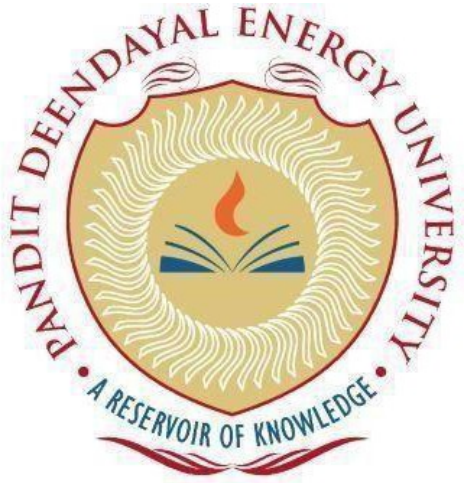
PANDIT DEENDAYAL ENERGY UNIVERSITY

SCHOOL OF TECHNOLOGY



Machine Learning Lab Project Report

on

Deep Learning-Based Diagnosis of Lung Diseases Using Chest X-Ray Images

B.Tech.

(Computer Science and Engineering)

Semester 7

Submitted To:
Dr. Yogesh Kumar

Submitted By:
Ritwik Garg, 20BCP135

1. Introduction

The anatomy of human lungs consists of microscopic air sacs called alveoli. These alveoli are situated at the end of the bronchioles and they serve the main purpose of facilitating the exchange of carbon dioxide and oxygen during the process of breathing in and breathing out. The alveoli have extremely thin cell membranes which allows for the seamless diffusion of gases, in and out of the air sac. Ultimately, these alveoli are essential to the process of breathing as they help introduce oxygen into the lungs and expel carbon dioxide simultaneously.

According to the World Health Organisation (WHO), Pneumonia is an acute respiratory infection that affects the lungs and particularly the alveoli inside. It may affect one or both lungs and is usually caused by infection through bacteria, viruses or fungi. The infection from a foreign substance prompts an almost immediate and severe response from the body's immune system which responds by sending an influx of white blood cells into the lungs along with other essential cells and substances. This causes the alveoli to fill up with fluid or pus inducing cough with phlegm, fever, chills and difficulty breathing. Ultimately this makes breathing painful for the infected patient and limits oxygen intake.

Streptococcus pneumonia is the most common pneumonia-causing agent in children and has claimed an estimated number of over 740,000 children under the age of 5 in 2019. This death toll accounted for 14% of the total deaths of children under 5 years old. Most deaths come from under developed and economically backward countries in Southern Asia and sub-Saharan Africa. Hence this CAD framework is especially relevant in countries like India which see a large number of pneumonia-related deaths year on year. Since most pneumonia related deaths occur in infants, we have chosen a dataset that contains 5000+ images of chest X-rays belonging to children of the ages 1 to 5 year old.

The early diagnosis of pneumonia is the most effective strategy to prevent the infection from becoming more severe and possibly fatal. The diagnosis of pneumonia is conducted through the use of radiological techniques such a Computed Tomography (CT) scans, radiography (X-rays) and Magnetic Resonance Imaging (MRI). Particularly, X-ray imaging provides a cost-effective and non-invasive avenue to perform diagnosis.

Fig 1 shows a contrast between an X-ray image of normal lungs and an X-ray image of pneumonic lungs. Generally, pneumonic lung X-rays exhibit regions of opacity that reflect as white coloured regions on the X-ray (indicated by red arrows). This condition is known as consolidation and indicates that a region of the lung has filled up with fluid, causing a change in density, which ultimately reflects in the from of regions of opacity in the X-ray. However, this observation is not conclusive as it may also indicate other issues like lung scarring or congestive heart failure.

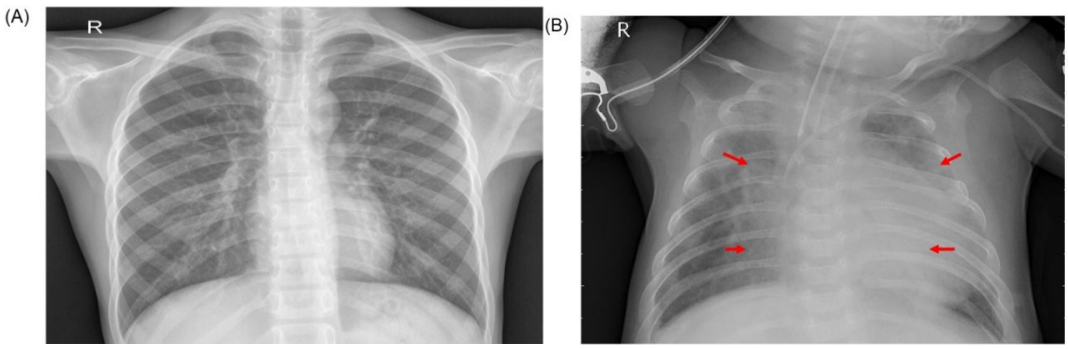


Fig 1. Two X-ray images of (A) healthy lungs and (B) pneumonic lungs

Recent advancements in Deep Learning have proven to be effective in a variety of medical applications. Models such as Convolution Neural Networks (CNNs) are especially useful in applications regarding complex computer vision problems. CNNs allow for added transferability between different datasets as they are able to perform feature selection as well feature extraction. However, a significant limiting factor of such models is the requirement of large amounts of clean and accurately labelled data, both of which are a challenging feat to achieve. In this report, we have developed a Computer Aided Diagnostic (CAD) framework that uses Deep Learning to accurately classify chest X-rays as normal or pneumonic.

In this project, the methodology mentioned in Fig 2. has been followed.

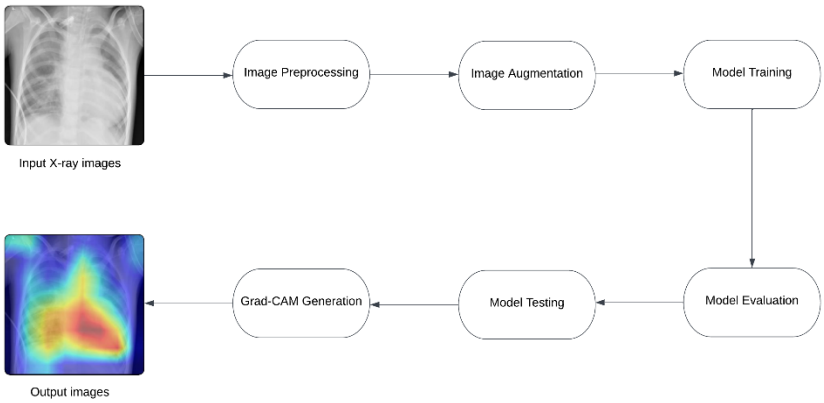


Fig 2. Proposed Methodology

2. Dataset

The data to train the model has been sourced from the Chest X-Ray Images (Pneumonia) dataset on Kaggle. The dataset contains a total of 5,863 X-Ray images in JPEG format with each image belonging to either one of two classes: Pneumonia or Normal. The X-ray images have been taken from an anterior-posterior view of the patients chest. The patients belonged to the age range of one to five years old. This data has been provided by the Guangzhou Women and Children's Medical Centre at Guangzhou, China.

Dataset Link: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

All chest radiographs were initially screened for quality control before being analysed, with all low quality or unreadable scans being removed. Before training the AI system, the diagnoses for the images were graded by two expert physicians. A third expert also reviewed the evaluation set to account for any grading errors.

The images in the dataset are of JPEG type and belong to two classes: Normal and Pneumonia. The dataset contains 3 folders namely: 'test', 'train' and 'val', referring to the test, train and validation datasets. Out of a total of 5,865 images, 4273 images belong to the pneumonia class while the remaining 1583 images belong to the normal class. The distribution of images among the datasets can be observed in Fig 3. And Fig 4.

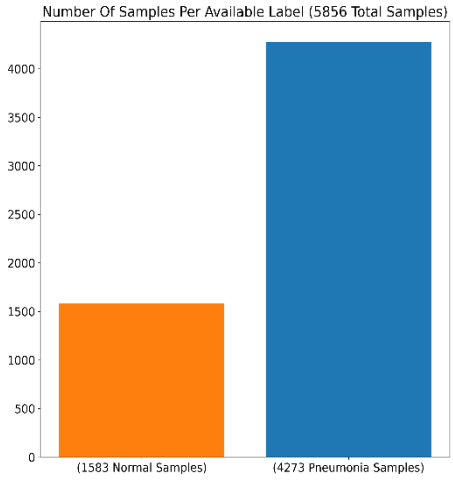


Fig 3. Class Distribution over entire Dataset

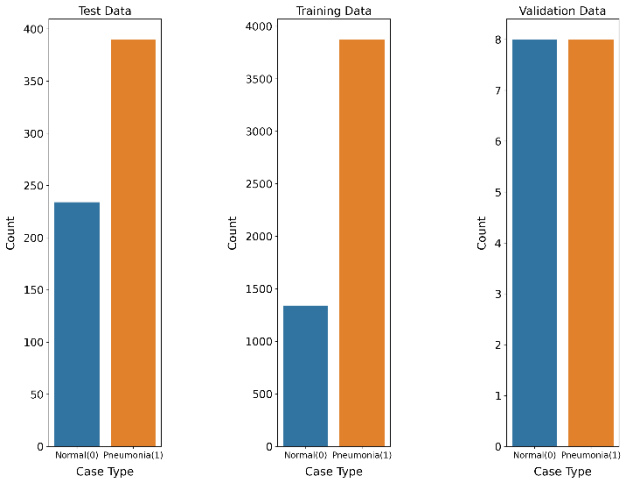


Fig 4. Class Distribution in Test, Train and Validation Datasets.

3. Data Pre-processing

The distribution of images over the dataset was highly uneven with the training dataset having over 4,000 images while the validation dataset only had a total of 8 images. Such an extreme division between the datasets is not preferable since it may hamper the models ability to learn and could lead to overfitting.

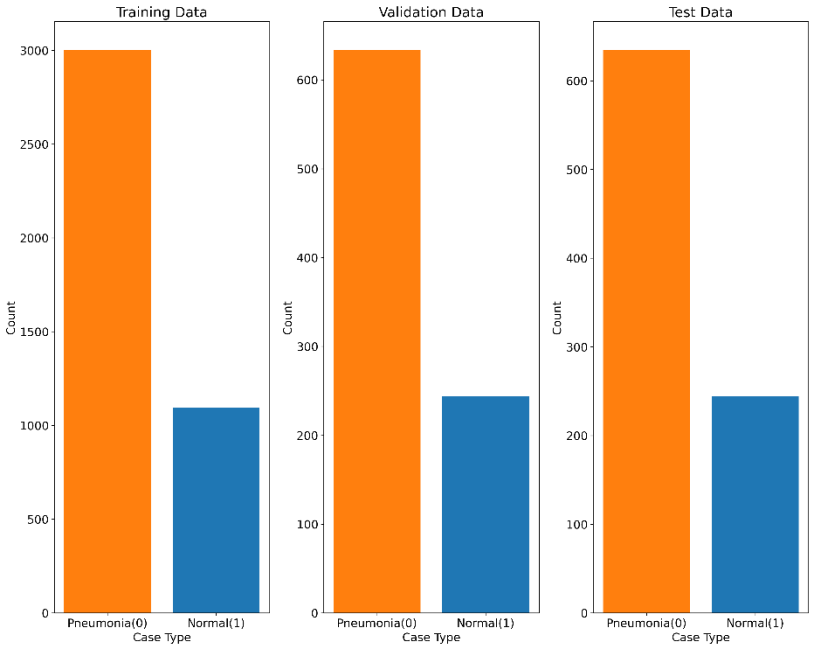


Fig 5. Class Distribution after Data Split

Hence, a new data split was devised which split the entire dataset of 5,865 images into a 70:15:15 ratio for the training, test and validation sets respectively. The newly created data split was homogenous and consistent and was fit for model training. The distribution of images in the dataset after the split can be observed in Fig 5.

The proposed framework uses a Convolutional Neural Network (CNN) for the analysis and classification of chest X-ray images into normal and pneumonia positive classes. We have opted to use a CNN for this application since CNNs have the ability to perform feature selection and feature extraction autonomously without the need for human supervision. This is of significant importance in applications such as chest X-ray analysis since programmers themselves do not have significant medical expertise to be able to hand pick relevant features that can be fed into the learning model.

The proposed model takes chest X-rays as an input. The X-rays are first passed through an array of filters in order to improve the image clarity and reduce the noise in the image. In this model, first a grayscale filter is applied, followed by a median filter and an unsharp mask, and finally Contrast Limiting Adaptive Histogram Equalisation (CLAHE) is performed. This is done as a part of the image pre-processing phase. The resultant image is much clearer and hence enables the model to learn more accurately and efficiently.

Followed by this, image augmentation has been performed on the pre-processed images. The image augmentation techniques used included rescaling, zooming, height and width shifting and/or rotating the images. This was done in order to create variance in the data, leading to a more robust and generalized model that is capable of adapting to new and different datasets.

3.1 Image Pre-processing

The quality of images given as an input have a significant effect on the ability of neural network to learn and make accurate predictions. The crux being that high quality data is essential for any machine learning applications. This emphasizes the requirement of image pre-processing techniques that are used to improve image quality.

Emil et. al. [4] proved that using a combination of median, Wiener, Bothat and Tophat filters greatly increased the quality of the X-ray image by removing noise and enhancing edges. Hence, we decided to incorporate 4 different filters into our image pre-processing pipeline, those being: grayscale filter, median filter, unsharp mask and CLAHE filter. The pre-processing pipeline has been depicted in Fig 6.

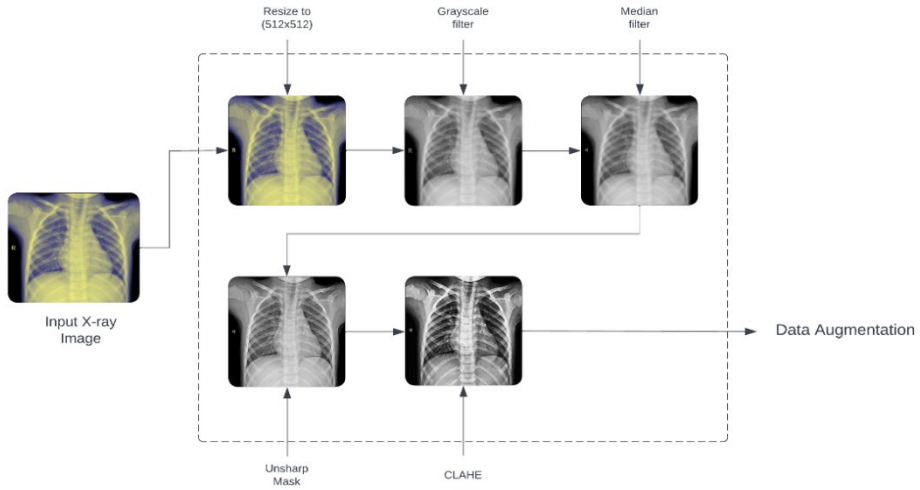


Fig 6. Image Pre-processing Pipeline

The pre-processing pipeline starts by taking an ambiguous sized X-Ray image as input and resizing it to a fixed size of (512, 512) pixels. This is done so that each resulting image has a homogenous size and the model doesn't discriminate between images of different sizes.

The resized image is then passed through a grayscale filter that converts the colour space of the image from Red, Green and Blue (RGB) to grayscale. This is done in order to again, homogenise the image to a single colour space so that the model doesn't discriminate between differently coloured images.

A median filter is then applied onto the grayscale image. A median filter is a non-linear filter that is effective in removing impulsive or salt-pepper noise. The median filter allows us to maintain the edges in the image while also helping to reduce random noise, hence proving effective in increasing the overall quality of the image.

Next, an unsharp mask is applied over the image. Unsharp mask is a technique used in image processing to sharpen images by increasing the contrast along the edges. It works by subtracting a blurred version of the image from the original image, thereby enhancing the high-frequency components of the image. This helps to enhance the appearance of edges within the images and results in a more crisp and clear image.

Finally, Contrast Limiting Adaptive Histogram Equalisation (CLAHE) is applied over the image. CLAHE is an image processing technique used to improve the image contrast. CLAHE, in contrast to traditional histogram equalisation, adapts the transformation to each small region of the image. This allows for better local contrast preservation and avoids over-amplification of noise in low-contrast areas.

The resulting image obtained after performing the aforementioned image processing techniques is magnitudes better than the original image in terms of image quality, contrast, noise presence and edge clarity. This can be clearly seen in Fig 7.

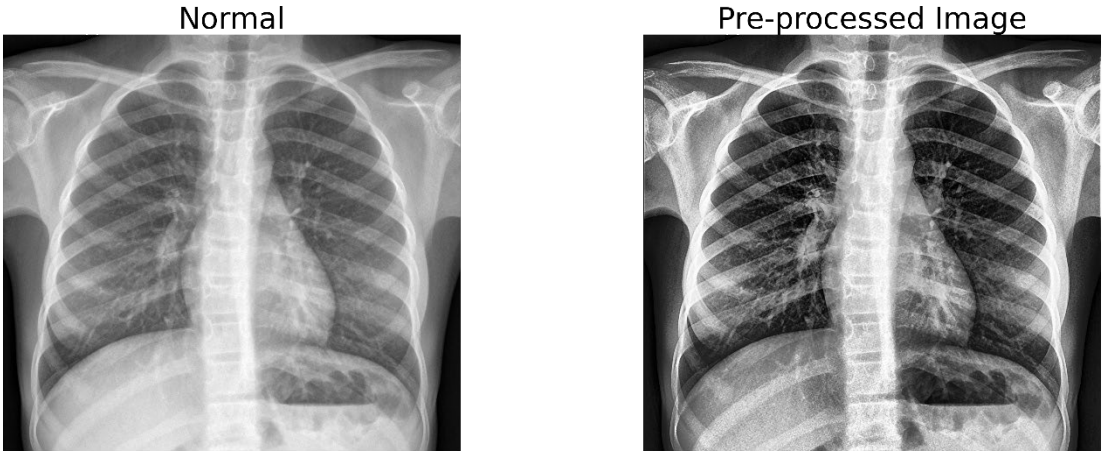


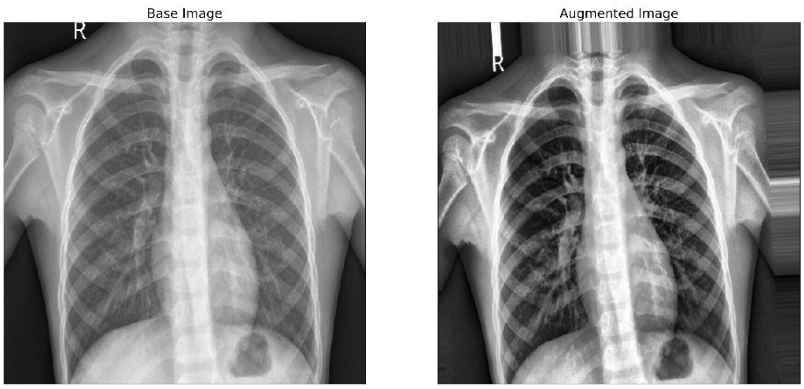
Fig 7. Normal Input Image vs. Pre-processed Image

3.2 Image Augmentation

Image augmentation is yet another integral part of the deep learning pipeline. Image augmentation is technique used to artificially increase the size of a dataset by applying various transformations to the image such as rotation, scaling, cropping, flipping and colour jittering. The objective of image augmentation is to generate data with higher variance and diversity compared to the original input data. This process is necessary since it allows the model to adapt and generalise to new and different datasets, hence increasing the models performance on new and unseen images. To sum up, image augmentation enables us to increase the size of the dataset while also helping to reduce overfitting and increases the model's robustness.

In the proposed framework, 5 image augmentation techniques were employed in order to generate new images. These 5 techniques included the rescale operation, zoom operation, rotation operation, height shift operation and width shift operation. Examples of the final augmented image has been shown in Fig 8.

The rescale operation converts an image of uint8 (8 bit integer) type to a float32 (32 bit floating point) type by dividing each pixel value by 255. This is done to normalise the image and bring all pixel values to a range between [0,1]. Normalisation reduces the magnitude of each pixel value in the image which



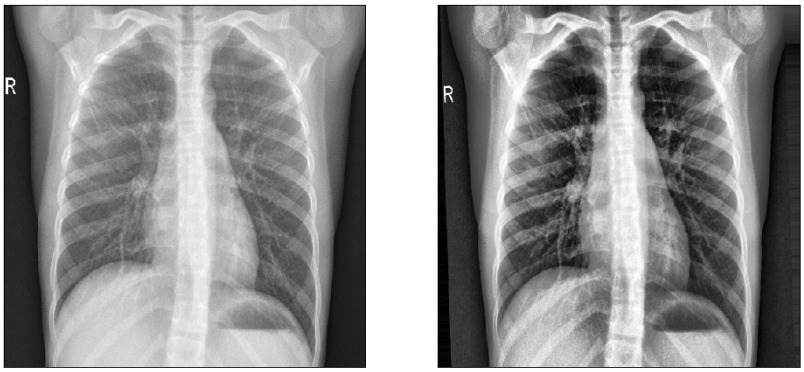


Fig 8. Base Images vs. Final Augmented Images

in turn reduces the impact of the differences in the intensity and contrast of the images, onto the performance of the machine learning algorithms. Rescaling also allows the model to converge faster since pixel values are now in a range of values between [0,1] instead of the original range of [0,255].

The zoom operation was used for zooming into or out of the image using a zoom range of 0.1. This implies that the image could be zoomed in or out by a random factor up to 10%. This operation helps the model to generate new and unseen images.

The rotation operation was used to rotate the images clockwise or anticlockwise using a rotation range of 5. This implies that the image could be rotated either clockwise or anti-clockwise by a random angle up to 5 degrees. This operation exposes the model to new image angles and hence enables the model to gain new insights.

The height and width shift operations were used to perform linear translation of the image along the y and x directions respectively. A height and width shift range of 0.05 was used. This implies that the image could randomly be shifted up to 5% of its total height or width in any direction during training.

A complete summary of the image augmentation techniques that we used has been mentioned in Table 1.

Method	**Setting**
Rescale	1./255
Zoom Range	0.1
Rotation Range	5
Height Shift	0.05
Width Shift	0.05

Table 1. Settings used during image augmentation

4. Model Training

The architecture of the neural network used has a major influence over the learning ability of a deep learning model. The architecture must be suitable for the use-case scenario that is being considered. In most computer vision related tasks, a CNN is used as the core architecture. A CNN is a discriminative classifier that can recognise complex patterns without requiring significant pre-processing. It is especially efficient at extracting global features from the image without the need of human supervision.

The neural network model consists of 1 input layer that is connected to 7 linearly connected convolutional blocks followed by a 2D Global Average Pooling Layer and finally 1 output neuron. Fig 9 depicts a detailed view of the entire neural network model architecture.

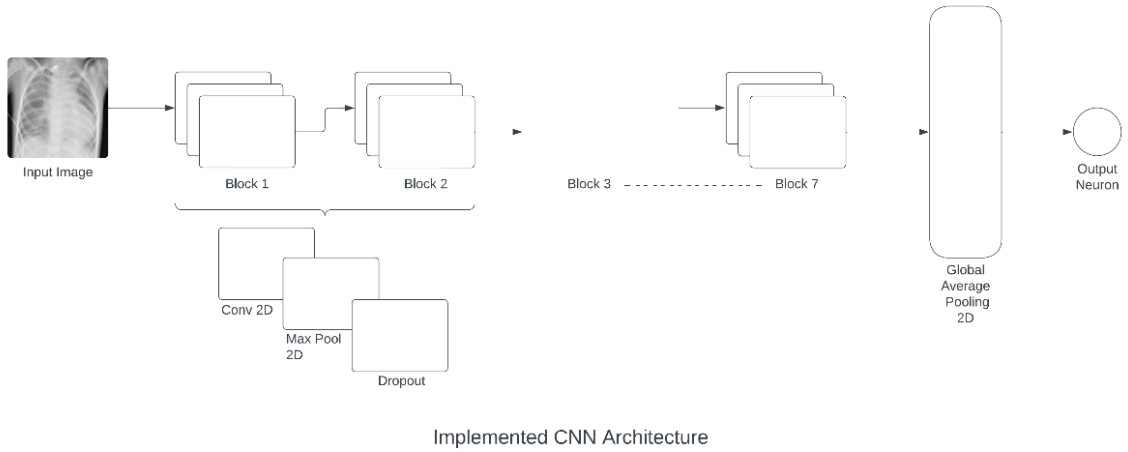


Fig 9. Architecture of Implemented CNN

The model takes a batch of images, each of shape (512, 512, 1) as input. This implies that images fed into the model are 512 x 512 sized grayscale images. This input layer is then connected to the first convolutional block. Each convolutional block is composed of 3 layers: a Conv2D layer, followed by a MaxPool2D layer and finally a Dropout Layer.

A Conv2D layer in a neural network is a two-dimensional convolutional layer that is commonly used in computer vision tasks like image recognition and classification. Convolutional layers operate by applying a set of filters to the input image in order to extract features that includes edges, corners, and other patterns that are relevant to the task at hand. Each filter in a Conv2D layer is a small weight matrix that is convolved with a local region of the input image to produce a single value in the output feature map. The Conv2D layer has several important hyperparameters that can be tuned to improve the performance of the network, such as the number of filters, the size of the filters, the stride of the convolution, and the padding used around the input image. By adjusting these hyperparameters, it's possible to create Conv2D layers that are optimized for a wide range of computer vision tasks. We have specified the hyperparameters such as the number of filters, kernel size, activation function and padding for each Conv2D layer. Table 2 shows a summary of the hyperparameter values that have been used in each Conv2D layer.

MaxPool2D layer is a type of pooling layer commonly used in CNNs. The MaxPool2D layer performs a down-sampling operation that reduces the spatial dimensions of the input tensor while preserving its most important features. In a MaxPool2D layer, the input tensor is divided into non-overlapping rectangular regions, and for each region, the maximum value is taken as the output. This operation

reduces the spatial dimensions of the input tensor, making it easier to process by subsequent layers in the network. In the proposed model, each MaxPool2D layer has a pooling size of 2.

Dropout layer is a regularisation technique that is commonly used to prevent overfitting in deep learning models. During each training epoch, a certain percentage of the input units are randomly selected and set to zero in a dropout layer. The corresponding neurons are thus "dropped out" of the network, and their contribution to the output is ignored. In the model a dropout rate of 0.1 is used. This implies that in each epoch 10% of the neurons in each block are “dropped out”, thus enabling the model to learn new features.

Block No.	Layer Name	Number of filters	Kernel size	Activation Function	Padding
1	conv2d_1	64	3	ReLu	same
2	conv2d_2	96	3	ReLu	same
3	conv2d_3	128	3	ReLu	same
4	conv2d_4	160	3	ReLu	same
5	conv2d_5	192	3	ReLu	same
6	conv2d_6	224	3	ReLu	same
7	conv2d_7	256	3	ReLu	same

Table 2. Hyperparameter values for Conv2D layers

The 7th block of the CNN Network is connected to a 2D Global Average Pooling (GAP) Layer. A global average pooling operation is applied to the entire feature map in a 2D GAP layer, resulting in a single value for each feature map channel. Unlike the MaxPool2D layer, which reduces the spatial dimensions of the input tensor by taking the maximum value in each pooling region, the 2D GAP layer computes the average of all feature map values along the spatial dimensions. This operation is carried out independently for each feature map channel, yielding a single value for each channel. In CNNs, a 2D GAP layer is frequently used in place of fully connected layers to reduce the number of parameters in the model and help prevent overfitting. Furthermore, because each output value corresponds to a specific feature map channel, 2D GAP layers can provide more interpretable features than fully connected layers.

Finally, the 2D GAP layer feeds into a single output neuron. Here a sigmoid activation function is used which classifies the chest X-Ray image into normal and pneumonia positive classes on the basis of threshold value. A threshold value of 0.75 was since it provided a good balance between precision, recall and accuracy values. This can be seen in Fig 14.

For model training, the adam optimizer was used for optimization and binary cross entropy was used for loss computation. The model was trained for a total of 30 epochs.

5. Result Analysis

5.1 CNN Results

The model was trained on a publically available dataset called 'Chest X-Ray Images (Pneumonia) dataset' that was taken from Kaggle. Out of a total of 5,865 images present in the dataset, 4273 images belong to the pneumonia class while the remaining 1583 images belong to the normal class.

The image data was first pre-processed and then underwent image augmentation to produce the final augmented image data which was to fed into the model for training. Training was done for 30 epochs in batch sizes of 32 images. The architecture of the CNN model used has been depicted in Fig 9.

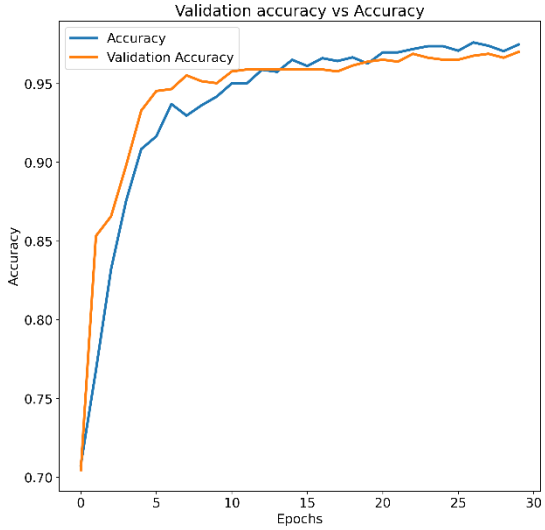


Fig 10. Training and Validation Accuracy Graph

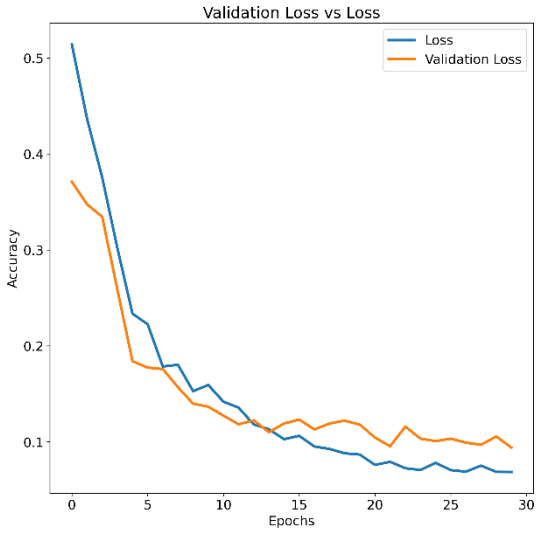


Fig 11. Training and Validation Loss Graph

Fig 10 shows the plot of training accuracy and validation accuracy versus the number of epochs. The model reaches an appreciable training and validation accuracy within 5 epochs. After the completion of 10 epochs, we can notice that the accuracy of the model starts to plateau and reaches a maximum of about 97% at the 28th epoch. We can observe that both, the training as well as validation accuracy show a continuous rise through out the training process. Both lines remain in close proximity to each other. This indicates that the model is performing well on both the training and validation data, and is not overfitting.

Fig 11 shows the plot of training loss and validation loss versus the number of epochs. We can see a continuous and consistent drop in both, the training and validation loss. Both training and validation loss lines remain within close proximity of each other throughout the training process. This is another positive indication that the model is not overfitting, and is able to generalize well to new data.

5.2 Evaluation Metric

The trained model was then evaluated using the Keras evaluation function. This was done in order to gain insights over how the model was performing on the test data. Fig 12 shows that the model was able to achieve an accuracy of 97.22%.

```
27/27 [==============================] - 54s 2s/step - loss: 0.0877 - accuracy: 0.9711
```

Fig 12. Evaluation Accuracy

Subsequently, a confusion matrix was also drawn to gain a deeper understanding of how well the model was able to predict the test data. Fig 13 shows a confusion matrix containing the number of correct and incorrect predictions. The model was able to make 839 correct predictions out of a total of 864 predictions. From this we can infer that the model has an accuracy of 97.11%. The model had an overall precision value of 97.91% and recall value of 98.06%.

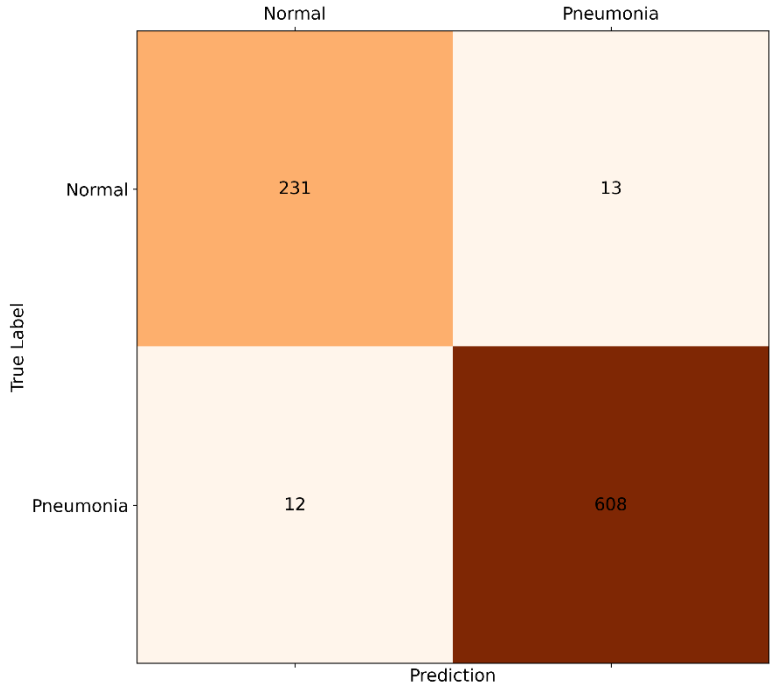


Fig 13. Confusion Matrix

Table 3 shows a complete classification report for each constituent class of the data.

	Precision	**Recall**	**F1-score**	**Support Cases**
Normal	0.95	0.95	0.95	224
Pneumonia	0.98	0.98	0.98	620
Accuracy	--	--	0.97	864
Macro Avg	0.96	0.96	0.96	864
Weighted Avg	0.97	0.97	0.97	864

Table 3. Classification Report

5.3 Threshold Calculation

Choosing the right threshold value can greatly impact the accuracy of predictions that are made by the deep learning model. Too low of a threshold may classify negative cases and positive and too high of a threshold may classify positive cases as negative. Hence choosing an optimal threshold value ensures that maximum accuracy is achieved while reducing the number of misclassified predictions.

In order to find the optimal threshold value, the model performance was tested for all threshold values between 0 to 100 and the corresponding metrics for each threshold value were calculated and plotted. These metrics included the precision, recall and accuracy of the model for that particular threshold value.

Fig 14 shows a graph where the precision, recall and accuracy for all threshold values ranging from 0 to 100 is plotted. It can be observed that the precision is low for low threshold values and is high for higher values of threshold. In contrast, the recall value is low for high values of threshold and is high for lower values of threshold. It can be observed that at a threshold value of 0.75 is optimal and achieves a good balance between precision, recall and accuracy values.
This can be seen as a vertical black line on the graph in Fig 14.

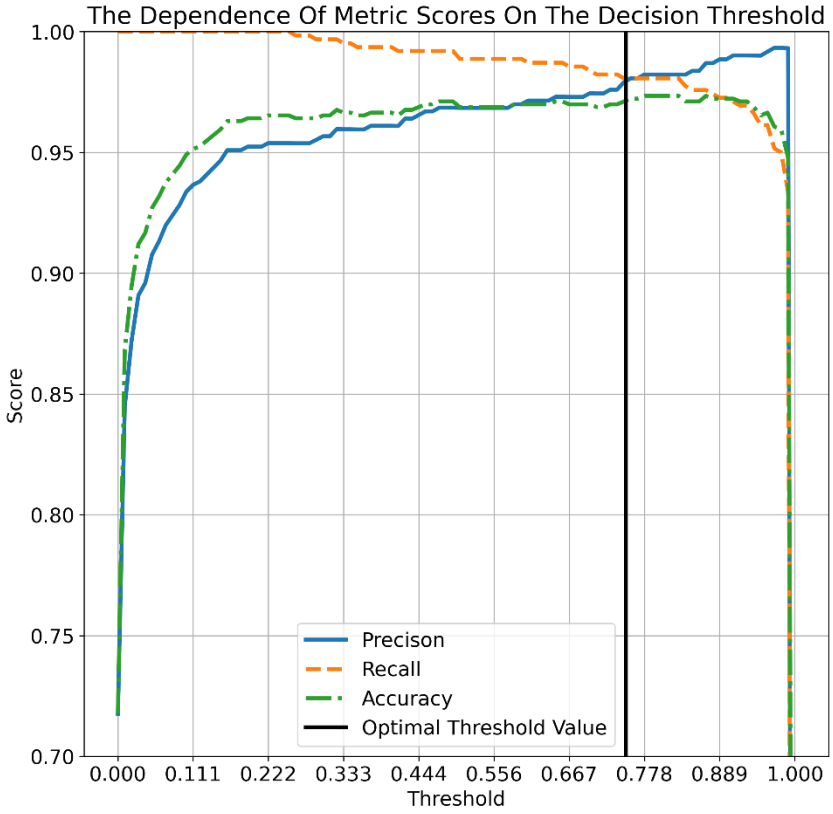


Fig 14. Variance of Metric Scores with Varying Threshold Values

5.4 Grad-CAM Visualisation

Grad-CAM (Gradient-weighted Class Activation Mapping) is a deep learning technique for visualising the regions of an input image that are relevant to a specific decision made by a convolutional neural network (CNN). Grad-CAM is a type of class activation map that highlights the most prominent regions in an image for a specific class.

Grad-CAM computes the gradients of the target class output in relation to the feature maps of the final convolutional layer. These gradients are then used to generate a weight map for each feature map channel. Combining the weight maps with the feature maps and applying a ReLU activation yields the final heatmap.

The resulting heatmap highlights the regions of the input image on which the CNN bases its classification decision. Grad-CAM can be used to explain a CNN's predictions as well as identify potential network weaknesses or biases.

Fig 15 shows a various chest X-Ray images along with their corresponding Grad-CAM heatmaps.

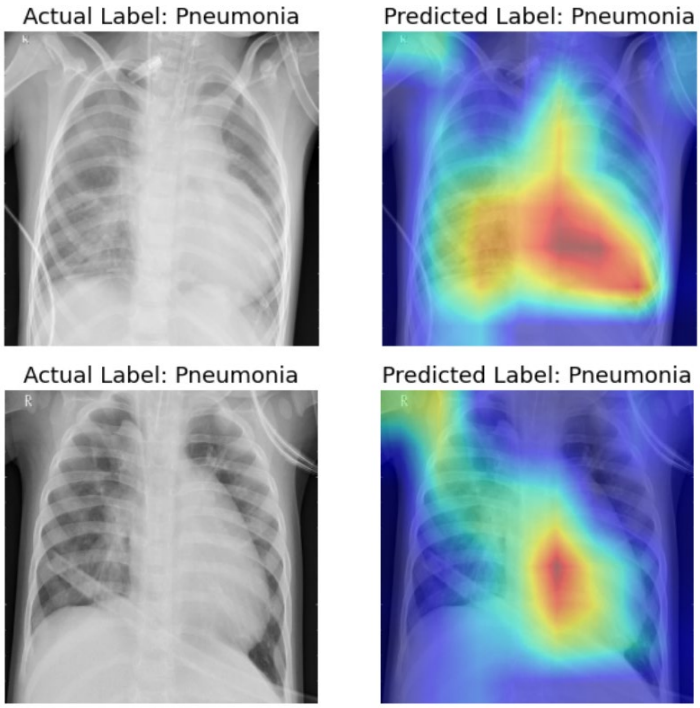


Fig 15. Grad-CAMs of Input X-Ray Images

6. Conclusion and Future Work

In conclusion, we were able to develop a robust and highly accurate CAD system that is able to effectively identify signs of pneumonia in chest X-Ray images. The model has an overall accuracy of 97.11% and precision and recall values of 97.91% and 98.06% respectively.

This CAD framework has significant applications in medical diagnostics and can be used a helping tool for medical professionals to help diagnose pulmonary related diseases such as pneumonia. This framework also finds significant application in medical diagnosis in areas where clinical expertise is unavailable or where there is a lack of appropriate medical resources.

However, it must be noted that the model is not completely reliable and should only be used as an advisory tool instead of being used an independent diagnostic tool. The model has only been trained on one dataset and hence may not be able to generalise well to X-Ray images in other datasets. In the future, a more variable and diverse dataset can be used to train the model inorder to improve the models ability to adapt and generalise to new data.

One significant limitation of this CAD framework is its inability to identify the correct features in the X-Ray image. As seen in Fig 15, the model tends to excessively highlight the top right and left hand corners of the X-Ray image. This shows that sometimes the model fails to correctly identify the affected regions in the lungs. A potential solution to this would be to use more complex pre-processing techniques to enhance the X-ray images. Image segmentation could be used to segment only the regions of the X-Ray that contains the lungs. This would help the model to only focus on the lungs and not surrounding features in the X-Ray.