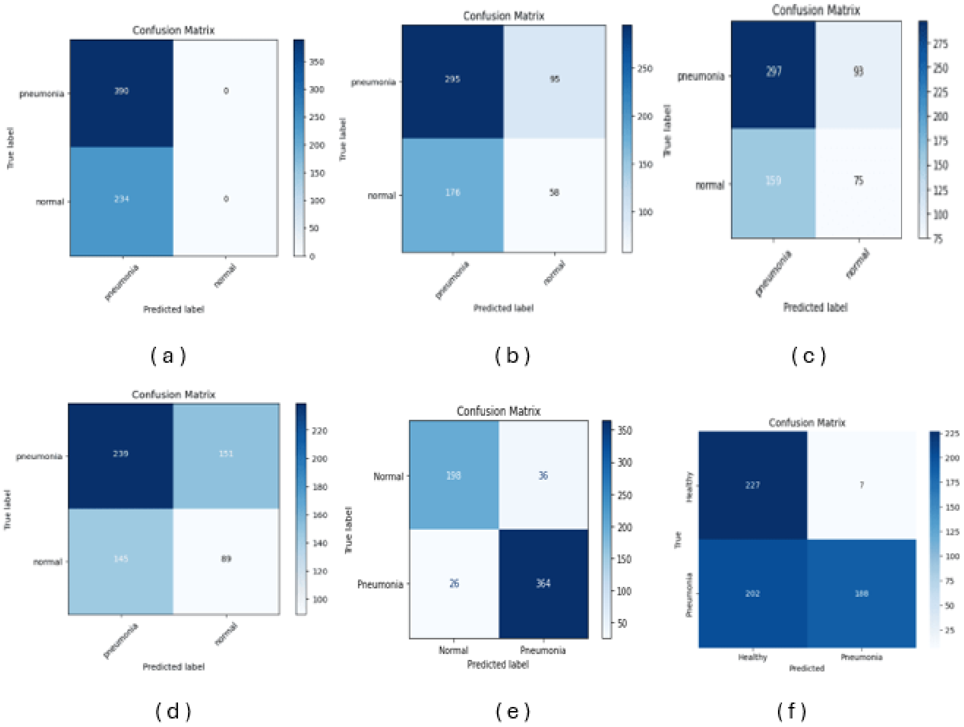


**Fig 9. Confusion matrix for CNN using a. softmax b. l2 regularizer c. rmsprop d. customized cnn e. dropout f. baseline CNN**

The confusion matrix returned identical results for the softmax function and dropout layer for our database containing pneumonia cases. In contrast, the customized CNN showed a good score of 239 for pneumonia cases and also detected 89 normal cases, which was little less. Baseline CNN gave 227 healthy cases and 188 pneumonia cases, which was quite average score for positive pneumonia cases. Normal cases were also detected, which seems to be zero for softmax function and dropout layer.

Customized CNN was used to improve the VGG16, which was developed in[13] and exhibited less accuracy of less than 90%. When compared to other CNN models such as VGG16, EfficientNetB1, VGG19, ResNet50, and InceptionV3 the testing accuracy of Customized CNN is higher. When compared to VGG16, VGG19, EfficientB1, and InceptionV3, ResNet50, customized CNN has the highest recall. Compared to EfficientB1, VGG19, ResNet50, and InceptionV3, VGG16 has greater precision. In comparison to customized CNN, ResNet50, VGG19, EfficeintB1, ResNet50, and InceptionV3, VGG16 has a higher AUC score. Much improved testing accuracy of 93.75%, recall of 98.20%, precision of 92.28%, and f1-score

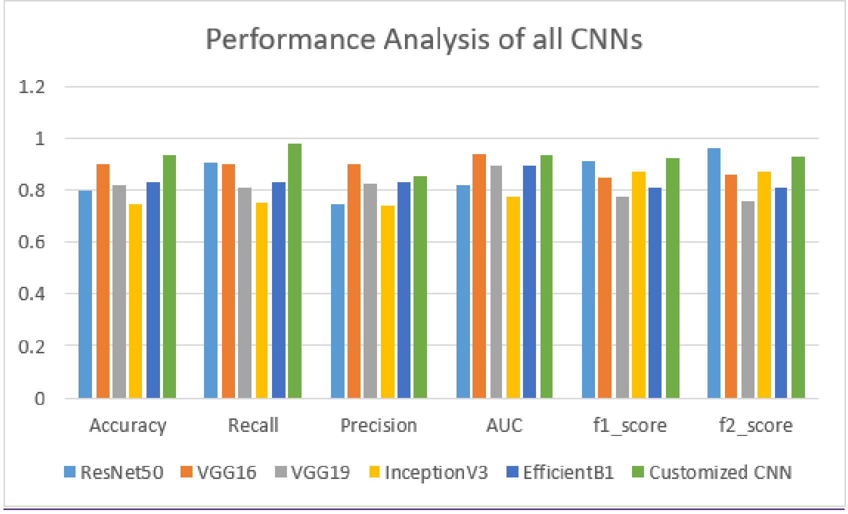


**Fig 10. Performance Analysis of different CNN models**

of 95.15% were obtained with the customized CNN. Baseline CNN was initial CNN model without any modifications or additional regularization techniques gave 82.05% accuracy, replacing the final layer's activation function with a softmax function shows 78.36% accuracy, l2 regularization lead to 66.50% accuracy, introducing dropout layer gave 79.48% accuracy and final customized CNN gave 93.75% accuracy. Customized CNN showed considerable testing results as compared to other pre-trained CNN and test loss was also low, only precision was little lower. Grad CAM was used for an explainable AI-based technique used with customized CNN to show pneumonia in affected regions.

## 4 Conclusion

The study presents a customized CNN approach for detection of pneumonia from chest X-ray images, having much better accuracy of 93.75%, recall of 98.20%, precision of 92.2%, f1_score of 95.15% as compared to other types of pre-trained CNNs. Recall obtained in customized CNN is good with 98.20% as compared to ResNet with 90.54%, VGG16 with 89.90%, VGG19 with 81.08%, EfficientB1 with 83.17 and lastly Inceptionv3 with 75.32%. In the future, researchers can do testing using more improvised CNN and various deep learning techniques. It is also possible to determine the proportion of the affected area or region of Chest from the area where the disease is present. To increase the accuracy of customized CNN testing, researchers can add more customized layers. Future work will include utilizing various data balancing strategies, increasing the amount of data to ensure that CNN is well-trained, and including real-time hospital datasets. In addition to Grad CAM utilized with a tailored CNN model, other explainable AI techniques can also be applied. A few drawbacks of our research include the following: the region of interest (left and right lung) is not extracted separately and can be extracted using various segmentation techniques, more advanced pre-processing techniques can be used for more efficient pre-processing, and the severity of pulmonary diseases can be identified to enable more effective patient treatment. Certain quantitative EX-AI techniques can make saliency map comprehension easier. Automatic report production can also be incorporated into the system to help patients and radiologists begin patient care precisely and on time.

## References

1) Ibrahim DM, Elshennawy NM, Sarhan AM. Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. *Computers in Biology and Medicine*. 2021;132:1–13. Available from: https://doi.org/10.1016/j.compbiomed.2021.104348.
2) Wahid F, Azhar S, Ali S, Zia MS, Almisned FA, Gumaei A. Pneumonia Detection in Chest X-Ray Images Using Enhanced Restricted Boltzmann Machine. *Journal of Healthcare Engineering*. 2022;2022:1–17. Available from: https://doi.org/10.1155/2022/1678000.
3) Quan H, Xu X, Zheng T, Li Z, Zhao M, Cui X. DenseCapsNet: Detection of COVID-19 from X-ray images using a capsule neural network. *Computers in Biology and Medicine*. 2021;133:1–11. Available from: https://doi.org/10.1016/j.compbiomed.2021.104399.
4) Alhudhaif A, Polat K, Karaman O. Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. *Expert Systems with Applications*. 2021;180:1–9. Available from: https://doi.org/10.1016/j.eswa.2021.115141115141.
5) Singh G, Yow KC. An Interpretable Deep Learning Model for Covid-19 Detection with Chest X-Ray Images. *IEEE Access*. 2021;9:85198–85208. Available from: https://doi.org/10.1109/ACCESS.2021.3087583.

6) Horry MJ, Chakraborty S, Paul M, Ulhaq A, Pradhan B, Saha M, et al. COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data. *IEEE Access*. 2020;8:149808–149824. Available from: https://doi.org/10.1109/access.2020.3016780.

7) Wang J, Bao YM, Wen YF, Lu HB, Luo H, Xiang YF, et al. Prior-Attention Residual Learning for More Discriminative COVID-19 Screening in CT Images. *IEEE Transactions on Medical Imaging*. 2020;39(8):2572–2583. Available from: https://doi.org/10.1109/TMI.2020.2994908.

8) Wang Z, Xiao Y, Li Y, Zhang J, Lu F, Hou M, et al. Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays. *Pattern Recognition*. 2021;110:1–9. Available from: https://doi.org/10.1016/j.patcog.2020.107613.

9) Brunese L, Mercaldo F, Reginelli A, Santone A. Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays. *Computer Methods and Programs in Biomedicine*. 2020;196:1–11. Available from: https://doi.org/10.1016/j.cmpb.2020.105608.

10) Suryaa VS, R AXA, Aiswarya MS. Efficient DNN Ensemble for Pneumonia Detection in Chest X-ray Images. *International Journal of Advanced Computer Science and Applications*. 2021;12(10):759–767. Available from: https://thesai.org/Downloads/Volume12No10/Paper_84-Efficient_DNN_Ensemble_for_Pneumonia_Detection.pdf.

11) Hashmi FM, Katiyar S, Keskar AG, Bokde DN, Geem ZW. Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning. *Diagnostics* . 2020;10(6):1–23. Available from: https://doi.org/10.3390/diagnostics10060417.

12) Yadav P, Menon N, Ravi V, Vishvanathan S. Unsupervised Representation Learning for Lung Disease Classification using Chest CT and X-ray Images. *IEEE Transactions on Engineering Management* . 2021;70(8):2774 –2786. Available from: https://doi.org/10.1109/TEM.2021.3103334.

13) Jiang Z, Liu Y, Shao Z, Huang K. An Improved VGG16 Model for Pneumonia Image Classification. *Applied Science*. 2021;p. 1–19. Available from: https://doi.org/10.3390/app112311185.

14) Mabrouk A, Redondo RPD, Dahou A, Elaziz MA, Kayed M. Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. *Applied Sciences*. 2022;12(13):1–10. Available from: https://doi.org/10.3390/app12136448.

15) Yao S, Chen Y, Tian X, Jiang R. Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN. *Computational and Mathematical Methods in Medicine*. 2021;2021:1–13. Available from: https://doi.org/10.1155/2021/8854892.

16) Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ. Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurement*. 2020;165. Available from: https://doi.org/10.1016/j.measurement.2020.108046.

17) Cha SM, Lee SS, Ko B. Attention-Based Transfer Learning for Efficient Pneumonia Detection in Chest X-ray Images. *Applied Sciences*. 2021;11(3):1–15. Available from: https://dx.doi.org/10.3390/app11031242.

18) Wu L, Zhang J, Wang Y, Ding R, Cao Y, Liu G, et al. Pneumonia detection based on RSNA dataset and anchor-free deep learning detector. *Scientific Reports*. 2024;14(1):1–8. Available from: https://dx.doi.org/10.1038/s41598-024-52156-7.

19) Guo K, Cheng J, Li K, Wang L, Lv Y, Cao D. Diagnosis and detection of pneumonia using weak-label based on X-ray images:a multi-center study. *BMC Medical Imaging*. 2023;23:1–8. Available from: https://doi.org/10.1186/s12880-023-01174-4.

20) Almaslukh B. A Lightweight Deep Learning-Based Pneumonia Detection Approach for Energy-Efficient Medical Systems. *Wireless Communications and Mobile Computing*. 2021;2021:1–14. Available from: https://dx.doi.org/10.1155/2021/5556635.

21) Salehi M, Mohammadi R, Ghaffari H, Sadighi N, Reiazi R. Automated detection of pneumonia cases using deep transfer learning with paediatric chest X-ray images. *The British Journal of Radiology*. 2021;94(1121). Available from: https://dx.doi.org/10.1259/bjr.20201263.

22) Mahmud T, Rahman MA, Fattah SA. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in Biology and Medicine*. 2020;122:1–10. Available from: https://dx.doi.org/10.1016/j.compbiomed.2020.103869.

23) Kundu R, Das R, Geem ZW, Han GT, Sarkar R. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLOS ONE*. 2021;16(9):1–29. Available from: https://dx.doi.org/10.1371/journal.pone.0256630.

24) Hanif A, Zhang X, Wood S. A Survey on Explainable Artificial Intelligence Techniques and Challenges. In: 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW). IEEE. 2021. Available from: https://doi.org/10.1109/EDOCW52865.2021.00036.