Pneumonia Detection in Chest X-Rays

D Veera Harsha Vardhan Reddy, V Sai Sumanth

Computer Science Department, BML Munjal University, Gurgaon, Haryana, India.

Abstract

Background and Objectives: Chest X-rays provide an effective method to diagnose lung infections in cases of pneumonia. Pneumonia presents a significant global health challenge, which requires accurate and efficient methods for diagnosing it. Computer-aided systems have shown promise in aiding radiologists for initial screening, utilizing convolutional neural networks (CNNs) and transfer learning techniques from pre-trained models like ResNet152V2 and VGG-16. The goal of this project is to use these approaches to accurately classify pneumonia cases. **Material & Methods:** This study explores the classification of chest X-ray images for pneumonia detection. Three different architectures are used and optimised for optimal performance: CNN and transfer learning with pre-trained models ResNet152V2 and VGG-16. The models are trained and assessed using pre-processed dataset from publicly accessible source, allowing for a comparative study of their effectiveness. **Results:** Each model's performance is evaluated using measures like F-score, accuracy, precision, and recall. When it comes to pneumonia classification, the CNN, ResNet152V2, and VGG-16 models show excellent F-scores, accuracy, and recall. Additionally, the comparison study demonstrates the better performance of a particular model, which is chosen for additional development. The best model was identified and utilized as the foundation for a website made using Streamlit that detects pneumonia and offers a user-friendly interface for quick and easy diagnosis.

Keywords: CNN, ResNet152V2, VGG-16, Streamlit, Chest X-rays, Pneumonia, Transfer Learning.

.

1 Introduction

An easily accessible method of assessing lung health and providing important information about respiratory disorders is the human chest X-ray. Key features from chest X-ray images may be retrieved by using image processing techniques. These features include the identification of infiltrates, consolidations, and opaque zones, which are important indications for illness diagnosis [1]. In the field of respiratory health, the characteristics derived from chest X-rays are essential diagnostic components that facilitate the identification and categorization of illnesses such as pneumonia.

In this study, we specifically utilise Deep learning approaches to pneumonia identification using chest X-rays, building on the effective use of convolutional neural networks (CNNs) in image processing. We propose the implementation of three distinctive models: a CNN model, transfer learning utilizing ResNet152V2[2], and a pre-trained VGG-16 model [1], to compare the results they produce in pneumonia classification on pre-processed dataset of a health centre children Chest X-Rays.

1. Literature Review: Reviewing previous research on pneumonia classification done using deep learning methods and chest X-ray images.

2. Methodology: Exploring the pre-processed dataset and preprocessing it by implementing data augmentation techniques so using the generated dataset Introducing and implementing CNN, ResNet152V2 transfer learning and fine tuning it, and VGG-16 pre-trained models

3. Comparison of Models: Evaluating and comparing the performance of these models based on metrics such as Accuracy, Precision, Recall, and F-score to determine the most effective approach for pneumonia detection.

The structure of this paper is organized as follows: Section 2 provides a comprehensive literature review on pneumonia detection methodologies using deep learning techniques with chest X-rays. Section 3 outlines the methodology adopted for this study and details the dataset utilized. In Section 4, we present the results obtained from the three models and engage in a discussion of their performance. Finally, in Section 5, conclusions are drawn based on the findings, emphasizing the most effective model for pneumonia detection.

2 Related Work

In recent years, there has been a lot of interest in the use of chest X-ray imaging to identify pneumonia. An automatic detection method utilizing transfer learning and CNNs more precisely, the VGG16 model was presented by Thakur et al. (2020). Their study, which obtained 90.54% accuracy, 98.71% recall, and 87.69% precision, demonstrates the effectiveness of CNN models in the diagnosis of pneumonia [6].

In order to identify pneumonia, researchers [5] presented a study that focused on the feature extraction and classification of chest X-ray images. They evaluated the effectiveness of several CNN architectures in pneumonia identification by extracting characteristics from images. To determine how data size affected CNN performance, they conducted dataset augmentation research. Furthermore, they also explored the use of CNN architectures for chest X-ray image-based pneumonia identification [1]. Given the seriousness of pneumonia, their research highlighted the need for immediate recognition and focused on creating CNN models that can extract characteristics and classify images for an accurate diagnosis [1].

Although earlier research has demonstrated the effectiveness of CNN-based models in the identification of pneumonia, these investigations greatly advance our knowledge of the effects of different CNN architectures, dataset sizes, and augmentation strategies on the precision and efficiency of pneumonia identification using chest X-ray images.

These research demonstrate that using Convolutional Neural Networks (CNNs), particularly architectures such as VGG16, is very successful in detecting pneumonia from chest X-ray images. The study demonstrated remarkable accuracy rates, suggesting the importance of CNNs and VGG 16s in accurately recognising pneumonia. Furthermore, investigating different CNN architectures, dataset sizes, and augmentation methods offers insightful information on how to improve these models' performance for accurate and effective pneumonia identification from chest X-ray images. All these results contribute to our growing understanding of the critical role CNN and VGG 16 models play in improving and automating the diagnosis of pneumonia using chest X-ray imaging.

3 Methodology

3.1 Dataset

There are two subfolders (train and test) for each image category (Pneumonia/Normal) in the well-organized dataset. There are 2 categories (Pneumonia/Normal) combining a total 5,863 X-Ray images in JPEG format. Anterior-posterior chest X-ray pictures were chosen from retrospective cohorts of paediatric patients from Guangzhou Women and Children's Medical Centre, Guangzhou, aged one to five. Every chest X-ray image was taken as a standard clinical procedure for the patients.

All chest radiographs were first screened for quality control by eliminating any low quality or unreadable scans before being subjected to the analysis of chest x-ray pictures. Before the photos' diagnosis could be used to train the AI systems,they were evaluated by two board-certified medical professionals. A third expert verified the assessment set to make sure there were no grading problems.

3.2 Pre-Processing

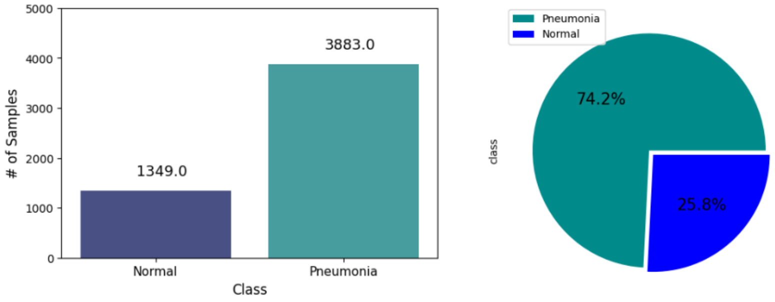


Fig: Graphical representation of Train dataset

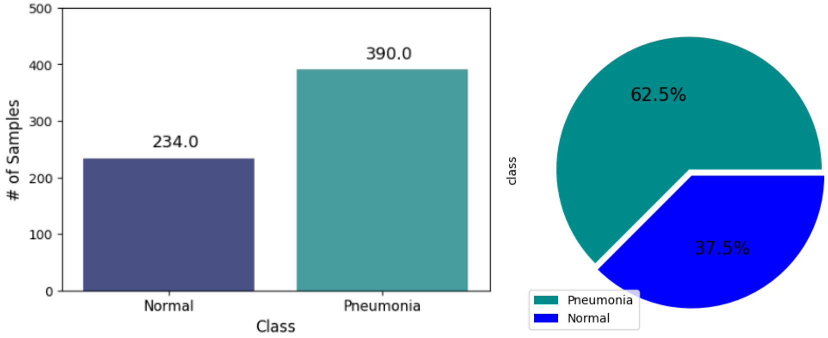


Fig: Graphical representation Test Dataset

The initial phase involved creating a validation set through a simple stratified split of the original training dataset. This division allocated 80% for actual training and 20% for validation purposes, ensuring a representative distribution across both sets. Both datasets exhibited slight imbalances, with a higher representation of samples from the positive class (Pneumonia), particularly noticeable in the training set.
To prepare the image data for model intake, Keras' Image Data Generator was implemented. With the help of this tool, it was easier to rescale pixel values and apply random transformation techniques for data augmentation in real time. There were two different generators defined:

1. Train Data Generator: This generator included transformations with the goal of augmenting the training set, such as rescaling, zooming, and shifting of width and height.

2. Validation Data Generator: This generator was primarily used for the test sets and validation, and it was only concerned with rescaling pixel values.
These generators were then used using the 'flow_from_dataframe' approach on each dataset. This process involved specifying parameters like target size for resizing images, batch size, and class mode for binary classification. Additionally, rescaling pixel values and guaranteeing consistent target sizes were important steps in getting the image datasets ready for model validation and training.

3.3 Model Architecture

We propose four different architectures: Convolutional Neural Networks (CNN), Transfer Learning using Resnet152V2 and fine tuning it and pre-trained VGG-16 to automatically classify the images of pneumonia Chest X-ray from normal ones.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
conv2d (Conv2D)	(None, 222, 222, 16)	448
batch_normalization (BatchNormalization)	(None, 222, 222, 16)	64
activation (Activation)	(None, 222, 222, 16)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
dropout (Dropout)	(None, 111, 111, 16)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	4640
batch_normalization_1 (BatchNormalization)	(None, 109, 109, 32)	128
activation_1 (Activation)	(None, 109, 109, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
dropout_1 (Dropout)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	18496
conv2d_3 (Conv2D)	(None, 50, 50, 64)	36928
batch_normalization_2 (BatchNormalization)	(None, 50, 50, 64)	256
activation_2 (Activation)	(None, 50, 50, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 25, 25, 64)	0
dropout_2 (Dropout)	(None, 25, 25, 64)	0
flatten (Flatten)	(None, 40000)	0
dense (Dense)	(None, 64)	2560064
dropout_3 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 2621089 (10.00 MB)
Trainable params: 2620865 (10.00 MB)
Non-trainable params: 224 (896.00 Byte)

Fig. 1: Figure showing the detailed architecture of proposed CNN model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 resnet152v2 (Functional)    (None, 7, 7, 2048)        58331648

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 128)               262272

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 58594049 (223.52 MB)
Trainable params: 262401 (1.00 MB)
Non-trainable params: 58331648 (222.52 MB)
_________________________________________________________________
```

Fig. 2: Figure showing the detailed architecture of Resnet152V2 model.

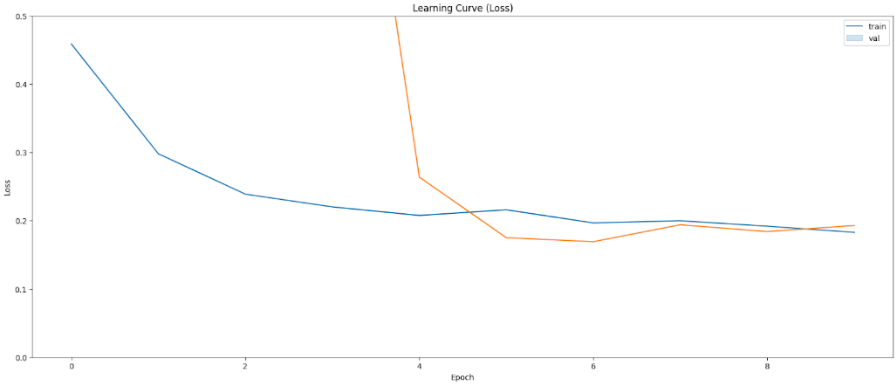
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 resnet152v2 (Functional)    (None, 7, 7, 2048)        58331648

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 128)               262272

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 58594049 (223.52 MB)
Trainable params: 4731137 (18.05 MB)
Non-trainable params: 53862912 (205.47 MB)
_________________________________________________________________
```

Fig. 3: Figure showing change in trainable and non-trainable parameters after fine tuning Resnet152V2.

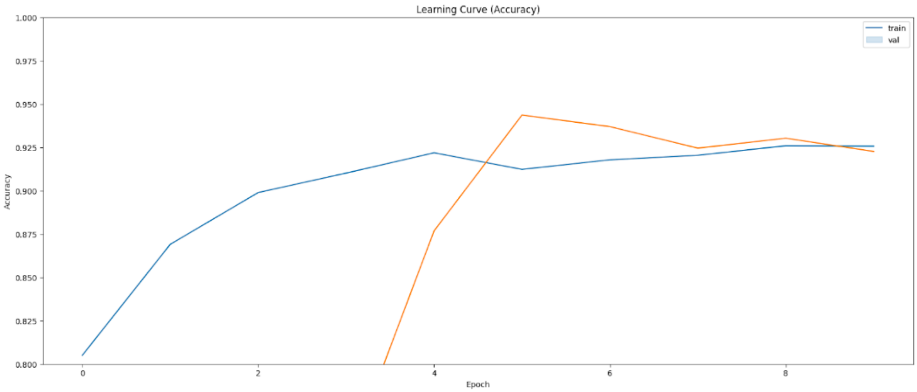
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense_2 (Dense)             (None, 1)                 25089

=================================================================
Total params: 14739777 (56.23 MB)
Trainable params: 25089 (98.00 KB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

Fig. 4: Figure showing the detailed architecture of pre-trained VGG-16 model.

CNN:



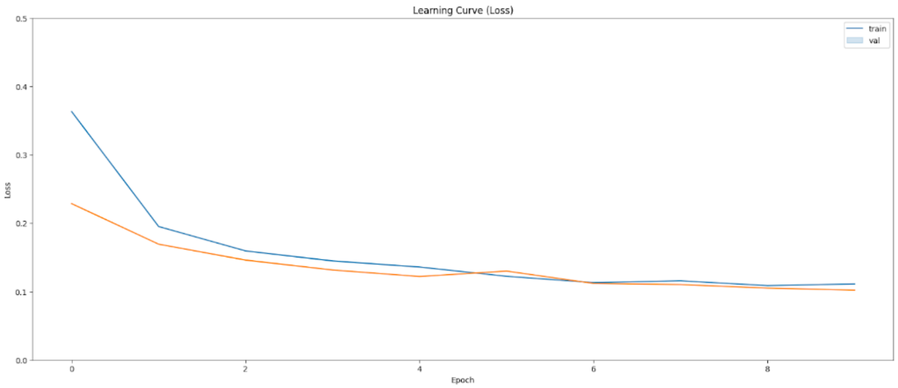
(a)



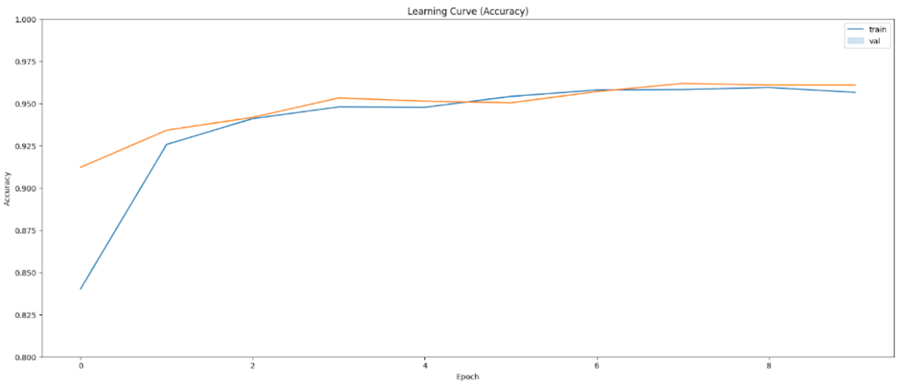
(b)

Fig. 5: Figure (a and b) showing the CNN Model loss and Accuracy.

Resnet152V2:



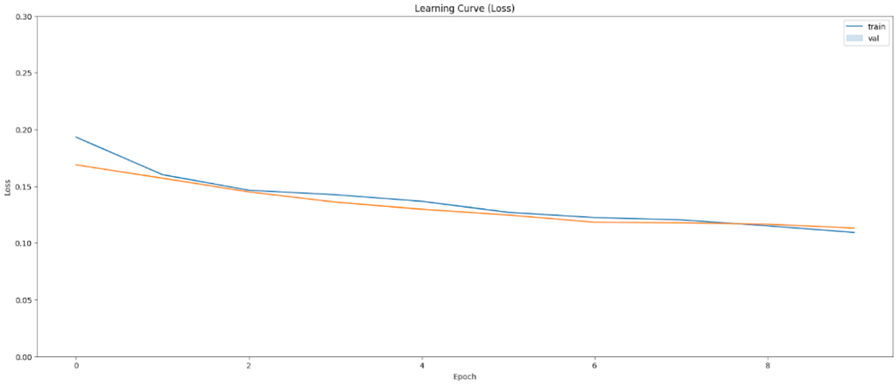
(c)



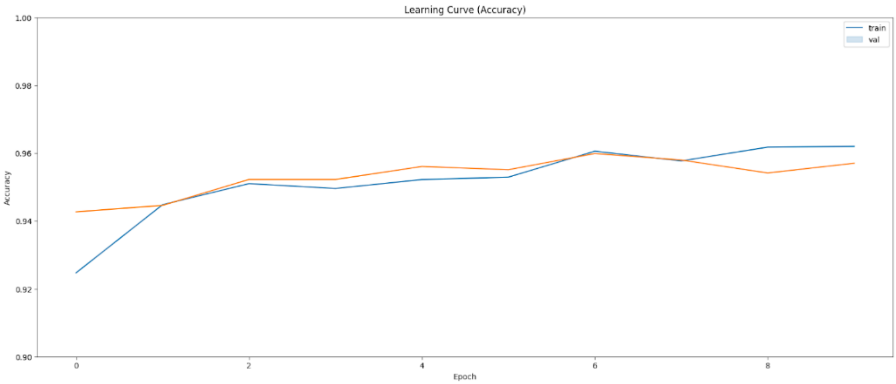
(d)

Fig. 6: Figure (c and d) showing the transfer learned Resnet152V2 Model loss and Accuracy.

Resnet152V2 fine-tuned.



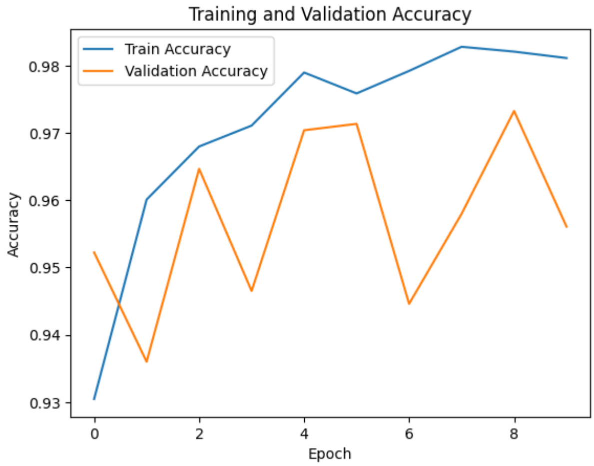
(e)



(f)

Fig. 7: Figure (e and f) showing the transfer learned Resnet152V2 Model After fine tuning loss and Accuracy.

VGG-16:



(g)

(h)

Fig.8: Figure (g and h) showing the pre-trained Model loss and Accuracy.

CNN:

Figure 1 shows the architecture stating that it consists of many convolutional layers, max-pooling, dropout, and finally dense layers for classification.

Input Layer:

The model begins with an input layer defining the shape of the input images (IMG_SIZE x IMG_SIZE) with 3 colour channels.

Blocks:

a) Block One: It consists of a convolutional layer with 16 filters, followed by batch normalization, ReLU activation, max-pooling, and dropout (20%).

b) Block Two: Similar to Block One, it has a convolutional layer with 32 filters, batch normalization, ReLU activation, max-pooling, and dropout (20%).

c)Block Three: This block includes two convolutional layers with 64 filters each, batch normalization, ReLU activation, max-pooling, and dropout (40%).

Head:

The head of the network consists of a flatten layer followed by a dense layer with 64 units and ReLU activation, coupled with dropout (50%).

Output Layer:

Finally, a dense layer with a single unit and sigmoid activation serves as the output layer for binary classification.

Model Compilation and Training:

- The model is compiled using the binary cross-entropy loss function and the Adam optimizer with a specific learning rate.
- `EarlyStopping` and `ReduceLROnPlateau` callbacks are employed for preventing overfitting and optimizing learning rate, respectively.
- The ‘fit’ function trains the model on the training dataset (‘ds_train’) for a specified number of epochs while validating on the validation dataset (‘ds_val’).

The CNN model constructed follows a sequential arrangement of convolutional, pooling, dropout, and dense layers, designed to learn hierarchical representations from chest X-ray images for accurate pneumonia detection. The utilization of dropout layers aids in preventing overfitting, while batch normalization helps in stabilizing and accelerating the training process.

The Custom CNN model architecture performed well on the training and validation sets. However, there seems to be some level of overfitting, as evidenced by a drop in performance on the unseen test data. The accuracy is relatively lower on the test set compared to validation, indicating a potential issue with generalization.

Transfer Learning using ResNet152V2:

The second approach called transfer learning using the model ResNet152V2 architecture shown in figure 2, consists of using a pretrained model as a feature extractor.

This model was already trained in another dataset (ImageNet). The loaded ResNet152V2 base model's layers are set to non-trainable. This step freezes the weights of the pre-trained layers, preventing them from being updated during subsequent training.

Taking advantage of its pre-trained weights and architecture to capture high-level features from chest X-ray images. By freezing the pre-trained layers and adding custom dense layers for classification, the model aims to extract meaningful representations from the images for pneumonia detection.

Fine Tuning the ResNet152V2:

Fine-tuning involves unfreezing some layers of a pre-trained model and retraining them along with the newly added classification layers. This allows the model to adapt and learn more specific features from the new dataset while retaining the learned representations from the initial pre-training on ImageNet.

1. Adaptation to Specific Data: Initially, the pre-trained model's weights were frozen to utilize its general feature extraction capabilities. However, fine-tuning allows to adjust and learn more specific and nuanced features from the pneumonia-related chest X-ray images, potentially improving its performance on this task.

2. Improved Representations: By enabling training on deeper layers while keeping some earlier layers frozen, the model might capture more intricate patterns relevant to pneumonia detection, enhancing its discriminatory power.

Fine-tuning ResNet152V2 on the pneumonia dataset allowed the model to specialize in detecting pneumonia-related features. The increased performance on the test set indicates successful transfer learning.

Fine-tuning is a crucial step in transfer learning, allowing the model to leverage both general knowledge from pre-training and domain-specific knowledge from the new dataset, potentially leading to better performance on the pneumonia detection task compared to using only the pre-trained model or training from scratch.

VGG-16:

VGG-16 is pre-trained on the large and diverse ImageNet dataset, allowing it to learn generic features that could be useful for pneumonia Chest X-Ray Image classification tasks. Fig: 4 shows the architecture of pretrained VGG-16. The convolutional layers of VGG-16 serve as effective feature extractors. By using transfer learning and freezing the weights, the model benefits from these learned features without losing them during training. The VGG-16 layers are frozen enabling the model to adapt to the features of the pneumonia chest X-ray dataset required.

The combination of these factors such as augmentation, architecture and feature extraction contribute to the potential effectiveness of the VGG-16 model in achieving high accuracy and low loss for pneumonia detection on chest X-ray images (shown in Table-1).

User Interface Implementation:

The developed website serves as an efficient framework for pneumonia detection from chest X-rays, simply combining the best model, accessible through Streamlit's user-friendly interface.

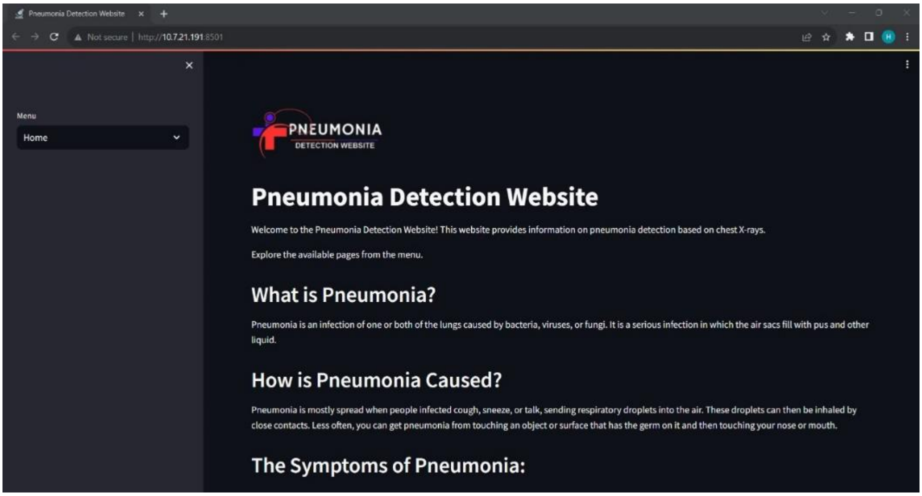


Fig: Image of home page of Website

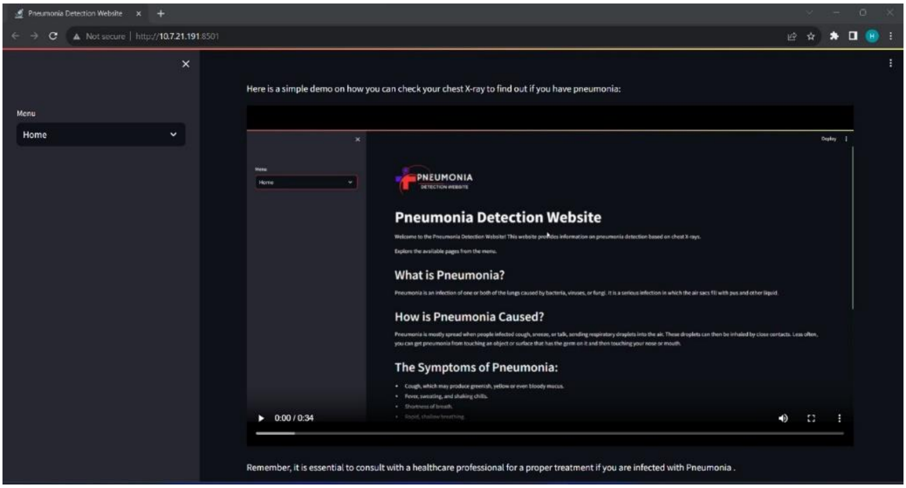


Fig: Demo Video on How to check the Pneumonia by placing the chest X-Ray

Users are met with informative material on the homepage that explains the causes of pneumonia, its consequences, and key information regarding the illness. A special section guides users through the site's execution and provides guidance on how to use it. This comes along with an instructional video that provides a visual guide and ensures users can browse and make easy use of the site's functions.

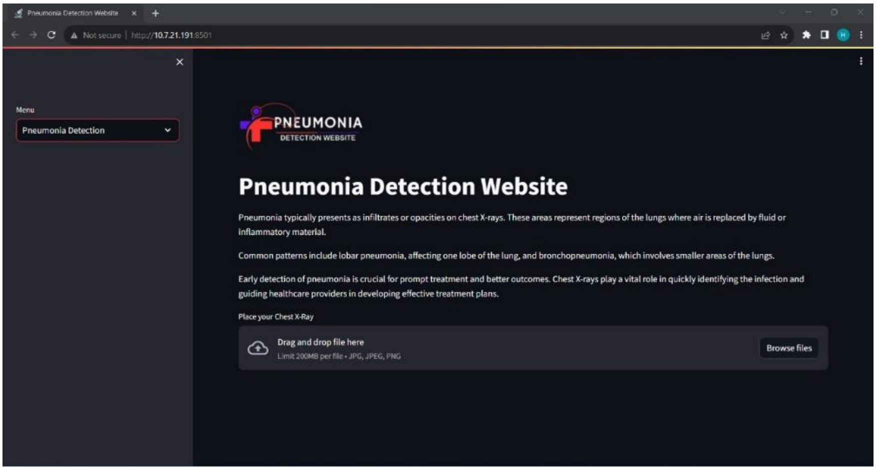


Fig: Webpage which detect pneumonia

Fundamentally, the website includes an advanced deep learning model that has been properly selected and stored as an H5 file. This model can correctly classify images from chest X- ray for pneumonia. All one has to do is drag and drop the X-ray images into the website, and the model will quickly evaluate the provided data and perform an accurate classification. Users are provided with rapid and precise diagnostic insights.

Fig: Prediction of the given X-ray

The above figure shows the prediction of user given chest x-ray as Normal by processing it through the trained model.

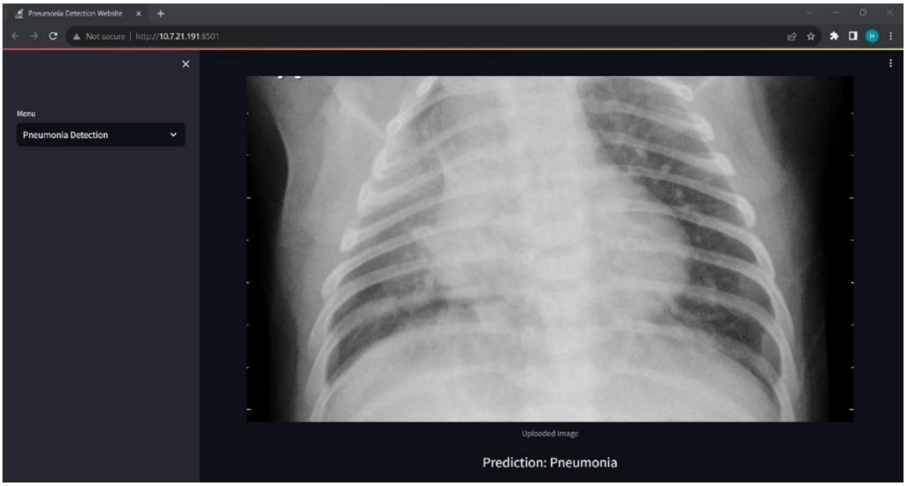


Fig: Prediction of given x-ray

The above figure shows the prediction of user given chest x-ray as Pneumonia by processing it through the trained model.

4 Results

The performance of the classification of pneumonia images is tested through multiple evaluation metrics such as test and train accuracy, precision, recall and F-score to assess the performance and the accuracy of the model

	CNN		Resnet152V2		Fine-tuned Resnet152V2		VGG-16	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Train	0.93	0.18	0.95	0.11	0.96	0.11	0.98	0.05
Val	0.92	0.19	0.96	0.10	0.96	0.11	0.96	0.10
Test	0.89	0.23	0.87	0.29	0.91	0.23	0.95	0.04

Table 1: Accuracy and loss for CNN, Resnet, fine-tuned Resnet and VGG16 model for binary classification.

Precision, recall, and F1 score are crucial metrics in evaluating classification models, providing a deeper understanding of their performance beyond accuracy.

Precision:

Precision measures the accuracy of positive predictions made by the model.

$$\text{Precision} = \text{true positive} / (\text{true positive} + \text{false positive})$$

It calculates the ratio of correctly predicted positive observations to the total predicted positive observations. A higher precision means fewer false positives. As high precision ensures that the model minimizes the chances of wrongly classifying a patient as positive for a disease when they don't have.

Recall:

Recall measures the ratio of correctly predicted positive observations to all actual positives in the data.

$$\text{Recall} = \text{true positive} / (\text{true positive} + \text{false negative})$$

It calculates the proportion of actual positives that were correctly identified by the model. Higher recall means fewer false negatives. As high recall ensures that the model identifies as many actual positive cases as possible, even if it means some false alarms (lower precision).

F1 Score:

The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$F1 := 2/\left(1/\text{precision} + 1/\text{recall}\right).$$

It considers both precision and recall, giving a single metric that reflects the model's overall performance. It's a useful metric when there's an uneven class distribution. As It's helpful when there's an imbalance in the classes or when aiming for a balance between false positives and false negatives.

	CNN			Transfer Learning & fine-tuned Model			VGG-16		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
0	0.90	0.82	0.86	0.95	0.80	0.87	0.98	0.83	0.92
1	0.90	0.95	0.92	0.89	0.97	0.93	0.93	0.99	0.96

Table 2: Precision, Recall and F1-score for CNN, fine-tuned Resnet and VGG16 model
(0 and 1 representing Normal and pneumonia)

5.Conclusion

The evaluation of multiple classification models CNN, ResNet152V2, pretrained and fine-tuned ResNet152V2, and pretrained VGG-16 revealed distinct performances in identifying pneumonia from chest X-ray images. Each model's accuracy and loss metrics demonstrated varying levels of predictive capability on different datasets: training, validation, and testing.

The study shows the significance it is to make use of deep learning architectures to detect pneumonia in chest X-rays. While CNN serves as a solid baseline, transfer learning models fine-tuned ResNet152V2 and VGG-16 stand out for their balanced precision and recall, ensuring minimal misclassification of pneumonia cases. The results support the use of transfer learning techniques to improve pneumonia detection accuracy and reliability, which may lead to more precise diagnostic devices for use in health care institutions.

References

[1]
S. Sharma and K. Guleria, "A Deep Learning based model for the Detection of Pneumonia from Chest X-Ray Images using VGG-16 and Neural Networks," *Procedia Computer Science*, vol. 218, pp. 357–366, 2023, doi: https://www.sciencedirect.com/science/article/pii/S1877050923000182#:~:text=In%20this%20work%2C%20a%20deep,937%20for%20the%20first%20dataset..

[2]
K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv.org*, Dec. 10, 2015. https://arxiv.org/abs/1512.03385

[3]
K. Team, "Keras documentation: ResNet and ResNetV2," *keras.io*. https://keras.io/api/applications/resnet/#resnet152v2-function

[4]
V. J, "Tutorial on Keras flow_from_dataframe," *Medium*, Apr. 27, 2019. https://vijayabhaskar96.medium.com/tutorial-on-keras-flow-from-dataframe-1fd4493d237c

[5]
H. Sharma, J. S. Jain, P. Bansal, and S. Gupta, "Feature Extraction and Classification of Chest X-Ray Images Using CNN to Detect Pneumonia," *IEEE Xplore*, Jan. 01, 2020.
https://ieeexplore.ieee.org/abstract/document/9057809/

[6]
S. Thakur, Y. Goplani, S. Arora, R. Upadhyay, and G. Sharma, "Chest X-Ray Images Based Automated Detection of Pneumonia Using Transfer Learning and CNN," *Proceedings of International Conference on Artificial Intelligence and Applications*, pp. 329–335, Jul. 2020, doi
: https://doi.org/10.1007/978-981-15-4992-2_31.

[7]
A. Kareem, H. Liu, and P. Sant, "Review on Pneumonia Image Detection: A Machine Learning Approach," *Human-Centric Intelligent Systems*, vol. 2, no. 1–2, pp. 31–43, May 2022, doi: https://doi.org/10.1007/s44230-022-00002-2.

Report

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dokumen.pub Internet Source	<1%
2	www.eurekaselect.com Internet Source	<1%
3	link.springer.com Internet Source	<1%
4	www.e3s-conferences.org Internet Source	<1%