A Deep Learning Approach for Pneumonia Classification from Chest X-Ray Images with Ensemble Modelling and Explainable AI

by

MST. NASRIN AKHTER
17366005

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mst. Nasrin Akhter
17366005

Approval

The thesis titled "A Deep Learning Approach for Pneumonia Classification from Chest X-Ray Images with Ensemble Modeling and Explainable AI" submitted by

1. Mst. Nasrin Akhter (17366005)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on June 08, 2021.

Examining Committee:

External Examiner:
(Member)

Haidar Ali,PhD
Professor
Department of Computer Science and Engineering
Dhaka University
haider@du.ac.bd

Internal Examiner:
(Member)

Md Khalilur Rhaman ,PhD
Associate Professor
Department of Computer Science and Engineering
Brac University
khalilur@bracu.ac.bd

Internal Examiner:
(Member)

Md. Golam Rabiul Alam ,PhD
Associate Professor
Department of Computer Science and Engineering
Brac University
rabiul.alam@bracu.ac.bd

Supervisor:
(Member)

Md. Ashraful Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University
ashraful.alam@bracu.ac.bd

Program Coordinator:
(Member)

Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University
amitabha@bracu.ac.bd

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University
skazi@bracu.ac.bd

Abstract

Pneumonia is one of those frightening diseases that has a high mortality rate among children and the elderly, with an estimated 2 million fatalities per year. Pneumonia affects the poorest people in Africa and Asia the most, due to a lack of medical surveillance in such areas. It is responsible for 28 percent of all child fatalities in Bangladesh each year, and the number is likely to be considerably higher. In recent years, a number of computer-assisted diagnostic methods have been developed to assist in the detection of pneumonia. In this study, an efficient model PNEXAI is proposed to identify pneumonia utilizing Chest X-Ray images. We gathered and classified data using VGG16, VGG19, ResNet 50, ResNet 101 and Inception v3. The accuracy rate of 97.17% was reached by VGG16, 97.69% by VGG19, 97.35% by ResNet50, 95.63% by ResNet101, and 94.86% by Inception V3, respectively. We then developed an ensemble model containing the top three classifications (VGG16, VGG19 and ResNet50) which delivered 98.46 % of best overall accuracy. Finally, to better comprehend our categorization, we included explainable artificial intelligence in our model.

Keywords: Pneumonia, Chest x-ray, Transfer learning, Convolutional Neural Network, VGG16, VGG19, ResNet50, ResNet101, Inception V3, PNEXAI.

Acknowledgement

At first, I thank almighty Allah for enabling me to complete this thesis work successfully. With gracious privilege I would like to show my deepest respect to my honorable supervisor Dr. Md. Ashraful Alam, Assistant Professor, Department of Computer Science and Engineering, BRAC; for his immensely knowledgeable guidance and supervision throughout this work. His generosity and helpful demeanor were the key to make this work a wholesome one.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The danger of pneumonia is huge, particularly in developing countries where billions live under the poverty line and live in an environment that is not safe for health [1]. The World Health Organization (WHO) gauges that more than 4 million unexpected losses happen every year because of air contamination related ailments including pneumonia. More than 150 million individuals get contaminated with pneumonia on a yearly basis, particularly youngsters that are under 5 years of age [2]. In rural and less developed areas, the issue can become even more morbid because of the shortage of medical assets and facilities. In Africa, there is a gap of 2.3 million doctors [3]. As a result, the people of these regions do not usually get the exact and quick medical support that the pneumonia treatment requires [4]. Even in case of emergency, the treatment is often very expensive and may cost a fortune. The treatment becomes even more expensive and difficult if pneumonia gets diagnosed at a later stage of the disease. Late detection of pneumonia can make it become fatal. This is especially true in the case of the children [5].

1.2 Aims and Objectives

Pneumonia must be identified as early as possible in order to minimize fatal harm from pneumonia and to lower the expense of medications. The aim is to develop a profound study model to determine the result of chest X-ray disease. Our main goal is to visualize the result and to comprehend the classification aspects. Understanding the relevance of features in future study can lead to higher performance.

1.3 Research Methodology

At first, we have observed several Machine Learning models such as InceptionV3, VGG16, VGG19, ResNet50, and ResNet101. We have pre-processed our Chest X-ray image data into a well-defined form of $224 \times 224 \times 3$ in the initial stage. Then we trained and evaluated those machine learning models. After evaluation, we took the best three classifiers and constructed various forms of ensemble with them. Finally, after comparing the performance of the ensembles, we applied Explainable AI to visualize the classification.

1.4 Research Orientation

In chapter 2, we discussed prior studies relating to our research topic. Then, in Chapter 3, we went over each architecture, convolution layer, and activation function we employed in our study. In Chapter 4, The implementation of our thesis work is discussed. We have also shown how we distribute our datasets and then describe the pre-processing strategies for the given data set. In chapter 5 we illustrated our results after the algorithms were implemented and then explained the analysis of the outcomes. Finally, we concluded and discussed the future work of our research in Chapter 6.

Chapter 2

Related Work

2.1 Literature Review

Nowadays Artificial intelligence(AI) is performing a great role in the medical diagnosis process, forecasting and prediction of disease evolution.There are so many research papers on this. For instance, MRI (Magnetic resonance imaging) can be used to diagnose brain abnormalities [1], clinical data can be used to diagnose cardiac disorders [2], and radiographs (X-rays) can be used to diagnose breast lesions [3]. There have also been numerous studies on the use of AI to detect Pneumonia.

A. Saraiva et al [4] demonstrated the classification of Pneumonia using Convolution Neural Network (CNN).The authors used a set of labeled Optical Coherence Tomography (OCT) images and chest x-ray images for their classification where a total of 5863 images were there in the dataset. Thedataset had two different classes, one is normal and another one is pneumonia. In their paper,they showed the implementation of k-fold cross-validation for CNN models where they achieved 95.30% accuracy.

In another research, Kermany et al [6] proposed an architecture of Convolutional Neural Network that used transfer learning technique to classify affected chest images. Although the main research was about the diagnosis of OCT images of the retina, a diagnosis of multiclass pediatric thoracic radiographs was also done in the extended part of the paper. For the main part, they classified four classes of OCT images: Choroidal Neovascularization (CNV), Diabetic Macular Edema (DME), drusen and normal. They obtained a precision of 96.6with a sensitivity of 97.8%, a specificity of 97.4% for this classification. On the other hand, in the case of the classification between pneumonia and normal x-ray images, they achieved an accuracy of 92.8%.

Researches on lung related diseases have also been done on various other domain of datasets apart from x-ray image dataset. Several researchers have proposed different algorithms for the diagnosis of lung diseases based on sound data. One of the parameters used for the detection of pulmonary sound is entropy. There are differences in the sound of a normal respiratory system and a system with the pathologies of pneumonia. A. Rizal et al [7] discussed several measures of entropy for the classification of pneumonia based on pulmonary sounds. The paper revealed that the usage of a single entropy was not enough to achieve high accuracy. Therefore, seven entropies were applied which achieved 94.95% accuracy using multilayer perceptron.

The structural Co-occurrence Matrix method for the classification of malignant and benign nodules was proposed by M. B. Rodrigues et al. [8] In the paper, they also managed to figure out the level of malignancy. The structural Co-occurrence Matrix technique was used to extract features of the nodule images so that the classification can be done. The process was implemented using gray-scale images of the Hounsfield unit with four filters, creating eight different configurations. The authors applied multilayer perceptron, SVM, k-Nearest Neighbors algorithm in two stages. One of the stages was to classify the nodule images as malignant or benign, and another one was to provide a level of malignancy to the nodule's pulmonary lesions. The level of malignancy was described to be between 0 and 5. They achieved an accuracy of 74.5% in their first task and accuracy of 53.2% in the second one.

In this paper [9], the researcher completed the research in two stages. In the first stage, heatmaps of different CNN models were generated and combined in the ensembled model.Then ,They used XAI technique to identify the region of interest for the classification .By which explainability and interpretability problems can be reduced.They ensembled the best result providing models and then tested it on a small dataset of pediatric X-rays. In the second stage , a new ensembled model is generated and trained with a smaller dataset. They believed that their newly created model had higher accuracy than the other pneumonia detection dataset.

In the study [10], the authors created a dataset consisting of 35,389 chest x-ray images and trained a prediction model which is capable of detecting pneumonia.The Bayesian network is used to create a XAI model from different CNN models. The findings show that multi-source data have improved efficiency and provide an intuitive description of diagnostic results.

The researchers proposed and built XAI approaches for COVID-19 classification models in the research [11], as well as comparing them. The findings suggest that by providing more detailed information from the learned XAI models' outputs, quantitative and qualitative visualizations might help physicians comprehend and make better judgments.

In addition, the researchers used an ensemble deep learning network to identify COVID-19 from CT scan images in article [12]. To produce model parameters, the model uses transfer learning, and it has pre-trained three deep CNN models: AlexNet, GoogleNet, and ResNet. In addition, relative majority voting is used to produce EDL-COVID, an ensemble classifier. Finally, the accuracy, sensitivity, specificity, F value, and Matthew's correlation coefficient of the ensemble classifier were compared to three different component classifiers.

In addition, a future paradigm is used in the article [13] for the COVID-19 risk prediction based on XAI (Explainable Artificial Intelligence). In order to predict the infection risk of COVID-19 in a non-clinical setting, the researchers developed a two-step technique.Initially, the primary risk of COVID-19 infection is determined by carefully examining selected factors linked to COVID-19 infection symptoms. In the second stage of the explainable AI based prospective framework that they have developed, after the result of the first step is obtained, an optional prediction system is also offered by anal yzing the chest x-ray images.

The paper [14] proposed a deep CNN architecture-based technique for identifying COVID-19 infected patients using chest x-ray images. Many cutting-edge CNN models, including DenseNet201, Resnet50V2, and InceptionV3, are merged and used in the proposed model. Individually trained models are then combined to predict a class value using a weighted average ensemble technique.

In this research [15], the author used a dataset containing 6000 chest x-ray images of children and trained these data in 12 convolution neural network models. They found VGG-19 as the best result providing model among all 12 CNN models. Additionally, these CNN models were combined together by using some learning methods such as Support Vector Machine, k-Nearest Neighbor, Random Forest, Naive Bayes and Multilayer Perceptron. Their combined model provided 96.47% accuracy, 96.46% F1 score, and 96.46% Precision.The author believes that their model will help the specialists to get faster and accurate results from chest x-ray images which will lead to a proper treatment.

This research describes how machine learning techniques are used to analyse chest X-ray images to aid in the diagnosing process. The project focuses on developing a processing model using a deep learning method based on a convolutional neural network. This model is tasked with assisting with a classification problem including determining whether a chest X-ray reveals alterations associated with pneumonia or not, and then categorizing the X-ray images into two categories based on the detection results [16].

Rohit KunduI, Ritacheta Das, Zong Woo Geem, Gi-Tae Han, Ram Sarkar created a computer-aided diagnosis system for automated pneumonia detection utilizing chest X-ray pictures in this paper work [17] to deal with the shortage of accessible data, they used deep transfer learning and created an ensemble of three convolutional neural network models: GoogLeNet, ResNet-18, and DenseNet-121. The weights provided to the base learners were chosen using a unique approach, resulting in a weighted average ensemble strategy.The scores of four typical assessment metrics, precision, recall, f1-score, and area under the curve, are fused to generate the weight vector, which was frequently set experimentally in studies in the literature, an approach that is prone to inaccuracy. Using a five-fold cross-validation scheme, the suggested approach was tested on two publicly available pneumonia X-ray datasets provided by Kermany et al. and the Radiological Society of North America (RSNA), respectively. On the Kermany and RSNA datasets, the suggested technique attained accuracy rates of 98.81% and 86.85%, and sensitivity rates of 98.80% and 87.02%, respectively. The results outperformed those of state-of-the-art approaches, and the method outperformed the commonly used ensemble techniques. This study created an automated CAD system that employs deep transfer learning to categorize chest X-ray pictures into two categories: “Pneumonia” and “Normal”. The ensemble architecture uses the decision scores from three CNN models, GoogleNet, ResNet-18, and DenseNet-121, to construct a weighted average ensemble. The classifier weights were calculated by combining precision, recall, f1-score, and AUC using the hyperbolic tangent function. On the Kermany dataset, the framework achieved 98.81% accuracy, 98.80% sensitivity, 98.82% precision, and 98.79% f1-score on the Kermany dataset, and 86.86% accuracy, 87.02% sensitivity, 86.89% precision, and 86.95% On these two datasets, it outperformed the competition. The proposed model has been statistically validated using McNemar's and ANOVA tests. Furthermore, because the suggested ensemble model is domain-independent, it may be used to a wide range of computer vision tasks.

In this work [18] by Mohammad Farukh Hashmi, Satyarth Katiyar, Avinash G Keskar, Neeraj Dhanraj Bokde and Zong Woo Geem proposes an efficient methodology for detecting pneumonia using digital chest X-ray images, which could help radiologists make better decisions. The ideal combination of weighted predictions from state-of-the-art deep learning models such as ResNet18, Xception, InceptionV3, DenseNet121, and MobileNetV3 is based on a weighted classifier. The network predicts the outcome based on the dataset's quality. Transfer learning is used to fine-tune deep learning models for training and validation. Partial data augmentation is used to balance the training dataset expansion.The suggested weighted classifier outperforms all other models. Finally, the model is evaluated not just on test accuracy, but also on AUC. AUC of 99.76 for the final proposed weighted classifier model on unseen data from Guangzhou Women and Children's Medical Center pneumonia dataset. As a result, the proposed approach can help radiologists diagnose pneumonia faster. Consequently, research and development on computer-aided diagnosis is urgently needed to reduce pneumonia mortality. The use of deep learning and computer vision algorithms in biological image identification has proven to be particularly effective in providing rapid and accurate illness diagnosis. Deep learning-based algorithms can't yet replace qualified doctors in medical diagnosis, but they're meant to help. This research describes a strategy based on deep learning and convolutional neural networks that can automatically diagnose pneumonia in patients. The proposed method uses a deep transfer learning algorithm to extract features from X-ray images that automatically describe disease and identify pneumonia. Due to its high test accuracy (98.43) and AUC score, the proposed technique could be used in clinical decision making (99.76). It can only help radiologists make decisions; a specialist must make the final call. In terms of testing accuracy, the proposed weighted classifier surpassed the condition where each model had equal weights by 0.98%. It was also proven that deep learning-based algorithms could detect pneumonia in chest X-rays using activation maps. The suggested weighted classifier enhanced testing accuracy by 0.43% compared to DenseNet121, which is significant on a large test dataset.

2.2 Neural Network

2.2.1 Biological Neuron

Our brain's basic computational unit is the neuron. We have 86 billion neurons that are linked together by approximately $10^{14} - 10^{15}$ synapses. Each neuron receives data input from its dendrites and sends out signals via its (single) axon. The axon eventually spreads out and attaches to the dendrites of other neurons through synapses [19][20] .In the neuronal calculation model, the signal x0 goes through the axons, interacting multiplying with the other neuron's dendrites, on the basis of the synaptic strength(w_0). Synaptic strength w_0 can be learned and the influence of one neuron on another can be controlled.

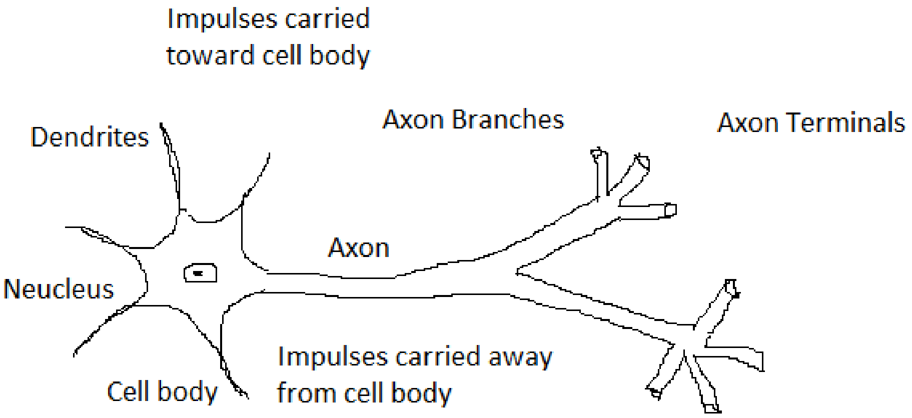

Impulses carried toward cell body
Dendrites
Axon Branches
Axon Terminals
Axon
Neucleus
Impulses carried away from cell body
Cell body

Figure 2.1: Biological Neuron

The dendrites pass the signal into the body of the cell in the basic model, where they are all summed up. If the final amount exceeds a certain limit, the neuron can incinerate and send its axon a spike. In the model, we assume that the accurate time frames of spikes are irrelevant and that information is communicated only at the frequency of firing [21].

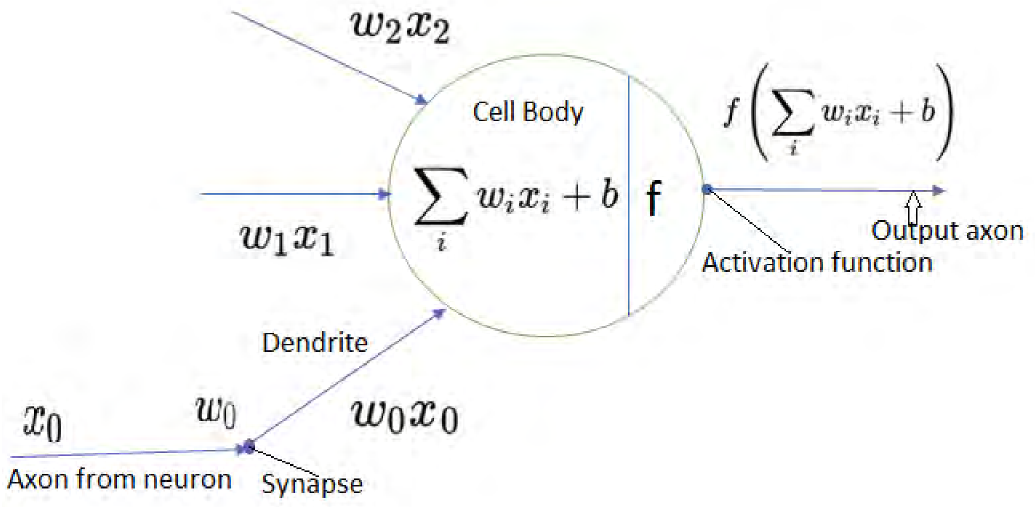


Figure 2.2: Mathematical model of biological neuron

2.2.2 Basics of Neural Network

From the name neural network we can understand that it is a network of neurons. Artificial neuron network consists of artificial neurons. There are links between the neurons and these links are modeled by some weights. For positive weights the connection is excitatory and for negative weights it is inhibitory. These weights are multiplied with the input value and summed which makes a linear combination.To control the output of the system , Activation function is used. There are many activation functions such as ReLU, Soft Max, Tanh etc. Based on the activation function output value can be 0 , 1 or 1 and −1.

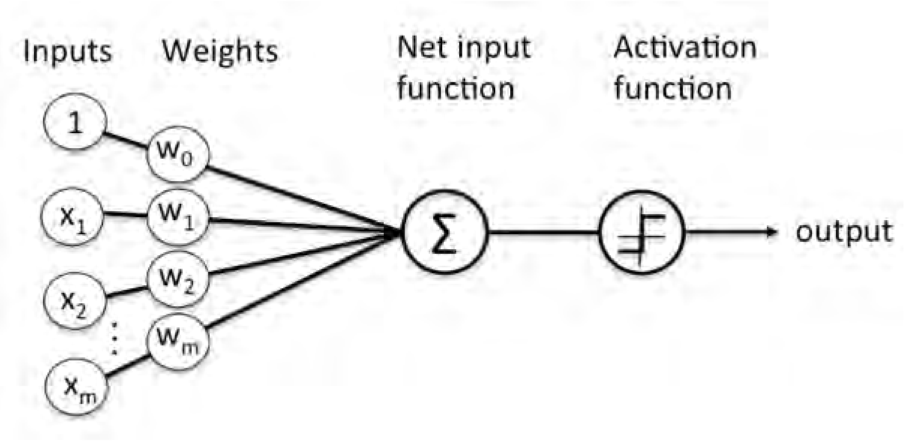


Figure 2.3: Block diagram of Neural network

There are 3 basic layers in a Neural network. These are-Input layer, Hidden layer and output layer. In the input layer , input values are passed. Here, input values are the pixel matrix which is generated from input images. In the hidden layer, the input values are divided into different regions with the help of activation functions. In the output layer, all values generated from the hidden layer are combined and give a final result.

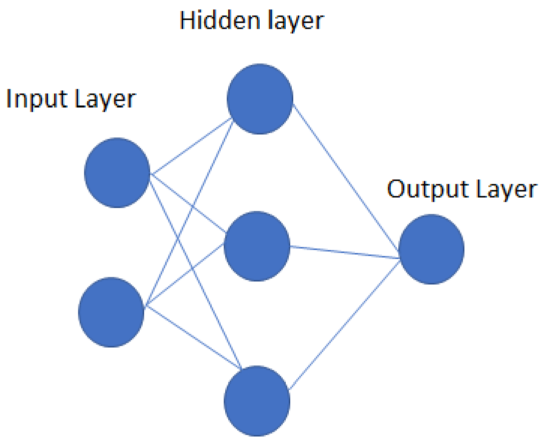


Figure 2.4: Main components of neural network

2.3 Convolutional Neural Networks

Convolutional Neural Networks are the most popular deep learning model to classify images. In general, there are two main parts of a CNN model [22] [23]. The first part is the convolution part that is used to extract feature information from images. CNN models treat images like a 2-dimensional matrix and multiply the image with a convolution kernel [24]. The value of the convolution kernel is determined by the CNN model itself and the kernel may take various shapes such as 3x3, 5x5 and 7x7 [25]. The second part of a CNN model consists of the actual classifier that classifies different labels from the extracted features provided by the convolution layers. The classifier part of a CNN may consist of different types of classifiers such as: Fully Connected (FC) layers, Support Vector Machine (SVM) or other general classifiers [26] [27].

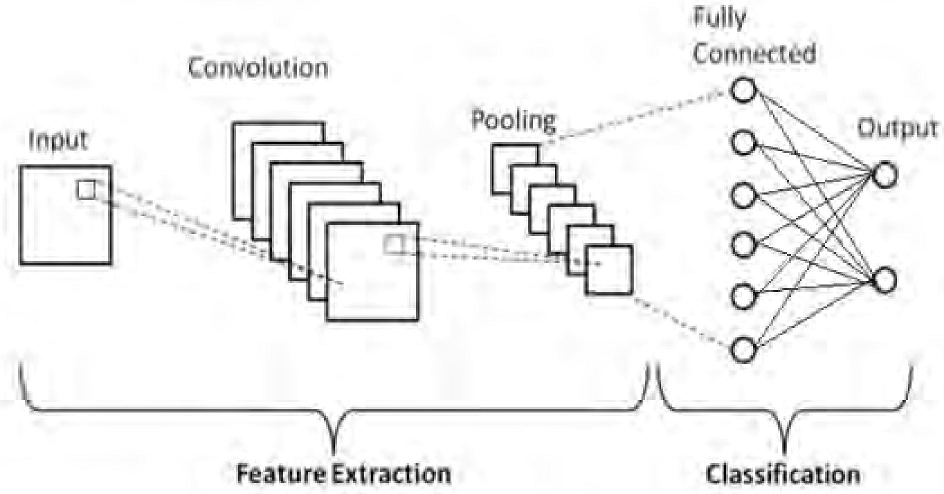


Figure 2.5: Basic Convolution Neural Network Architecture

Convolutional neural network consists of some layers, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm [28].

2.3.1 Convolution Layer

In Convolution layer [29] , weights of each neuron will be multiplied with the specific pixels of the input image to which it is connected. After the summation of all calculated values will generate an output value for a neuron. An activation function is used to control the output value, generally rectified linear unit which is also known as ReLU is used to control the the value of convolution layer.There are some others activation functions are also available.Feature extractions are completed in this layer. Kernel is the main parameter of this layer.

Kernel/Filter

Kernel is a matrix which is multiply to the image matrix to extract image features.Kernel size can be $3 * 3$, $5 * 5$ or $7 * 7$.
If we consider Image Matrix Dimension as IMD.

$$IMD = H * W * D \tag{2.1}$$

Here, H is the height, W is the width and D is the dimension of the image. A filter /kernel size is mentioned by k.

$$K = (fh * fw * d) \tag{2.2}$$

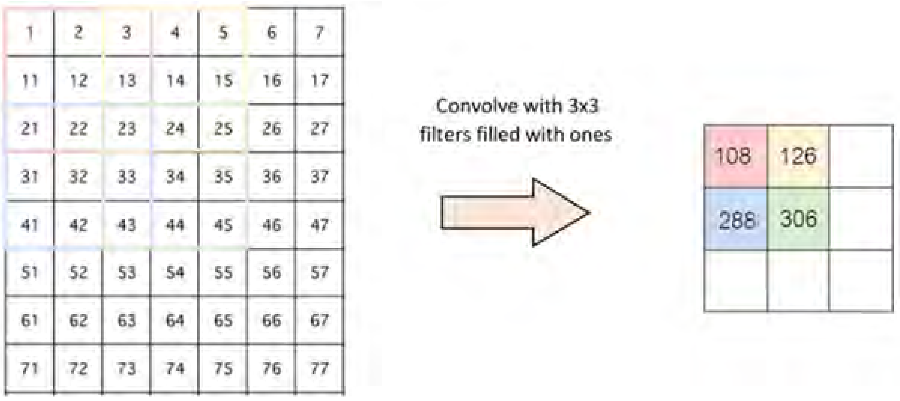
fh is kernel height, fw is kernel width and d is kernel dimension. Outputs of a volume dimension is

$$D = (h - fh + 1) + (w - fw + 1) + 1 \tag{2.3}$$

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters [30].

Strides

Stride is a parameter which is moving through the input matrix. Stride value can be 1 or 2 or 3. Kernel moves through the input matrix based on stride value. If stride is 1 kernel moves 1 pixel , if the stride is 2 kernel moves 2 pixels . [31]. Figure 2.6 shows an example of strides.



1	2	3	4	5	6	7
11	12	13	14	15	16	17
21	22	23	24	25	26	27
31	32	33	34	35	36	37
41	42	43	44	45	46	47
51	52	53	54	55	56	57
61	62	63	64	65	66	67
71	72	73	74	75	76	77

108	126	
288	306	

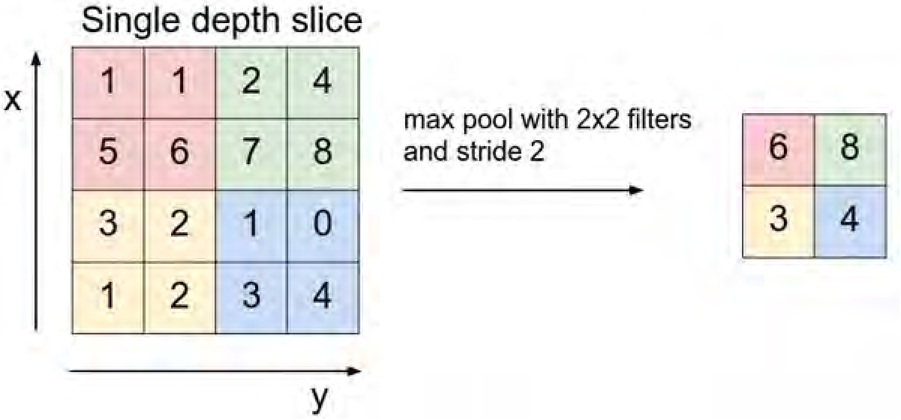
Figure 2.6: Example of Strides

Padding

When a kernel can not fit properly in the input matrix we can increase the input matrix size by adding 0 in all sides. Adding these additional zeros are called padding. If we do not add padding we have to subtract the extra part of the image which is not fitting. But by this way , our image features can be lossed. So padding is the best option.In general practice zero padding is used [32].

2.3.2 Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types such as, Max Pooling, Average Pooling, Sum Pooling.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 2.7: Example of Pooling Layer

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling

2.3.3 Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network [33] [34].

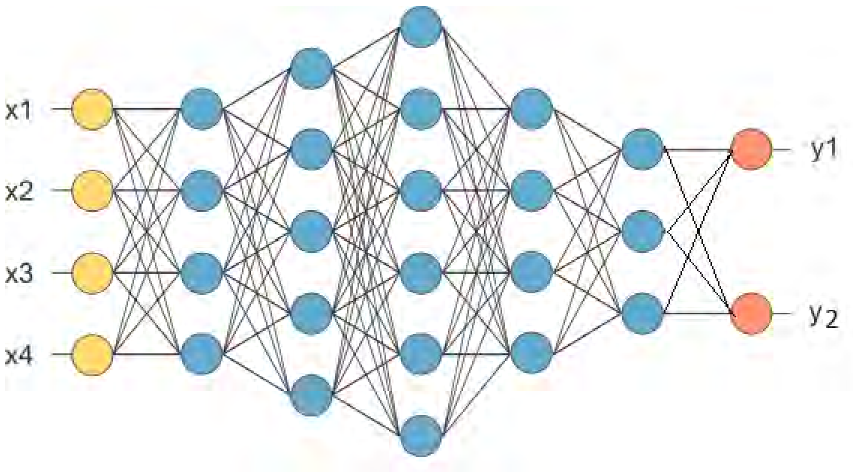


Figure 2.8: Flattened as Fully Connected layer after Pooling Layer

In Figure 2.8, the feature map matrix will be converted as vector $(x_1, x_2, x_3, \ldots, x_n)$. With the fully connected layers, features are combined together to create a model. Finally, an activation function such as ReLU is used to classify the outputs as y1 and y2.

ReLU Activation Layer

ReLU stands for Rectified Linear Unit. It is an activation function that converts all the numerical values to a value between 0 and infinity.

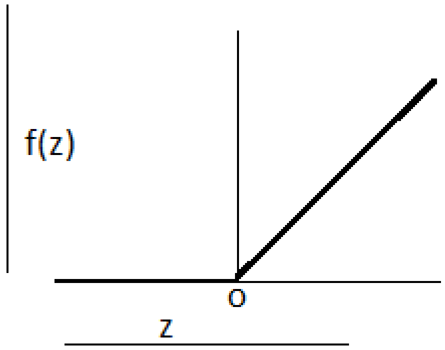


Figure 2.9: Rectified Linear Unit

The ReLU function is,

$$R(z) = max(0, z) \tag{2.4}$$

From the equation, we can see that, if the value of z is under zero, the ReLU function converts is back to zero. If the value of z is positive, the ReLU function returns whatever the value is. This function provides non-linearity to the entire CNN hypothesis. Figure 10 shows the functions of Rectified Linear unit [35].

2.3.4 Adam Optimizer

Adam optimizer is a modified version of stochastic gradient descent that helps to correct the values of the weights of a CNN through back-propagation. Adam optimizer is developed specifically to work well with Deep CNNs [36].

2.3.5 Cross Validation

In a machine prediction or learning task, cross-validation is one of the most important approaches for method evaluation and parameter selection. As a result, the model was evaluated using K-fold cross validation.

The sample is divided into k equal-sized sets at random in K-fold cross-validation. A single set is segregated as validation data to test the model in each of the k shares, while the remaining k 1 sets are used as training data. The cross-validation procedure is then performed k times, with each of the k sets being validated only once [37].The method's evaluation index is then calculated using the mean performance.

This method is computationally expensive, but it fully utilizes the entire set of 30 data, which is especially useful when the number of samples is small [38]. This method can also show how the trained model can be generalized to unknown data, avoiding the use of data that has been deliberately chosen.

2.3.6 Confusion Matrix

The confusion matrix is an array that contains correct and incorrect predictions of the algorithm and the actual situation [7].

Elements of confusion Matrix are,

- **True Positive (TP):** Number of individuals who really have pneumonia as indicated by the model.
- **False Negative (FN):** Number of individuals who have pneumonia but classified as healthy.
- **False Positive (FP):** Number of individuals who are actually healthy, however, classified as pneumonia, as per the model.
- **True Negative (TN):** Number of individuals who are actually healthy and classified as healthy, indicated by the model.

From this theory, accuracy, F1 score, sensitivity, specificity and precision are calculated using the following equations,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

$$f1 - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.8}$$

Chapter 3

Research Methodology

3.1 Our Methodology

The workflow diagram in Figure 3.1 gives an overview of each step we have done to train and evaluate our models.

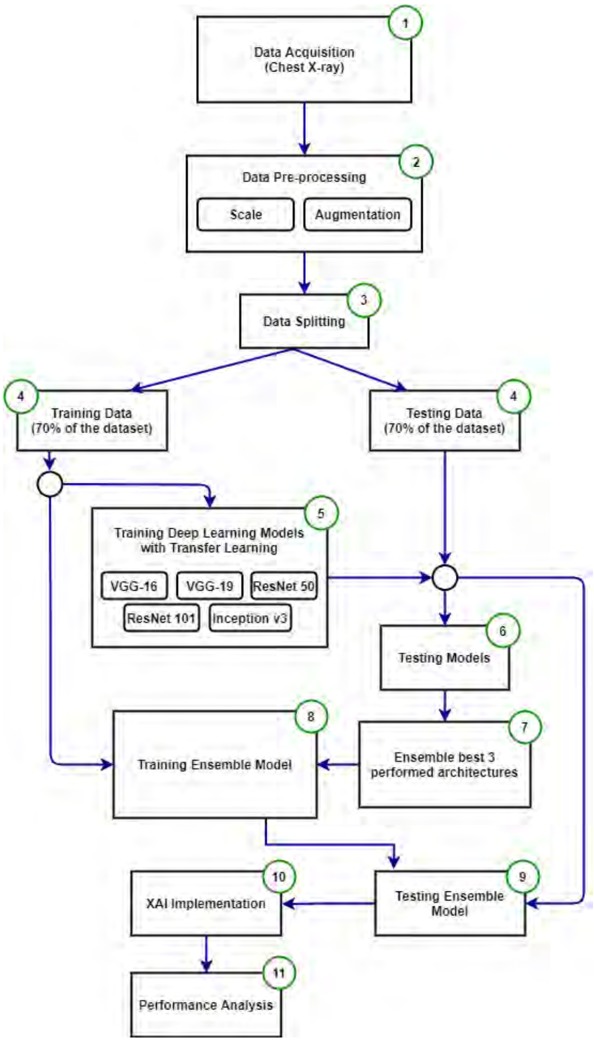


Figure 3.1: WorkFlow Diagram of the proposed PNEXAI Model

The workflow diagram in Figure 3.1 gives an overview of each step we have done to train and evaluate our models. Firstly, we collected and pre-processed our data which included data scaling and augmenting. Then, via the deep learning models (VGG16, VGG19, ResNet50, ResNet101 and Inception V3), we trained our data set. The dataset was subsequently verified and validated. We chose the three best architectures to join the ensemble in the following phase. The ensemble model was then trained and tested to find the accuracy. Lastly, Explainable AI (XAI) was implemented to analyze the ensemble model.

3.2 Used Architectures

In this study, We used five different CNN architectures named VGG-16, VGG-19, ResNet50, ResNet101, and Inception v3. Transfer learning has been used with ImageNet weights. We used these CNN models and compared the performance of them in terms of classifying the pneumonia of children patients. Then we ensembled the best three performing architectures and implementerd Explainable AI on that ensemble model.

3.2.1 VGG16

The VGG16 architecture contains about 16 convolution layers, as the name suggests. The default VGG16 architecture takes an image of shape 224x224x3 as input and provides a volume of 7x7x512 feature slices at the end of the convolutional layers. All the convolution blocks follow a common pattern: multiple stacked convolution layers followed by a max pool layer by the end of it. The originally proposed VGG16 architecture had 2 Fully Connected (FC) layers by the end of the convolution layers. The first FC hidden layer had 4096 FC neurons and the final output FC layer had 1000 FC neurons, each corresponding to one of all 1000 classes it had to classify [39].

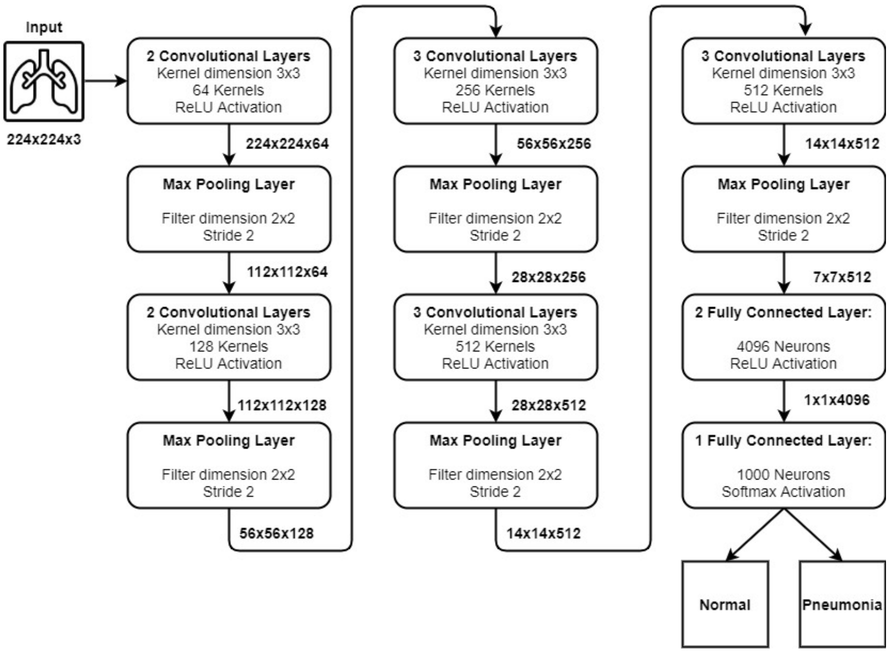


Figure 3.2: The detailed architecture of VGG16

3.2.2 VGG19

VGG 19 is the upgraded version of the VGG-16. It consists of 16 convolution layers and 3 fully connected layers[40]. This is a deeper CNN with more layers. To reduce the number of parameters in such deep networks, it uses small 3×3 filters in all convolutional layers and is best utilized with its 7.3% error rate. This Model is trained on Imagenet dataset and can classify images into more than 1000 objects [41]. The features of this deep CNN architecture are, it's input size is 224*224*3, size of convolution kernel is 3*3 with stride size of 1 pixel. For the spatial resolution preservation, spatial padding is used.2*2 pixel windows are used to perform max pooling with stride 2.

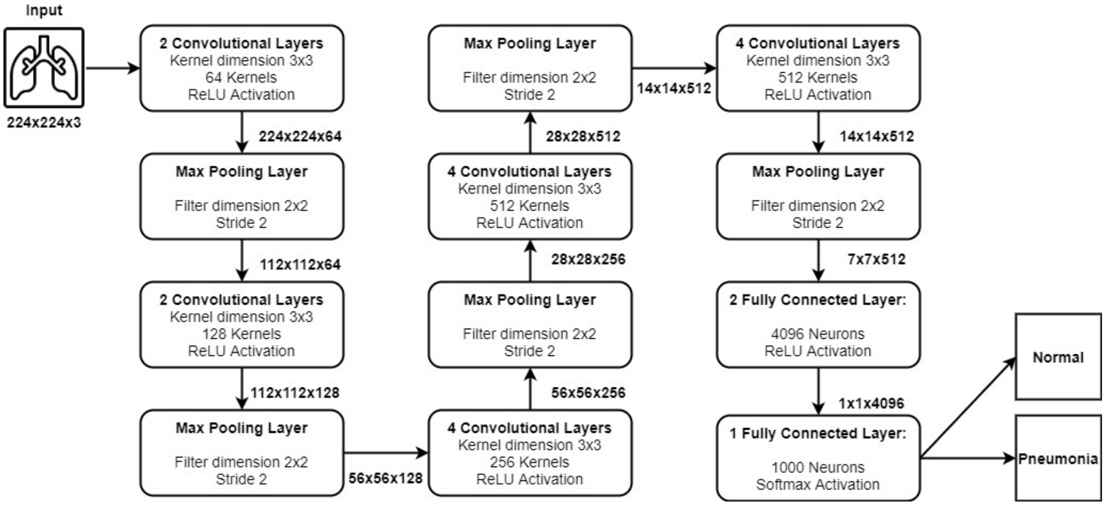


Figure 3.3: The detailed architecture of VGG19

3.2.3 ResNet50

ResNet [7] (2015) proposed the residual block with bypass layer, which allows the gradient to flow more easily, even with deeper layers. ResNet-50 has 25.5 million parameters across 49 convolution layers and one fully-connected layer. Each residual block element-wise adds the current feature map with the feature map from the previous residual block. There is also a bottleneck layer with 1*1convolution that shields a large number of channels for the more expensive 3*3 layer. The pre-trained ResNet 50 Py Torch model achieved a top-1 accuracy of 76% and a top-5 accuracy of 92.9% on ImageNet. Resnet-50 is the most popular version of the ResNet family balancing computational complexity and prediction accuracy.

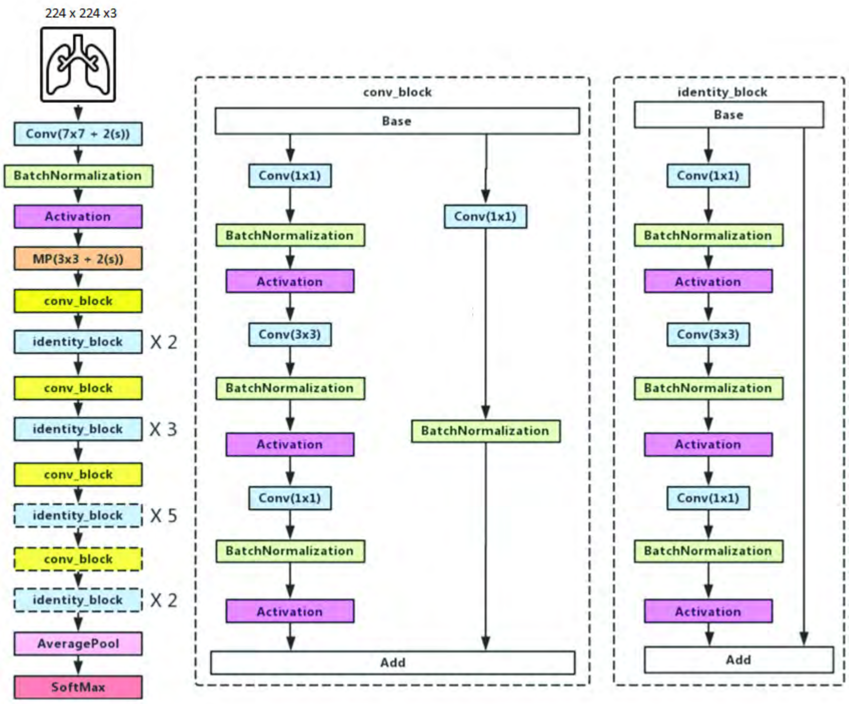


Figure 3.4: Internal Architecture of ResNet50

3.2.4 ResNet101

ResNet101 is a variant of the ResNet model which consists of 101 deep layers. A pretrained model is loaded and trained on Imagenet Dataset. By classifying images from 1000 object categories it has learned high feature representation [42]. This network take input size of 224*224*3.

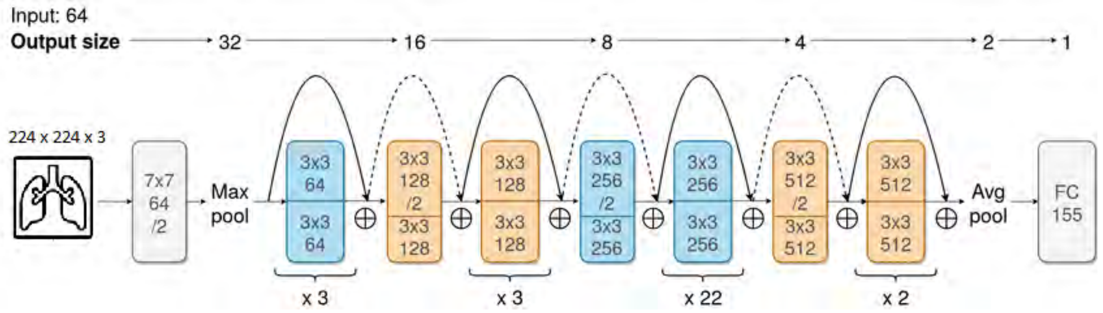


Figure 3.5: Internal Architecture of ResNet101

3.2.5 Inception V3

GoogleNet (Inception-V1) [43] (2014) is very parameter-efficient. It has 7 million parameters across 57 convolutional layers and only one fully connected layer. GoogleNet has nine inception modules. Each inception module consists of four branches with 1×1, 3×3, 5×5 convolutions and down-sampling. Two auxiliary loss layers inject loss from the intermediate layers and prevent gradient vanishing. At inference time, the auxiliary layers can be removed.

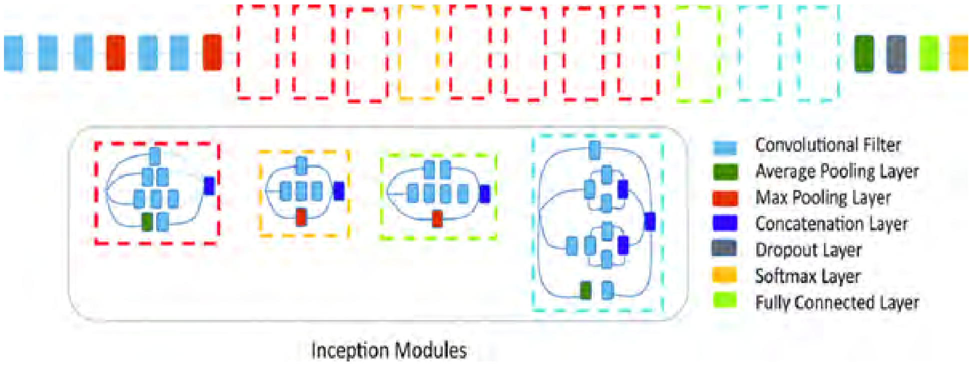


Figure 3.6: Internal Architecture of Inception v3

The GoogleNet Caffe model achieved a top-1 accuracy of 68.9% and a top-5 accuracy of 89.0% on ImageNet. We used the Inception-V3 model in our deep compression experiments. Compared with Inception-V1, the 5×5 convolutions are replaced with two 3×3 convolutions, separable kernels came into place, and batch normalization is added in Inception-V3. The pre-trained Inception-V3 PyTorch model achieved a top-1 accuracy of 77.45% and a top-5 accuracy of 93.6% on ImageNet.

3.3 Ensemble Modeling

Ensemble modeling is a process of multilayer diverse base models which are used to predict an outcome either by using many different modeling algorithms or using different training data sets. The aim of this modeling is to reduce the generalization error of a prediction and have the possibility of higher accuracy results than a single classifier.

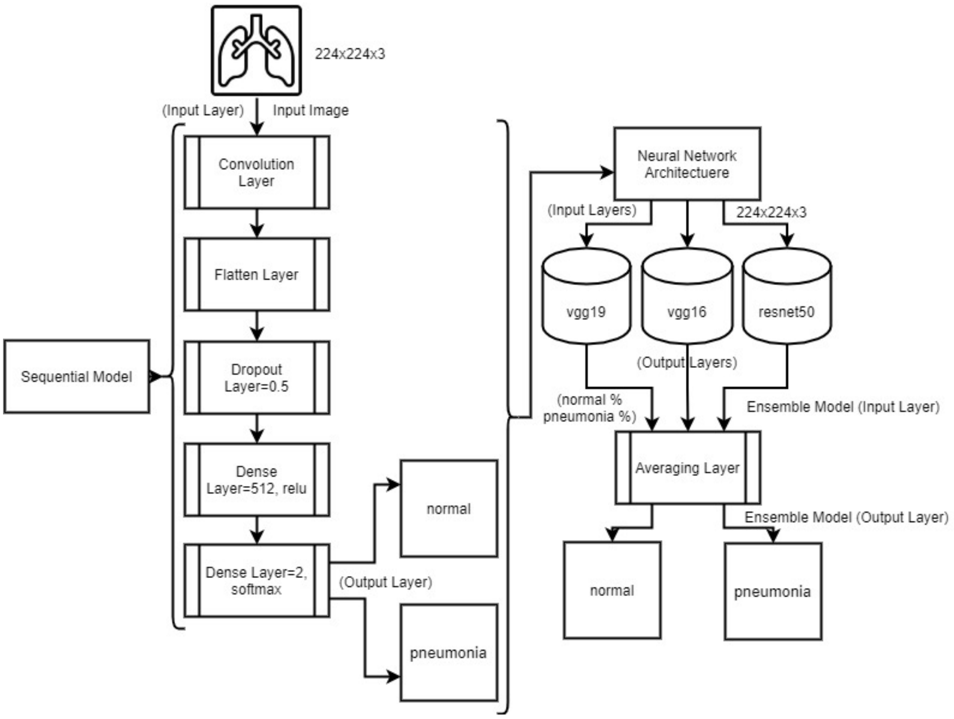


Figure 3.7: Construction of Ensemble model

In Ensembled model, multiple models are combined and act as a single model.In neural networks, average voting is a common ensemble method where averaging softmax class probabilities a posterior label is generated for all base learners. In each model the inputs and outputs remain same for averaging layer modification [44].

Averaging Layer

In the proposed model, the output probability of VGG16, VGG19 and ResNet50 will be taken as input for averaging layers and it will generate an average value for two labels , one is Pneumonia positive and another is pneumonia negative.

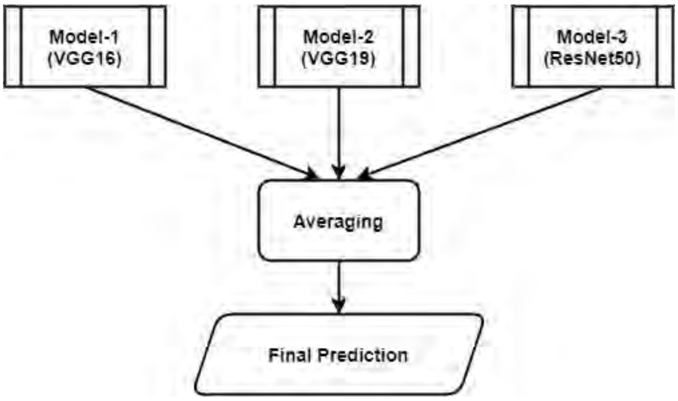


Figure 3.8: Output layers of the ensemble model hierarchy for PNEXAI

Output Layer

Based on the result of the averaging layer pneumonia patients can be identified from the chest x-ray images. If the probability of the first index is greater then the second index, the patient is pneumonia negative, otherwise it is pneumonia positive.

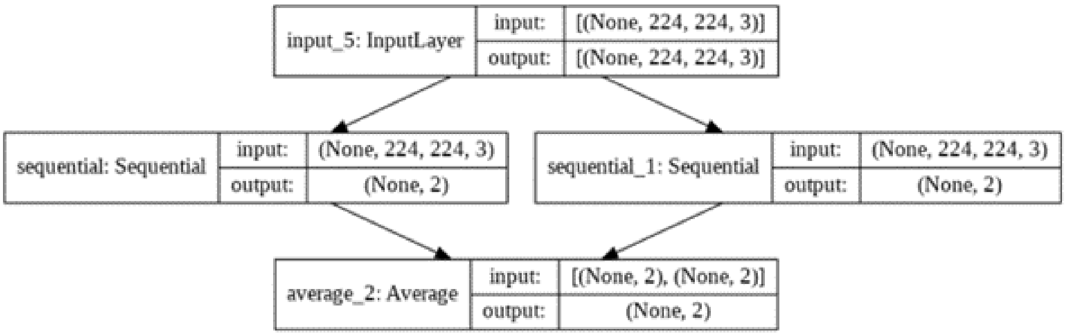


Figure 3.9: Output layers of the ensemble model hierarchy for PNEXAI

3.4 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence is a technique where a more explainable model is generated with a high level of learning performance [45]. By this technique human can easily understand ,trust and manage the emerging generation of intelligent partners. XAI allows supervised care by adapting deep learning techniques to obtain explainable attributes. It also contains procedures for acquiring more hierarchical, generalizable, and explanatory representations, and other pattern inference tools for asserting an understandable paradigm with any black -box testing model. Client happiness, influencing the multispectral images goal attainment, credibility analysis, and consistency are all areas where XAI thrives. In other terms, Explainable AI (XAI) is artificial intelligence that permits learners to acknowledge the consequences of the procedure. It is a collection of tools and regulatory frameworks that help individuals in comprehending and deciphering machine learning model projections. Feature attributions for model predictions in Auto ML Tables and AI Platform, and visually investigate model behavior using the What-If Tool can also be generated by the help of XAI.

Generally two types of techniques are used for explainable AI systems, these are ante-hoc and post-hoc techniques [46]. Ante-hoc techniques are applied in the AI models from the start of the implementations. Two most used ante-hoc techniques are Reverse Time Attention Model (RETAIN) and Bayesian Deep Learning (BDL). Post-hoc techniques involve explainability during testing stages. The training stages are carried out normally. Post-hoc model analysis is a very common approach towards explaining AI in production.Local interpretable Model-Agnostic Explanations (LIME) and Black Box Explanation through Transparent Approximation (BETA) are two types of Post-hoc techniques. In our research, we used the LIME technique.

Local interpretable Model-Agnostic Explanations (LIME)

LIME is a popular post-hoc method that learns an interpretable model and attempts to explain its prediction. Only after a decision has been made will it provide an explanation. Here is how it works. LIME receives input, then generates a new dataset composed of refined data samples. The next step involves populating corresponding predictions that would have been made by a black-box model if the aforementioned samples would have been used as input.

Next is the training of an interpretable model. The model is trained on the new dataset to help explain changes in the key extracted features.

The concept comes from a work [47] in which the authors tamper with the original data points, input them into a black box model, and then evaluate the outputs. The algorithm then weighs the new data points based on their distance from the original position. Finally, it uses those sample weights to train a surrogate model on the dataset, such as linear regression. The newly trained explanation model can then be used to explain each of the original data points.

More precisely, the explanation for a data point x is the model g that minimizes the locality-aware loss $L(f, g, \pi_x)$ measuring how unfaithful g approximates the model to be explained f in its vicinity π_x while keeping the model complexity denoted low.

$$argmin_x L(f, g, \pi_x) + \omega(g) \tag{3.1}$$

As a result, model integrity and complexity are traded off in LIME.. LIME is most important for AI systems. To trust the AI system, Models must be explainable to users. AI interpretability reveals what's going on inside these systems and aids in the detection of potential problems including information leakage, model bias, robustness, and causality. LIME provides a generic framework for uncovering black boxes and explaining why AI-generated predictions or recommendations are made.

Chapter 4

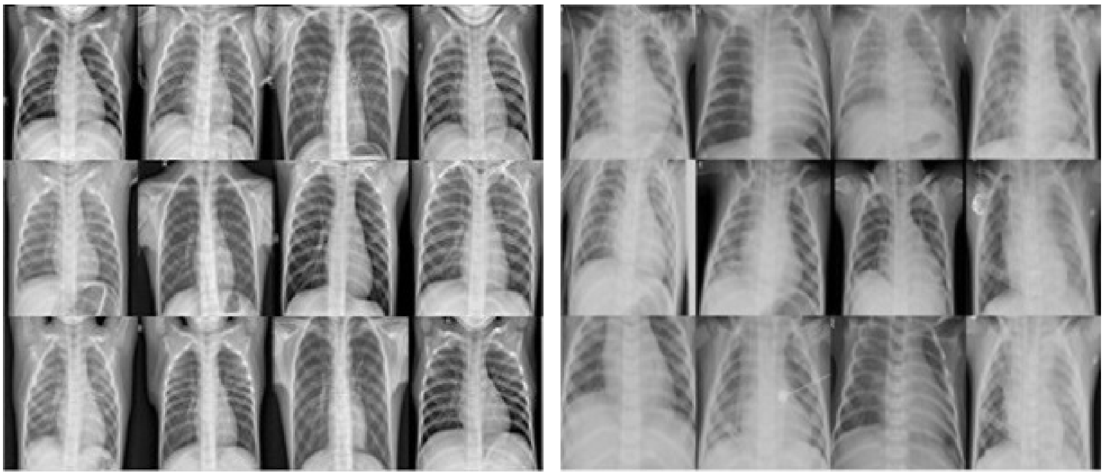
Implementation

4.1 Dataset

In the research work, we have used a publicly available chest X-ray dataset which was proposed by Kermany et al. [6] The dataset was collected from the Guangzhou Women and Children's Medical Center. It contains chest x-ray images of children who are between 0 to 5 years old. The database contains total 5,842 X-ray images that are of two different classes: normal and pneumonia.

4.1.1 Data Sample

One of the main symptoms of pneumonia in chest x-ray images is, the alveoli get filled with secretion and appear as a white portion on the chest radiograph. Figure 4.1(a) shows the normal and figure 4.1(b) shows pneumonia affected X-ray images obtained from the dataset.



(a) Normal Images

(b) Pneumonia affected images

Figure 4.1: Sample Data of the Dataset

4.1.2 Data Classification

We classified the Train and test images into 8:2 in our study. We have taken 4,263 images as training set where 3,198 Pneumonia affected images and 1,065 Normal Images. On the other hand, We have taken 1,828 images as training set where 1,279 Pneumonia affected images and 549 Normal Images. Table 4.1 represents the distribution of our dataset.

	Train Set	**Test Set**	**Total**
Pneumonia	3,876	390	4,266
Normal	1,342	234	1,576
Total	5,218	624	5,842

Table 4.1: Distribution of our Dataset

Training Set

The stage during which labeled example data with the responses or output labels are given to the machine learning algorithm process.

Testing Set

In certain instances, as the algorithm iterates to enhance performance, a series of examples is used for real-world research, it may learn special characteristics of the training set. For an unseen test collection, good results will improve confidence that the algorithm can have right answers in the real world.

4.2 Data Pre-processing

4.2.1 Image Resize

The aim of our model is to detect pneumonia from chest x-ray images with better accuracy. To do so, We have used Chest x-ray images of pneumonia patients of both pneumonia positive and negative images.Some CNN models are trained with this data and for this training process it was necessary to resize the image according to the model requirements. As VGG,ResNet and Inception models receive 224*224 size images, we have resized our input images into 224*224 shape.

For this resizing process we have used some python frameworks such as TensorFlow [48], Scikit Image [49] and Cafffe [50]. To convert the image data into the pixel values ImageDataGenerator class of Keras has been used.This class accumulates our image dataset during generation, verification, or assessment, then restores photographs to the algorithm in batches and scales as needed.During the modeling of neural networks, this provides a robust and logical technique to scale visual data. The Image Data Generator can deal with a range of feature selection methods as well as pixel scaling options based on percentages. This class allows a reference to leveling because it mostly employs the mean determined from the training dataset as feature-wise centering. Statistics must be calibrated before regression on the training sample.

4.2.2 Normalization and Scaling Images

Normalization is the process of reducing data redundancy and removing less important image information. We have used PCA technique for this normalization. PCA or Principal Component Analysis is a method by which a large data variable is converted into a small data variable with most of the information [51]. It generates Eigen flat fields and merged them to normalize the Chest X-ray image projection.The systematic errors of projection intensity normalization are reduced by using dynamic flat fields [52] [53]. We have done this task by using Keras ImageDataGenerator class.

Normalization techniques reduce data to a scale of 0-1 by converting re-scale input to a ratio that can be multiplied by each pixel.

The data set contains chest x-ray images in .jpg format. Before training the CNN models with the data, the pixel values of all the images were converted between 0 and 1 through min-max scaling.

4.2.3 Data Augmentation

We applied some data augmentation on the images. Generally, it is advised not to make big modifications to medical image datasets as the images should represent the actual data as closely as possible. As a result, the amount of augmentation was kept as limited as possible. As chest x-ray images are nearly symmetrical from the horizontal view, we applied x-axis flip on the x-ray images. Furthermore, we varied the brightness of the images just slightly. All the data augmentation was only done on the training dataset so that better training can be achieved. The test set was kept as it was. Although, both the train and the test set went through scaling. Last but not least, all the training and test images were resized into 224x224x3 resolution in order to fit the pre-existing CNN architectures.

4.3 Architecture Training

Modern convolutional neural networks such as VGG, ResNet, or Inception, would be able to perform classification task with an accuracy over 99%. But these models are deep and complex. So, they are hard to train, and a very large number of images are necessary to train these networks without overfitting. To improve classification performance with small dataset Transfer learning is the best choice. It's technically very easy to implement. We have used five different CNN models, VGG16, VGG19, Inception V3, ResNet101, and ResNet50 integrating transfer learning. These five Models were trained on the large ImageNet dataset. We have downloaded these pretrained models with the weights resulting from the training on ImageNet. We took the convolution blocks of all the five neural network models and dropped the classification layer portion. We used our own classification layer. This classifier has only two output neurons in the last layer, one for pneumonia affected patients and another for non pneumonia patients. We used cross-validation during training and therefore, the pattern of train and test set for each epoch was different. Finally, all the resultant outputs were compared, analyzed and discussed.

4.4 Hardware specification

We ran this dataset on our system, which had the following configuration settings are CPU: Intel Core i5 8500, RAM: 16GB, and GPU: NVIDIA GTX2070 Super 8GB. This configuration is good enough to process this number of images.

4.5 Explainable AI Implementation

We used LIME based explainable AI framework to mark the positive, negative and interruption regions of the images. Using LIME architecture will provide us a better understanding about the ‘black box’ neural network classification.

Chapter 5

Result and Analysis

In the result and analysis of our study, the confusion matrix and the performance information such as Validation accuracy, recall, precision and F1 score were computed for each model. These were evaluated as the performance measures.

5.1 Individual Architecture

The performance of this research is explained using training curve and validation curve. Along with it, Confusion Matrix is used to analyze the performance of each model with some performance measures like validation accuracy,recall, precision and F1 score .

5.1.1 Performance Analysis with Learning Curve(s)

VGG16

During the training period through the customised VGG16 model, we generated the training curves that included the training accuracy and the loss curve. However, we could observe from the figure 5.1 that the training accuracy curve was maintaining a constant reach close to 0.99 and it's having an increasing slope between epoch 32 and 33. On the other hand, the loss curve was maintaining a constant reach close to 0.01 and it's having a decreasing slope between epoch 32 and 33.

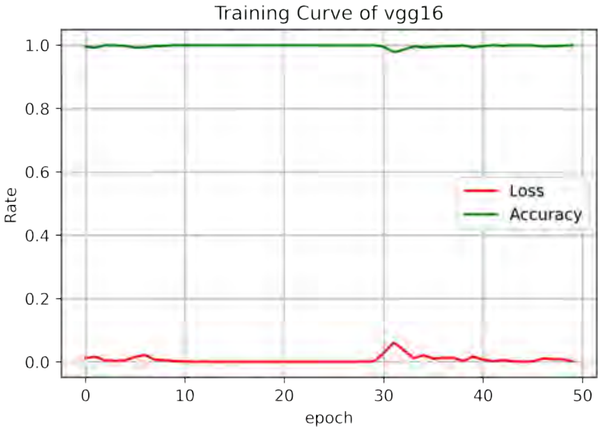


Figure 5.1: Training Curve(s) of VGG16

During the validation period through the customised VGG16 model, we generated the validation curves for making a comparison with the training curves that included the validation accuracy curve and the validating loss rate curve. However, we could observe from the figure 5.2 that the validation accuracy curve was maintaining a constant reach close to 0.97 and it's having increasing slope between multiple epochs. On the other hand, the validating loss rate curve was maintaining a constant reach close to 0.09 and it's having a decreasing slope between multiple epochs.

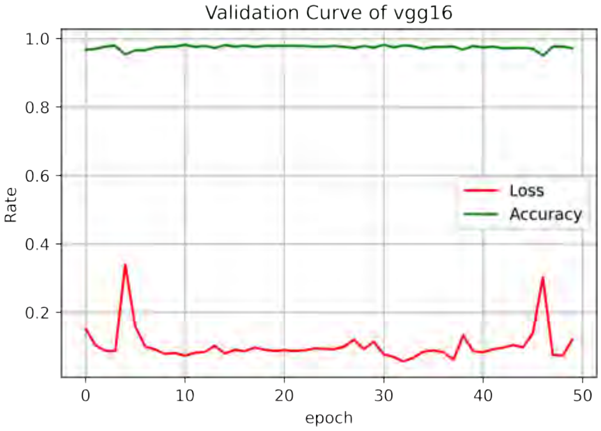


Figure 5.2: Validation Curve(s) of VGG16

VGG19

We obtained the training curves, that included learning accuracy curve and the loss rate curve, all across the training period using the customized VGG19 model. The training accuracy curve, on the other hand, maintained a constant value approximately at 0.99 and had a rising slope between epoch 4 and 5, as seen in the figure 5.3. The loss rate curve, just from the other hand, maintained a constant value approximately at 0.01 and had decreasing slope between multiple epochs.

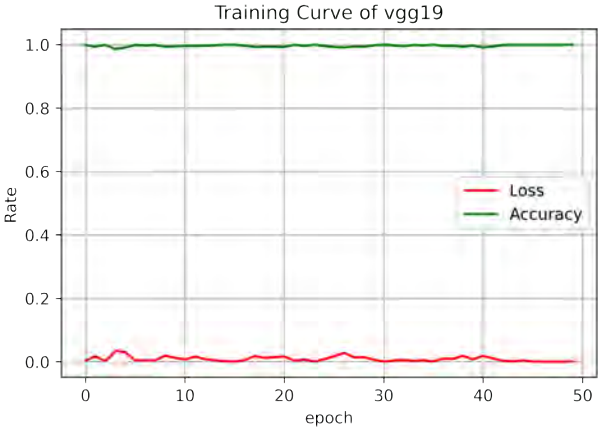


Figure 5.3: Training Curve(s) of VGG19

We created the validation curves for comparison with the training curves during calibration process using the customized VGG19 model, which included the validation accuracy curve and the validating loss rate curve. The validation accuracy curve, on the other hand, maintained a constant value approximately at 0.97 and had a rising slope throughout several epochs, as seen in the figure 5.4. The validating loss rate graph, on the other hand, maintained a constant value at 0.1 and had a decreasing slope throughout several epochs.

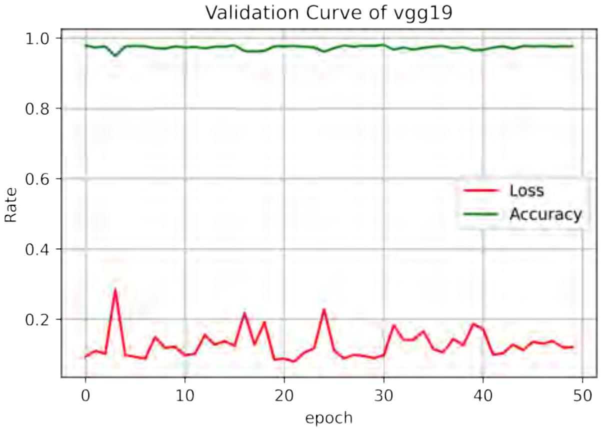


Figure 5.4: Validation Curve(s) of VGG19

ResNet50

The training curves, which included the training accuracy curve and the loss rate curve, were created throughout the training period using the customized resnet-50 model. The training accuracy curve, on the other hand, was keeping a steady value at 0.99 and had a rising slope between successive epochs, as seen in the figure 5.5. The loss rate curve, on the other hand, was keeping a constant value at 0.01 and had a decreasing slope between successive epochs.

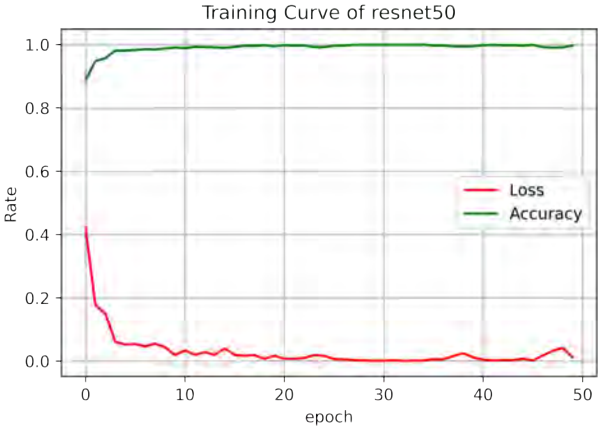


Figure 5.5: Training Curve(s) of ResNet50

The validation curves, which included the validation accuracy curve and the validating loss rate curve, were created during the validation period using the customized resnet-50 model for comparison with the training curves. The validation accuracy curve, on the other hand, maintained a consistent value at 0.96 and had a rising slope throughout several epochs, as seen in the figure 5.6. The validating loss rate curve, on the other hand, has a decreasing slope between successive epochs and maintains a constant value close to 0.01-0.05.

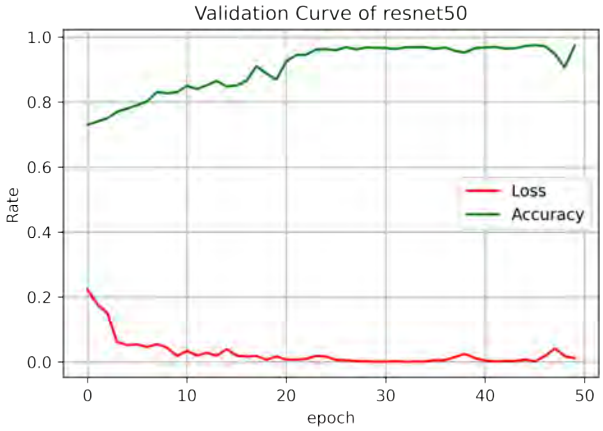


Figure 5.6: Validation Curve(s) of ResNet50

ResNet101

Throughout the training period, the modified resnet-101 model was used to construct the training curves, which contained the training accuracy curve and the loss rate curve. In contrast, the training accuracy curve remained constant at 0.99 and exhibited an increasing slope between successive epochs, as seen in the figure 5.7. Between successive epochs, however, the loss rate curve maintained a constant value of 0.01 and exhibited a decreasing slope.

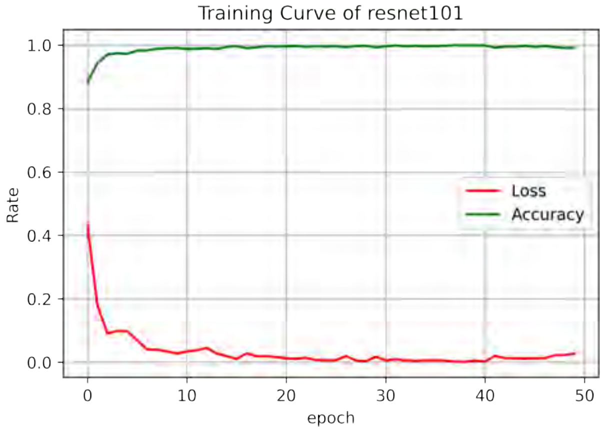


Figure 5.7: Training Curve(s) of ResNet101

During the validation phase, the validation accuracy curve and the validating loss rate curve were produced for comparison with the training curves using the modified resnet-101 model. The validation accuracy curve, on the other hand, stayed around 0.96 over multiple epochs and had an increasing slope, as seen in the figure 5.8. The validating loss rate curve, on the other hand, shows a decreasing slope between epochs and stays near to 0.09.

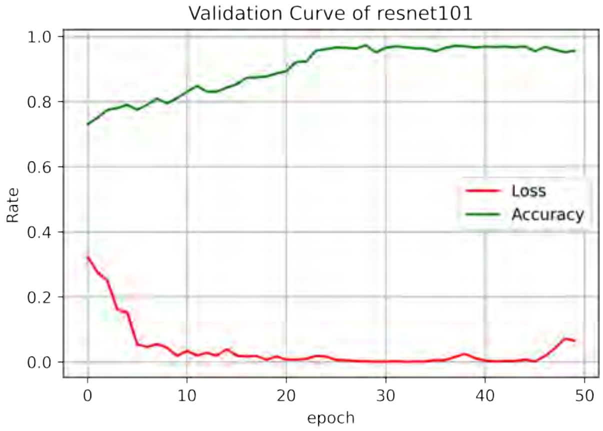


Figure 5.8: Validation Curve(s) of ResNet101

Inception v3

Using the modified inception-v3 model, we acquired the training curves, which included the learning accuracy curve and the loss rate curve, during the training period. The training accuracy curve, on the other hand, remained roughly constant at 0.99 and had an increasing slope between epochs 1 and 12, as seen in the figure 5.9. On the other hand, the loss rate curve maintained a constant value of around 0.01 and had a decreasing slope throughout several epochs.

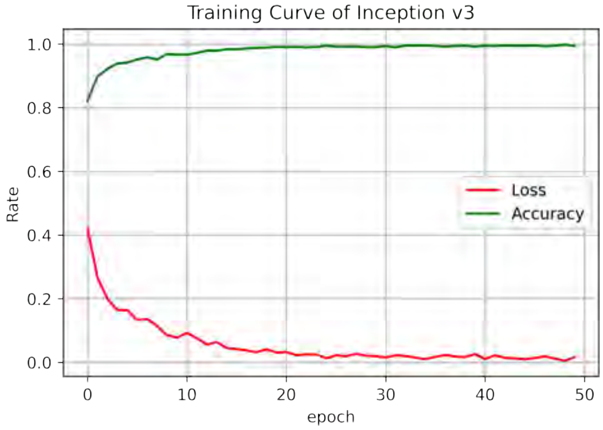


Figure 5.9: Training Curve(s) of Inception v3

We used the modified inception-v3 model to develop the validation curves for evaluation with the training curves during the calibration procedure, that included validation accuracy curve and the validating loss rate curve. The validation accuracy curve, on the other hand, remained constant approximately at 0.95 over multiple periods of history had an increasing slope, as seen in the figure 5.10. The validation loss rate graph, on the other hand, remained constant at 0.1 over multiple epochs and exhibited a declining slope.

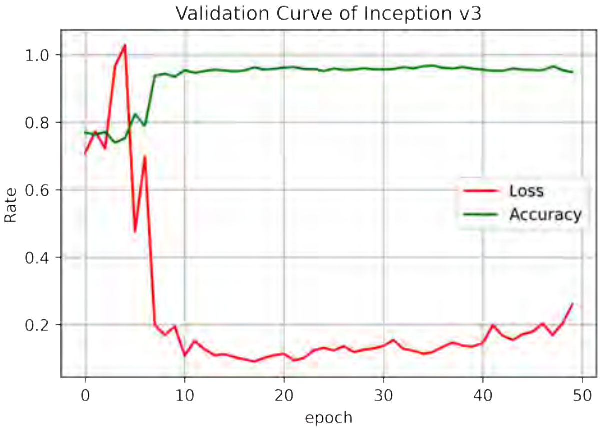


Figure 5.10: Validation Curve(s) of Inception v3

5.1.2 Confusion Matrix

The confusion matrix is an array that contains correct and incorrect predictions of the algorithm and the actual situation [7]. In this research, we have used confusion matrix technique to summarize the performances of VGG16, VGG19, ResNet50, ResNet101, and Inception V3.

VGG16

In figure 5.11, the confusion matrix for VGG16 is shown where 825 out of 853 pneumonia affected images are classified as pneumonia.

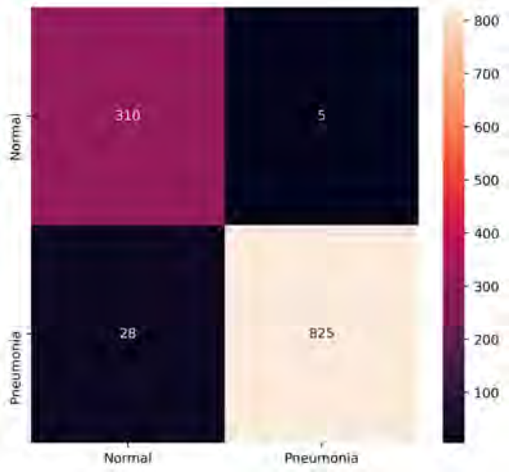


Figure 5.11: Confusion Matrix of VGG16

On the other hand, 28 images were classified as non pneumonia which was an error. Again, 310 images of normal chest X-rays are classified correctly and 5 images got the wrong classification.

VGG19

In figure 5.12, The confusion matrix of VGG-19 reveals that it can successfully identify 838 pneumonia affected images from 1168 chest x-ray images. It could not identify 15 pneumonia positive images. Additionally from 315 normal images, 303 images were identified as normal and the rest of the 12 images identified as pneumonia positive.

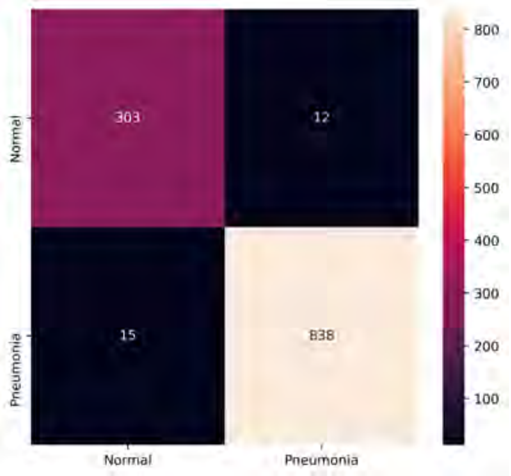


Figure 5.12: Confusion Matrix of VGG19

ResNet50

Figure 5.13 presents the confusion matrix of ResNet-50. Out of 853 pneumonia affected images, 832 images were correctly classified and 21 images got the wrong classification.

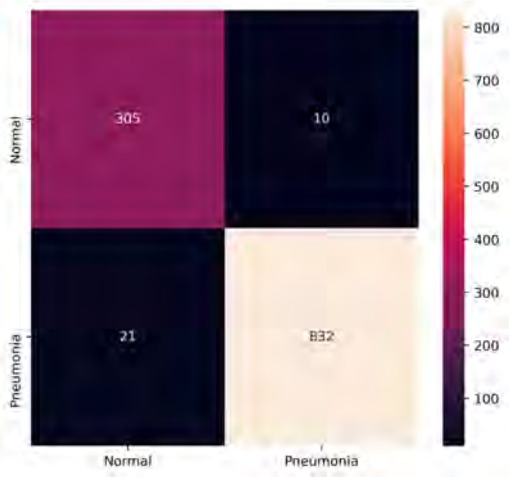


Figure 5.13: Confusion Matrix of ResNet50

On the flip side, out of 315 normal chest images, 305 images were classified correctly while 10 images got the wrong classification.

ResNet101

From the confusion matrix figure 5.4 ,it is clear that ResNet-101 can detect 306 out of 315 normal images . On the opposite side, 811 out of 853 pneumonia affected images are identified as pneumonia positive. It got a wrong prediction for 9 normal images and 42 pneumonia affected images.

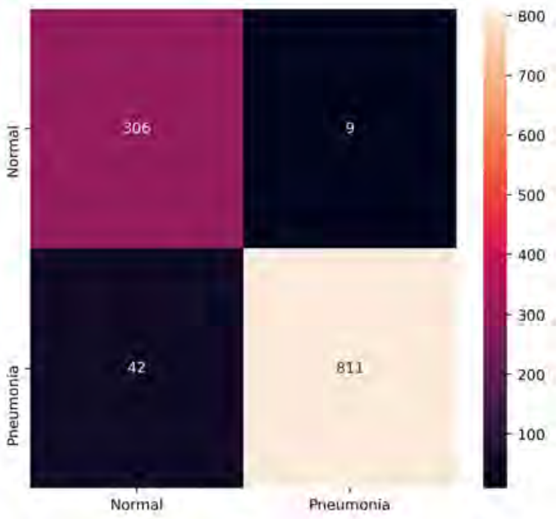


Figure 5.14: Confusion Matrix of ResNet101

Inception v3

Figure 5.5 shows the confusion matrix for Inception v3 where 802 out of 853 pneumonia affected images were classified as pneumonia, where 51 images were classified as normal which was the false classification.

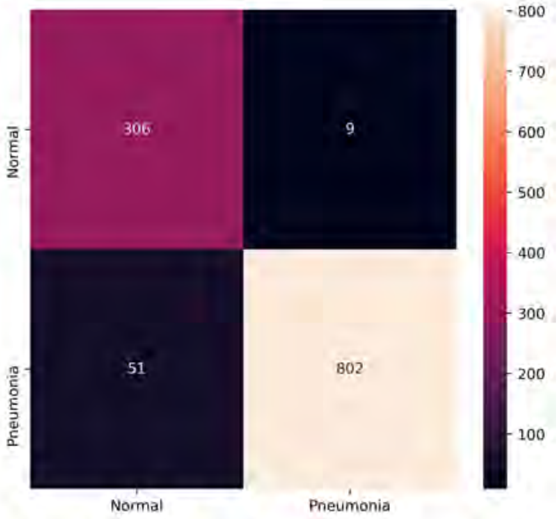


Figure 5.15: Confusion Matrix of Inception v3

In contrast, out of 315 normal chest images, 306 images were correctly classified and 9 images got the wrong classification.

5.1.3 Comparison Analysis

We achieved the accuracy rate of 97.17% by VGG16, 97.69% by VGG19, 97.35% by ResNet50, 95.63% by ResNet101, and 94.86% by Inception V3, respectively.
From the comparison of table 5.1, we can determine that VGG19, VGG16 and ResNet50 shows us 97.17%, 97.69% and 97.35%. So, these are the best three performed architecture. Therefore, we are considering VGG19, VGG16 and ResNet50

Architecture	**Accuracy**	**Precision**	**Recall**	**f1-score**
VGG16	97.17%	95.56%	97.57%	96.49%
VGG19	97.69%	96.94%	97.22%	97.07%
ResNet50	97.35%	96.19%	97.18%	96.67%
ResNet101	95.63%	93.42%	96.11%	94.63%
Inception v3	94.86%	92.30%	95.58%	93.73%

Table 5.1: Comparison between our used models

for the next level implementation.

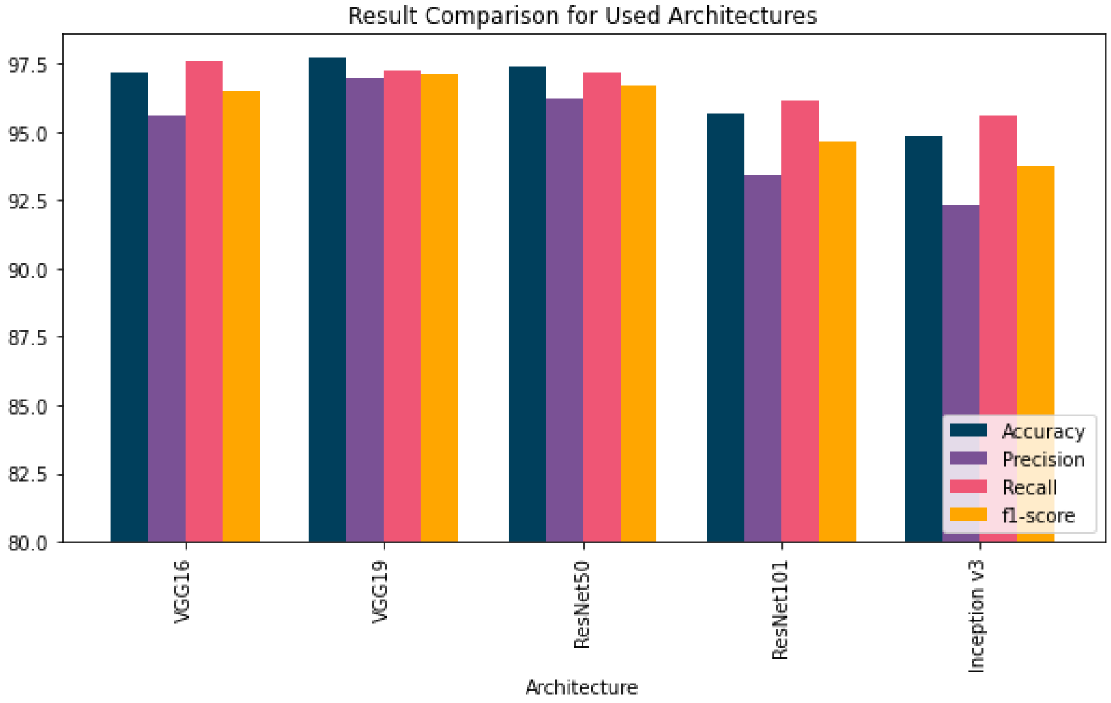


Figure 5.16: Illustration of Comparison Among the Used Architectures Using a Bar-Chart

5.2 PNEXAI (Ensemble of VGG16, VGG19 and ResNet50)

In our ”PNEXAI” model, we combined our three best performed architecture which are VGG16, VGG19 and ResNet50 for ensemble modeling.

5.2.1 Performance Analysis with Learning Curve(s)

Using the improved PNEXAI model, we obtained the training curves, which included the learning accuracy curve and the loss rate curve, throughout the training period.

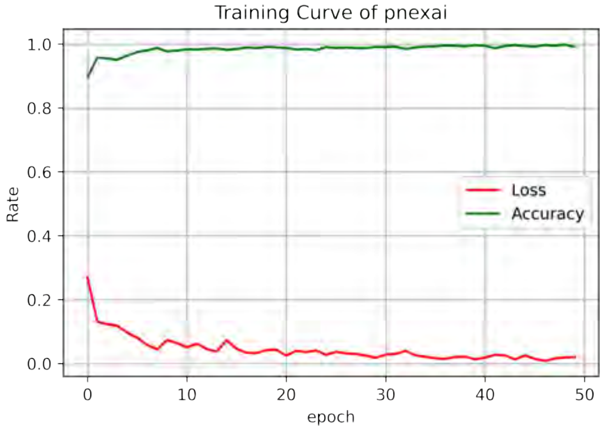


Figure 5.17: Training Curve(s) of PNEXAI

The training accuracy curve, on the other hand, stayed approximately constant at 0.99 and had an increasing slope between successive epochs, as seen in the graph. The loss rate curve, on the other hand, remained constant at about 0.01 and exhibited a decreasing slope across multiple epochs.

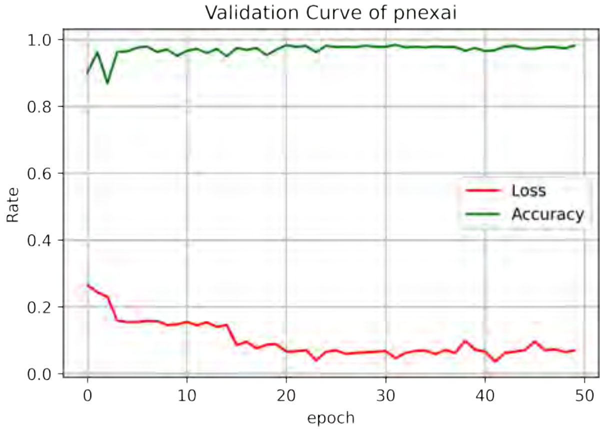


Figure 5.18: Validation curve(s) of PNEXAI

The validation curves for assessment with the training curves during the calibration method were developed using the modified PNEXAI model, which comprised the validation accuracy curve and the validating loss rate curve. The validation accuracy curve, on the other hand, had an increasing slope and stayed roughly constant at 0.97 across numerous periods of history, as seen in the graph. The validation loss rate graph, on the other hand, had a decreasing slope and stayed constant at 0.06-0.1 throughout several epochs.

5.2.2 Confusion Matrix of PNEXAI

Figure 5.19 shows the proposed PNEXAI model's confusion matrix in which 840 of the 853 images affected by pneumonia were categorized as pneumonia, 13 of which were listed as normal, which is false. In contrast, 310 images were classified correctly from the 315 normal chest pictures, and 5 were classified incorrectly.

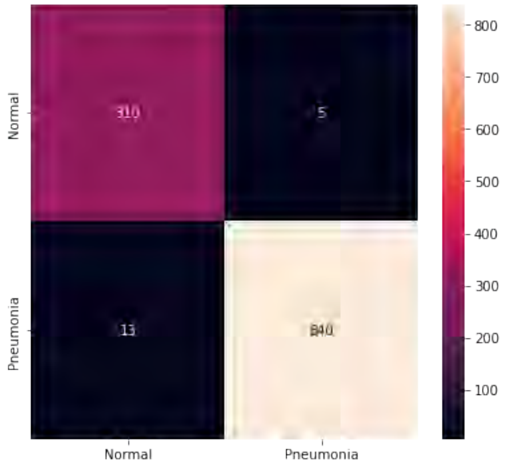


Figure 5.19: Confusion Matrix of PNEXAI

5.2.3 Result Analysis of PNEXAI

Our improved PNEXAI model achieved an accuracy of 98.46% with precision of 98.48%, recall of 99.41% and f1-score of 98.94%. We can see the illustration of this result in figure 5.20.

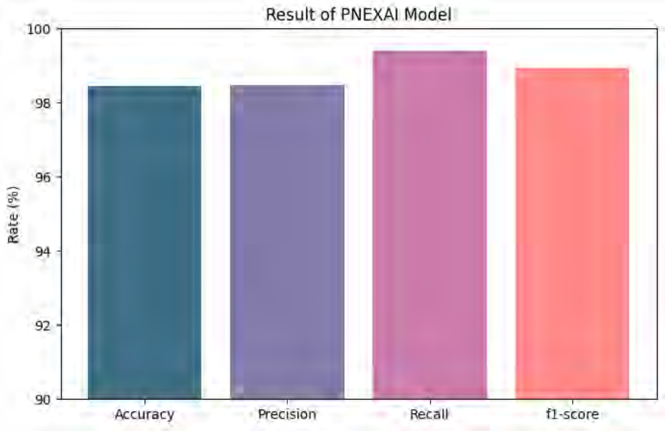


Figure 5.20: Illustration of the Result of PNEXAI Model

5.3 Explainable Artificial Intelligence (XAI)

After implementing Explainable AI on "PNEXAI", we have noticed that there are 3 types of masks represent by 3 different colors Yellow, Red and Green. Here, Yellow borders represent the interpretable regions, Green mask represents the Non infected responding regions and Red mask represents the infected responding regions.

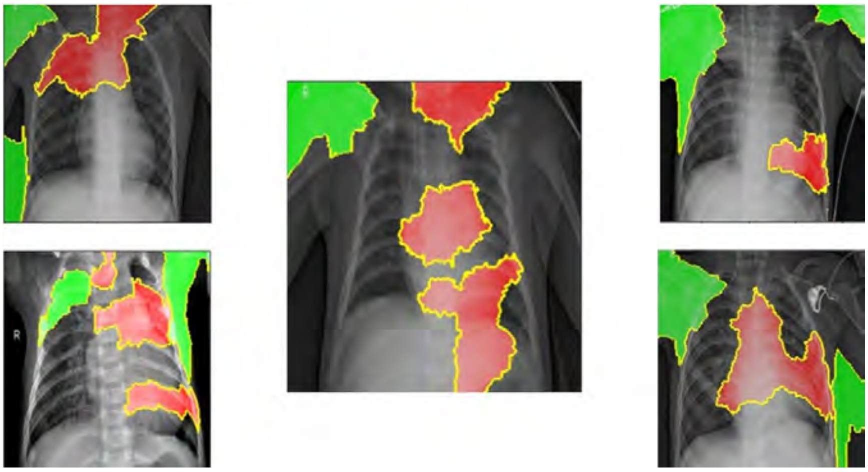


Figure 5.21: Output of XAI for Pneumonia cases

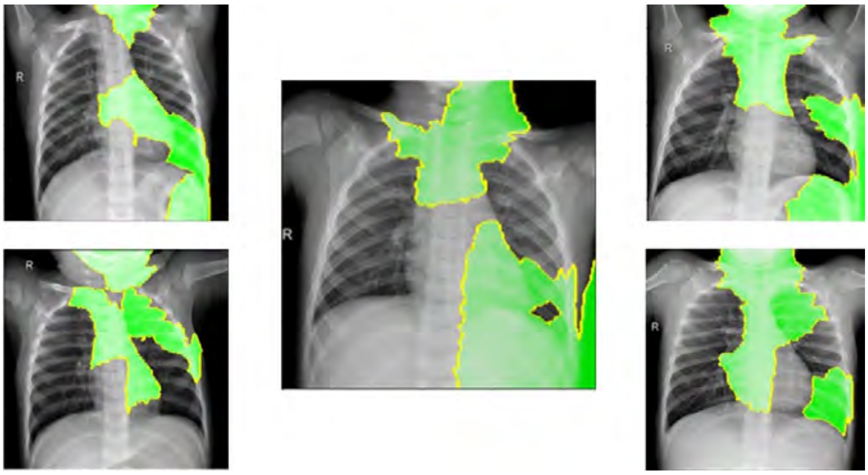


Figure 5.22: Output of XAI for Normal cases

We can see that from figure 5.4 and 5.5 that Pneumonia infected region responded mostly right portion of the lungs.

Chapter 6

Conclusion and Future Work

6.1 Conclusion and Future Work

It is vital to have faster medical monitoring in order to diagnose Pneumonia faster. The major purpose of this study is to create a computer-aided diagnostic system that can aid in the identification of pneumonia and hence prevent unfavorable consequences (such as mortality). In this research, we can see that deep learning models can identify pneumonia very effectively. Chest X-ray images are used in the developed model PNEXAI. First, we trained our dataset using VGG16, VGG19, ResNet50, Resnet101, and Inception V3 which obtained 97.17% , 97.69%, 97.35%, 95.63%, and 94.86% accuracy respectively. Next, we ensembled the best three performed models (VGG16, VGG19 and ResNet50) and achieved 98.46% overall accuracy. For the identification of the affected regions and better understanding of the classification , Explainable AI (XAI) is applied on PNEXAI model.

We can gather more chest x-ray images in the future to enhance the dataset, which could improve pneumonia detection accuracy and correctly identify the pneumonia-affected regions of the patient. By doing so, an effective method for detecting pneumonia will be discovered.

Bibliography

[1] R. E. Black, S. Cousens, H. L. Johnson, J. E. Lawn, I. Rudan, D. G. Bassani, P. Jha, H. Campbell, C. F. Walker, R. Cibulskis, *et al.*, "Global, regional, and national causes of child mortality in 2008: A systematic analysis," *The lancet*, vol. 375, no. 9730, pp. 1969–1987, 2010.

[2] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong, "An efficient deep learning approach to pneumonia classification in healthcare," *Journal of healthcare engineering*, vol. 2019, 2019.

[3] S. Naicker, J. Plange-Rhule, R. C. Tutt, and J. B. Eastwood, "Shortage of healthcare workers in developing countries–africa," *Ethnicity & disease*, vol. 19, no. 1, p. 60, 2009.

[4] A. A. Saraiva, N. M. F. Ferreira, L. L. de Sousa, N. J. C. Costa, J. V. M. Sousa, D. Santos, A. Valente, and S. Soares, "Classification of images of childhood pneumonia using convolutional neural networks.," in *BIOIMAGING*, 2019, pp. 112–119.

[5] J. K. Rajaratnam, J. R. Marcus, A. D. Flaxman, H. Wang, A. Levin-Rector, L. Dwyer, M. Costa, A. D. Lopez, and C. J. Murray, "Neonatal, postneonatal, childhood, and under-5 mortality for 187 countries, 1970–2010: A systematic analysis of progress towards millennium development goal 4," *The Lancet*, vol. 375, no. 9730, pp. 1988–2008, 2010.

[6] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[7] A. Rizal, R. Hidayat, and H. A. Nugroho, "Entropy measurement as features extraction in automatic lung sound classification," in *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, IEEE, 2017, pp. 93–97.

[8] M. B. Rodrigues, R. V. M. Da Nobrega, S. S. A. Alves, P. P. Reboucas Filho, J. B. F. Duarte, A. K. Sangaiah, and V. H. C. De Albuquerque, "Health of things algorithms for malignancy level classification of lung nodules," *IEEE Access*, vol. 6, pp. 18 592–18 601, 2018.

[9] H. Liz, M. Sánchez-Montañés, A. Tagarro, S. Domınguez-Rodrıguez, R. Dagan, and D. Camacho, "Ensembles of convolutional neural network models for pediatric pneumonia diagnosis," *Future Generation Computer Systems*, vol. 122, pp. 220–233, 2021.

[10] H. Ren, A. B. Wong, W. Lian, W. Cheng, Y. Zhang, J. He, Q. Liu, J. Yang, C. J. Zhang, K. Wu, *et al.*, “Interpretable pneumonia detection by combining deep learning and explainable models with multisource data,” *IEEE Access*, vol. 9, pp. 95 872–95 883, 2021.

[11] Q. Ye, J. Xia, and G. Yang, “Explainable ai for covid-19 ct classifiers: An initial comparison study,” *arXiv preprint arXiv:2104.14506*, 2021.

[12] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, “The ensemble deep learning model for novel covid-19 on ct images,” *Applied Soft Computing*, vol. 98, p. 106 885, 2021.

[13] V. Sharma, S. Chhatwal, B. Singh, *et al.*, “An explainable artificial intelligence based prospective framework for covid-19 risk prediction,” *medRxiv*, 2021.

[14] A. K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, and A. Chakrabarti, “Automatic covid-19 detection from x-ray images using ensemble learning with convolutional neural network,” *Pattern Analysis and Applications*, pp. 1–14, 2021.

[15] J. V. S. das Chagas, D. d. A. Rodrigues, R. F. Ivo, M. M. Hassan, V. H. C. de Albuquerque, and P. P. Rebouças Filho, “A new approach for the detection of pneumonia in children using cxr images based on an real-time iot system,” *Journal of Real-Time Image Processing*, pp. 1–16, 2021.

[16] L. Račić, T. Popović, S. Šandi, *et al.*, “Pneumonia detection using deep learning based on convolutional neural network,” in *2021 25th International Conference on Information Technology (IT)*, IEEE, 2021, pp. 1–4.

[17] R. Kundu, R. Das, Z. W. Geem, G.-T. Han, and R. Sarkar, “Pneumonia detection in chest x-ray images using an ensemble of deep learning models,” *Plos one*, vol. 16, no. 9, e0256630, 2021.

[18] M. F. Hashmi, S. Katiyar, A. G. Keskar, N. D. Bokde, and Z. W. Geem, “Efficient pneumonia detection in chest xray images using deep transfer learning,” *Diagnostics*, vol. 10, no. 6, p. 417, 2020.

[19] D. Kim and T. MacKinnon, “Artificial intelligence in fracture detection: Transfer learning from deep convolutional neural networks,” *Clinical radiology*, vol. 73, no. 5, pp. 439–445, 2018.

[20] J. Bernal, K. Kushibar, D. S. Asfaw, S. Valverde, A. Oliver, R. Martı, and X. Lladó, “Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: A review,” *Artificial intelligence in medicine*, vol. 95, pp. 64–81, 2019.

[21] D. J. Hemanth, C. K. S. Vijila, A. I. Selvakumar, and J. Anitha, “Performance improved iteration-free artificial neural networks for abnormal magnetic resonance brain image classification,” *Neurocomputing*, vol. 130, pp. 98–107, 2014.

[22] K. R. Mopuri, U. Garg, and R. V. Babu, “Cnn fixations: An unraveling approach to visualize the discriminative image regions,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2116–2125, 2018.

[23] M. Zhang, W. Li, and Q. Du, “Diverse region-based cnn for hyperspectral image classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2623–2634, 2018.

[24] H. Fu, B. Wu, and Y. Shao, “Multi-feature-based bilinear cnn for single image dehazing,” *IEEE Access*, vol. 7, pp. 74 316–74 326, 2019.

[25] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, “Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification,” *arXiv preprint arXiv:1801.09927*, 2018.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[27] Y. Xu, Z. Jia, Y. Ai, F. Zhang, M. Lai, I. Eric, and C. Chang, “Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, pp. 947–951.

[28] X. Bi, S. Li, B. Xiao, Y. Li, G. Wang, and X. Ma, “Computer aided alzheimer’s disease diagnosis by an unsupervised deep learning technology,” *Neurocomputing*, vol. 392, pp. 296–304, 2020.

[29] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

[30] M. Edwards and X. Xie, “Graph based convolutional neural network,” *arXiv preprint arXiv:1609.08965*, 2016.

[31] S. H. Khan, M. Hayat, and F. Porikli, “Regularization of deep neural networks with spectral dropout,” *Neural Networks*, vol. 110, pp. 82–90, 2019.

[32] Z. Polkowski, J. Vora, S. Tanwar, S. Tyagi, P. K. Singh, and Y. Singh, “Machine learning-based software effort estimation: An analysis,” in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2019, pp. 1–6.

[33] T. Gonzalez, “Imagenet classification with deep convolutional neural networks,” *Handbook of Approximation Algorithms and Metaheuristics*, pp. 1–1432, 2007.

[34] M. Cicero, A. Bilbily, E. Colak, T. Dowdell, B. Gray, K. Perampaladas, and J. Barfett, “Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs,” *Investigative radiology*, vol. 52, no. 5, pp. 281–287, 2017.

[35] A. Kukkar, R. Mohana, A. Nayyar, J. Kim, B.-G. Kang, and N. Chilamkurti, “A novel deep-learning-based bug severity classification technique using convolutional neural networks and random forest with boosting,” *Sensors*, vol. 19, no. 13, p. 2964, 2019.

[36] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[37] C. Fan and H. Hauser, “Fast and accurate cnn-based brushing in scatterplots,” in *Computer Graphics Forum*, Wiley Online Library, vol. 37, 2018, pp. 111–120.

[38] K. Men, H. Geng, C. Cheng, H. Zhong, M. Huang, Y. Fan, J. P. Plastaras, A. Lin, and Y. Xiao, “More accurate and efficient segmentation of organs-at-risk in radiotherapy with convolutional neural networks cascades,” *Medical physics*, vol. 46, no. 1, pp. 286–292, 2019.

[39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[40] Y. Zheng, C. Yang, and A. Merkulov, “Breast cancer screening using convolutional neural network and follow-up digital mammography,” in *Computational Imaging III*, International Society for Optics and Photonics, vol. 10669, 2018, p. 1 066 905.

[41] J. Xiao, J. Wang, S. Cao, and B. Li, “Application of a novel and improved vgg-19 network in the detection of workers wearing masks,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1518, 2020, p. 012 041.

[42] S. Tao, Y. Guo, C. Zhu, H. Chen, Y. Zhang, J. Yang, and J. Liu, “Highly efficient follicular segmentation in thyroid cytopathological whole slide image,” in *International Workshop on Health Intelligence*, Springer, 2019, pp. 149–157.

[43] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019.

[44] A. Yazdizadeh, Z. Patterson, and B. Farooq, “Ensemble convolutional neural networks for mode inference in smartphone travel survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2232–2239, 2019.

[45] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, “Xai—explainable artificial intelligence,” *Science Robotics*, vol. 4, no. 37, eaay7120, 2019.

[46] *Explainable ai - an introduction*. [Online]. Available: https://www.section.io/engineering-education/explainable-ai/.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin, “” why should i trust you?” explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[48] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[51] Z. Jaadi, *A step-by-step explanation of principal component analysis (pca).* [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[52] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[53] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.