PREDICTION OF LUNG DISEASE(PNEUMONIA) USING DEEP LEARNING

A.T.V SATYA SAI ROHIT

16951A05M4

PREDICTION OF LUNG DISEASE(PNEUMONIA) USING DEEP LEARNING

A Project Report

Submitted in partial fulfillment of the

requirements for the award of the degree of

Bachelor of Technology

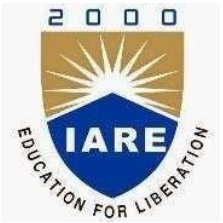
in

Computer Science & Engineering

by

A.T.V Satya Sai Rohit

16951A05M4



Department of Computer Science & Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500 043, Telangana

April, 2020

DECLARATION

I certify that

a. the work contained in this report is original and has been done by me under the guidance of my supervisor(s).
b. the work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute in preparing the report.
d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
e. whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place : Signature of the Student

Date : Roll No.

CERTIFICATE

This is to certify that the project report entitled **Prediction of Lung Disease (Pneumonia) Using Deep Learning** submitted by **Mr. A.T.V Satya Sai Rohit** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **Computer Science & Engineering** is a bonafide record of work carried out by him/her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor **Head of the Department**

Date

APPROVAL SHEET

This project report entitled **Prediction of Lung Disease (Pneumonia) Using Deep Learning** by **A.T.V Satya Sai Rohit** is approved for the award of the Degree Bachelor of Technology in **Computer Science & Engineering**

Examiners **Supervisor (s)**

Principal

Date:

Place:

ACKNOWLEDGEMENT

The completion of this project work reflects the efforts and time I had spent which at last gave me immense satisfaction and some best results. I would like to take this opportunity to thank Ms. B. Padmaja mam who constantly encouraged and pushed me to get more perfect and accurate outcomes in the work I have been doing from a very long time. I would also like to thank my team members for their contribution in this work. I would also like to thank college management for providing the necessary infrastructure and facilities. Finally, I would like to thank one and all who directly or indirectly helped me and supported me during the course of this work.

ABSTRACT

Keywords : Transfer Learning, Convolutional neural network, VGG 16, convolution, max pooling.

Medical diagnosis is one of the fields in which deep neural network has a large scope. Deep neural networks can be used for higher efficiency in terms of image classification. Pneumonia is a severe infectious disease which can be predicted using chest x-rays by radiologists. Using deep learning we can develop an automated prediction system by building a convolutional network which can help us in predicting the disease. In deep learning framework, transfer learning technique is one of the methods based on the image net classifications. In transfer learning, we will use VGG 16 using Keras which is a type of convolution neural network used to build our model which can predict pneumonia disease . In transfer learning basically the data sets are previously classified by various groups which could typically be used in some other applications. Therefore the datasets can be easily adapted by the model we are building. Here the datasets we are using include several chest x-rays or radiographs which are grouped into normal ones and the one's with pneumonia disease.

Traditional approaches like building a neural network from scratch or other convolutional models will have their limitations. In this paper, we are evaluating the effectiveness of the application of Transfer learning VGG 16 in Pneumonia identification using the chest x-ray datasets.. The experimental results revealed that the proposed scheme is effective and shows improved performance over its predecessor.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHAPTER 1

Introduction

The computer and tech world has been evolving with some amazing improvements and transformations. One such stream which changed the course of many applications in varied fields of research and problems in machine learning. Machine learning found its application in almost every field which includes social media, digital marketing, medical, image classification, security and so on. Machine learning algorithms work well with processed data sets i.e pre-processes data sets which are structured in a good way as they can be used to train a model efficiently. But when it comes to unstructured data like images and videos, it is not much efficient. Therefore deep learning algorithms like neural networks have evolved and has some good performance. Deep learning algorithms basically include some networks which are trained continuously and which require a lot of computational power and time.

Gradually, deep learning algorithms can be used in the medical field. One such application is detecting lung disease like pneumonia using chest x-rays. Many deep learning algorithms can be used to perform this task like Multilayer extreme machine or online sequential extreme machine or other networks. However building CNN's from scratch has no use since there involve many parameters to choose which involves layers, filters, weights etc. So, here we are using a pre-trained network known as VGG 16 of transfer learning which provides state of the art algorithms which classifies different kinds of images. So, basically we are building a model by training it with several sets of images classified into pneumonia and normal ones. Finally, the model will be able to predict the occurrence of pneumonia disease from chest x-rays with better accuracy.

1.1 Existing System

- There were some existing systems which utilized some machine learning Extreme Learning Algorithms (ELM and OS-ELM) which is an artificial neural network which takes advantages of both deep learning and extreme learning machines but the accuracy obtained was not up to the mark.

- Even some deep learning techniques involved the building of CNN models from scratch which is not very helpful.

- Deep neural networks have demonstrated high levels of accuracy in the fields of image classification when transfer learning framework is used.

- Since we are using images, deep learning approach using any neural network would be helpful and efficient but varies in the levels of accuracy.

1.1.1 Disadvantages

1) Disadvantages of using ELM and OS-ELM :

- An ELM is basically a 2-layer neural net as you can see in Figure 1.1 in which the first layer is fixed and random, and the second layer is trained.

- An ELM is much faster to train, but cannot encode more than 1 layer of abstraction, so it cannot be built deep.

- Another problem is though you can train them in a faster way, you afford for it as it has a very slow evaluation. For most massive applications, evaluation pace and speed are way more important than speed of training. Testing and training accuracy is around 90% but it is not efficient.

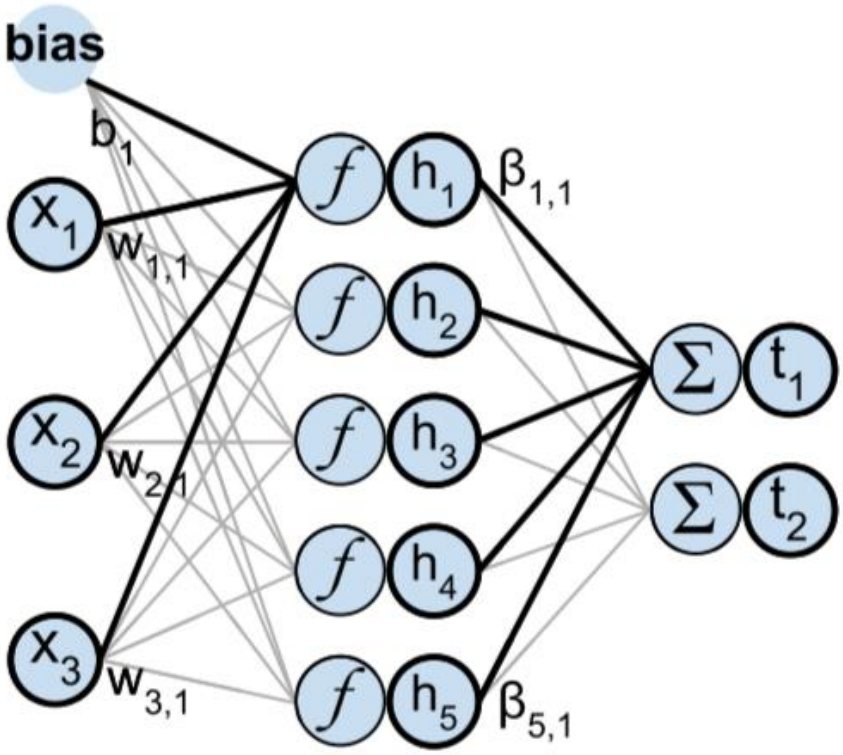


Figure 1.1 Extreme Learning Machines

2) Disadvantages of building a new CNN

- It is difficult to build a CNN from scratch because it requires a large amount of computing power and also a large amount of training data.

- Even if you are able to train them in a faster way, you face trouble in terms of slow evaluation. For most applications, evaluation speed is more important than training speed.

- The latter aspect in training a CNN turns out to be an obstacle in the case of medical image analysis. A basic CNN can be seen in Figure 1.2.

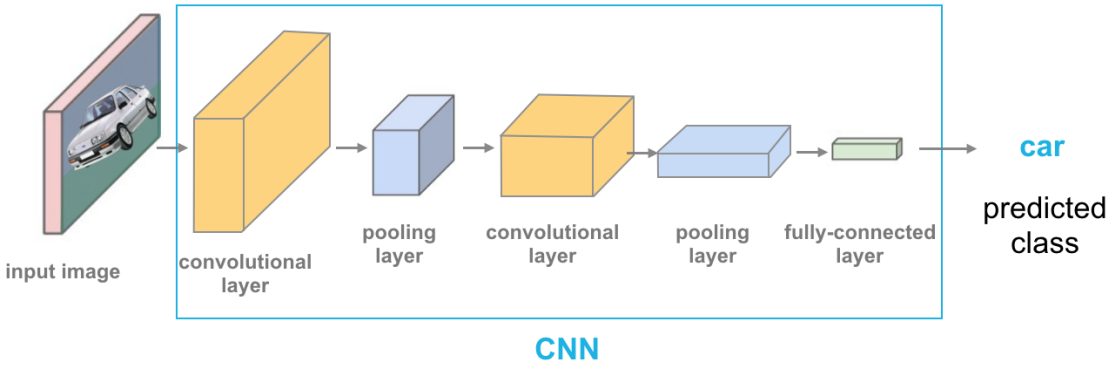


Figure 1.2 Basic Convolutional Neural Network

1.2 Proposed System

- The proposed system is based on the concept of utilizing a pre-trained network which has the potential to classify various images and consistently uses it for our application of building and classifying the set of chest x-rays

- To overcome the difficulties and problems faced by existing systems, we will be using a deep learning framework known as transfer learning.

- Deep neural networks have demonstrated high levels of accuracy in the fields of image classification when transfer learning framework is used.

- We will be using VGG 16 which is a pre-trained network for building our model

1.2.1 Advantages

- Pre-trained CNN's provide better performance and efficiency than training a CNN from scratch when provided with sufficient training.

- It is a state of the art algorithms, so we need not train weights in the backpropagation.

- If we build a model in this way, it can offer some good amount of accuracy ranges of greater than 90% based on the layers.

- Also it could be ready as fast as possible by just dropping the last layer and adding our own layers with the required number of classified outputs to our solution.

1.3 Project Scope

The scope of the project is very high because the future of health care is data-driven. These deep learning frameworks when trained accurately, provide some prominent results which play a vital role in the field of medical science. By training our model with chest x-rays we can predict the pneumonia disease in an effective way. There is a huge scope of Deep Learning in this sector in making the healthcare and treatments effective and improving the predictions. These kinds of predictions with improved accuracy will help the doctors in various tasks and decisions.

CHAPTER 2

Review of Literature

There are many applications in medical science which are being considered to be implemented using deep learning frameworks. Deep learning has a large scope in the field of medical science and radiology [2]. The analysis of chest radiography has a vital role in medical diagnosis and application [5]. Latest improvements in deep learning frameworks and models are facilitating algorithms to outperform medical experts in various medical imaging tasks such as skin cancer classification, diabetes detection, heart disease detection [6], lung disease detection. The diagnoses utilizing the chest radiographs and x-rays are finding some good popularity and preference. The analysis of chest radiography has a vital role in medical diagnosis and application [5]. These kinds of algorithms are being used to predict various lung diseases involving classification [7]. The performance of various convolutional models on varied factors relying on datasets which are publically available found that the similar deep convolutional network architecture differs in performance across abnormalities. Ensemble models improved various factors like classification, accuracy and finally, deep learning method improved accuracy when compared to normal methods like rule-based methods[12].

In various categories of learning algorithms and techniques, CNN proved to be one of the best algorithms for image classification, analysis etc [6]. Transfer learning is one such technique which mingles well with the applications of deep learning when large, multi-class data sets are available. Using standard architectures in transfer learning like VGG16 [8] we can explore the various pooling and flattening operations. And even more when pre-trained networks are utilized the results accuracy was found to be more promising[10]. Transfer learning is one such technique which mingles well with the applications of deep learning when large, multi-class data sets are available. However, these techniques don’t work on small data sets and fail abruptly. Our main aim here is to provide a novel ensemble approach based on transfer learning (VGG 16) in modelling our application and prediction of lung disease [11]-[15].

CHAPTER 3

Methodology

3.1 Classification

A method which is employed to extract valuable information from datasets is termed as Classification. This method divides the data into several categories. This division of data is based on some key features. The key point here is to develop a model that can sort out objects after training it on data objects where the category, or label, is known. The developed model should also be able to classify unlabeled data with good or sufficient accuracy. Many models were developed for solving classification problem and among them, one prominent example for classification model is neural networks.

3.2 Artificial neural networks

Machine learning is one of the fields in computer science which is based on the similar patterns of human learning methods. Artificial Neural Networks is a kind of machine learning techniques which tries to mimic the human brain. The ANN is a large network built on a series of interconnected neurons or links. The neurons take some input and produce output based on several factors like weights, activation functions etc. The typical structure of a neuron is shown in Figure 3.1. The ANN is constructed by utilizing these several numbers of neurons arranged in various layers and possessing different weighted connections with each other. All these layers work parallelly and in a sequential way and also backpropagation is involved if required. This network will continuously pass through many cycles until the weights are adjusted so that the expected output is obtained.

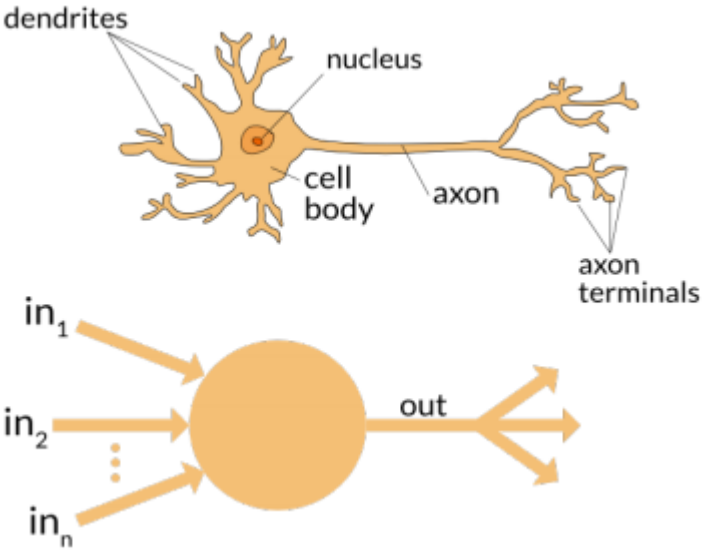


Figure 3.1 A comparision between a biological and an artificial neuron

This process of iterating and adjusting weights in a neural network is called backpropagation. The iterations continuously see up to it that the weights are adjusted by knowing the difference between the expected output and the present output obtained, basically reducing the loss of the difference occurred. In this way, the network works in iterations and adjustments to reach a perfect level of accuracy and classification.

3.2.1 Convolutional Neural Networks

Convolutional neural networks usually are used in the case of image classification. This particular neural network consists of a combination of convolutional and max-pooling layers. These layers are finally combined or attached to dense layers. The main operational characteristic of these convolutional networks is to extract several features from the images by using various kinds of filters of different sizes and combinations. A convolutional layer can consist of different filters which are applied in the same step and are capable of extracting different features out of it. These sizes of the filters depend upon the output features required to extract. After the convolution operation is performed, some activation functions and optimizers are applied to foster the outcome. The usual activation functions include ReLu and sigmoid function. We are utilizing ReLu function in our project. ReLu function is defined as $f(x) = \max(0, x)$, with the input thresholds to zero.

After a feature is extracted from the image, the spatial size of the image is preferred to be reduced. Basically, only specific intensity values are picked up. This process is known as subsampling which is typically done by a pooling layer. There are many kinds of pooling layers like average pooling, minimum pooling, max pooling. In our project, we are using max-pooling layers which are usually preferred a lot in convolutional neural networks. Here only high-intensity values are considered and the subsamples are formed. Figure 3.2 can demonstrate this process.

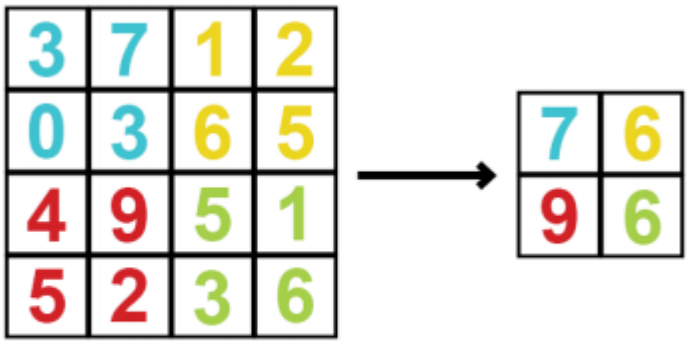


Figure 3.2 An example of max pooling

3.2.2 Training a network

We have already seen in ANN that CNN's are trained using a method known as backpropagation. The main aim is to adjust the weights correctly and for this step we require a huge amount of training data. These type of networks don't specify a required amount of data i.e there's no fixed limit whether how much data is required to train a network. If the difference within a class is too high, the number of training objects should be larger. The training process involves running or passing of data objects until the weights are adjusted. When the entire training data runs through the network, we say that one epoch has been completed. The underfitting or overfitting of the model depends mainly on the number of epochs and the number of iterations or steps.

A model shouldn't be overfitted i.e it should not get very much attached or adapted to the training data. If that happens the model may underperform in general cases. So it is better to prevent unbalanced classes in the dataset we consider. Another problem which can occur is underfitting in which the model doesn't fit well with the training data. Underfitting can cause overgeneralization problem by the model. We can prevent this by using regularization and some good optimizers.

3.2.3 Loss function

The important factor in the advancement of a neural network is the calculation of the loss function. It's the loss function that conveys us the cost of the loss occurred i.e the inaccuracy in our predictions. In order to achieve the desired output and to reduce the errors, we need to minimize the loss function. In case of these neural networks which are based on classification problems, we mostly prefer soft max cross-entropy functions given by :

$$H_{y'}(\mathrm{y}) = -\sum_i y_i' \log y_i \tag{3.1}$$

Where 'y' is the probability function and y' is the distribution which is true.

Gradient Descent is the most preferred optimization method to obtain minimal loss functions in machine learning algorithms.

3.3 Transfer learning

Transfer learning is a methodology in machine learning where one model developed for a particular task can be reused for another task. Pre-trained models are the core of many deep learning applications. In other words, transfer learning is like a model used for optimization for progress or improved performance while modelling another task. Here the knowledge gained from the previous task is transferred into the current one which makes the process easier and efficient. Transfer learning is basically applied to multi-task learning. Transfer learning is popular in deep learning when there is a need to train humongous and challenging data sets. Transfer learning is only useful in deep learning if the features of the model learned from the first task are general. Therefore transfer learning is basically an optimizer to save some unnecessary time on training. The transfer learning approach is very much appreciated in deep learning when there are datasets of images. One such competition is ImageNets in various networks are built which are capable of categorizing more than 1000 different objects. These are considered as the state of the art algorithms as they are balanced with the right amount of weights and optimizations. Every year this competition is held known as Image Net Classification. There will be different problem statements and new CNN's are designed to cope up with the solution which is able to classify 1000 different images and the one with higher accuracy will be given some prize money. Some of the algorithms that are designed for image classification with weights trained on Image Net are :

- Xception
- VGG 16 / VGG 19
- ResNet / ResNetV2 / ResNet
- InceptionV3 / Inception Resnet V2
- MobileNet / MobileNetV2
- DenseNet etc

In our model, we are utilizing a pre-trained network of VGG 16.

CHAPTER 4

System Design

4.1 Hardware Requirements

As we are dealing with deep learning which involves neural networks there is a need for some high-end hardware requirements. However, it is preferred to use GPU's to compute various levels of accuracy in different ways and to try on different sizes of datasets to compare the overall performance of the model. Some of the minimum hardware requirements needed are :

- Processor : Intel i3 or above
- Speed : 3 GHz or above (Quad-core is prefered)
- RAM : 8 GB or above
- Hard Disk : 40 GB or above

4.2 Software Requirements

4.2.1 Python Programming Language

Python is an integrated, object-oriented and high-level programming language. Python is a programming language with dynamic semantics. It is very easy to understand with no larger and complicated syntaxes. Python supports modularity of the program which helps in the reuse of the code. Python is copyrighted now and can be used under the GNU general public license and is maintained by a core team at the institute where it was developed. Python is often loved by programmers because it generally comes with a lot of features which include easy to learn, easy to read, easy to maintain, interactive, portable, extendable and scalable.

Python can be also used as a scripting language and it also provides very high-level data types and also supports high-level datatype checking. Python also supports automatic garbage collection and can be easily integrated with c, c++, ActiveX, COBRA, and java. English keywords are used in python and they are also reserved words. In the python language, the keywords are case sensitive. There are generally 33 keywords in python 3.7 and can vary in the due course of the time. Keywords in python are represented in lower case. Python integrates systems more efficiently.

4.2.2 TensorFlow

Tensor flow is an open-source library that helps to develop and train machine learning models. Generally, tensor flow is a math library and is used in machine learning which involves neural networks. This was primarily developed by google brain team for its internal use but later was released and made open source under Apache license. tensor flow can be operated on various CPU'S and GPU'S and is available on 64-bit Linux, Windows operating systems and are also available on mobile platforms such as android and IOS. The use of the tensor flow rapidly grew in developing both research and commerce applications so google has assigned various computer scientists on it to make it more robust and faster. The tensorflow.js version can be used for machine learning in javascript later the version 2.0 was released and now in the latest version, the computer graphics can be studied easily.

4.2.3 Keras

Keras is an open-source neural network library written in python language. deep neural networks can be fastly experimented using the Keras and also it is friendly, extensible and modular. This was primarily designed by the research team which involves Francois Chollet an engineer from google and he is also the one who maintains it now. Keras uses the model data structure to organize the layers. Generally, for simple architectures, we use a sequential model and for complex architectures, we use Keras functional API, which gives the functionality of building graphs of layers.

There are different types of Keras layers such as core layers, convolutional layers, pooling layers, recurrent layers, embedding layers, merge layers, normalization layers, and noise layers and all the layers have the most number of methods in common. Keras also contains optimizers, activations, callbacks, datasets, etc.

4.2.4 Software environment

There are many IDE's to choose which include Jupyter, anaconda or pycharm. These IDE's can offer various plugins and has all kinds of libraries support. Once can even prefer cloud platforms like Google Collab which offers GPU's with advance computing facilities on the cloud. In this project , we are going to use Google Collab cloud platform.

4.3 Architecture Diagram

The proposed system is built on the VGG 16 network which is a pre trained model based on the ImageNet Dataset. It consists of series of convolutional and max pooling layers combine with flatten , dense and fully connected layers which overall forms an efficient network which can classify 1000 different classes. VGG 16 is a Convolutional Neural Network which is a pre-trained model. It is based on the ImageNet Dataset. This model can classify 1000 different classes. It's accuracy when trained on ImageNet Dataset is more than 90%.It consists of 13 Convolutional layers, 5 Max pooling layers, Activation and Fully connected layers along with 3 dense layers.

As presented in Figure 4.1, it is evident that different components are provided in order to show the system to provide intended functionality. It has different series of operations involved.

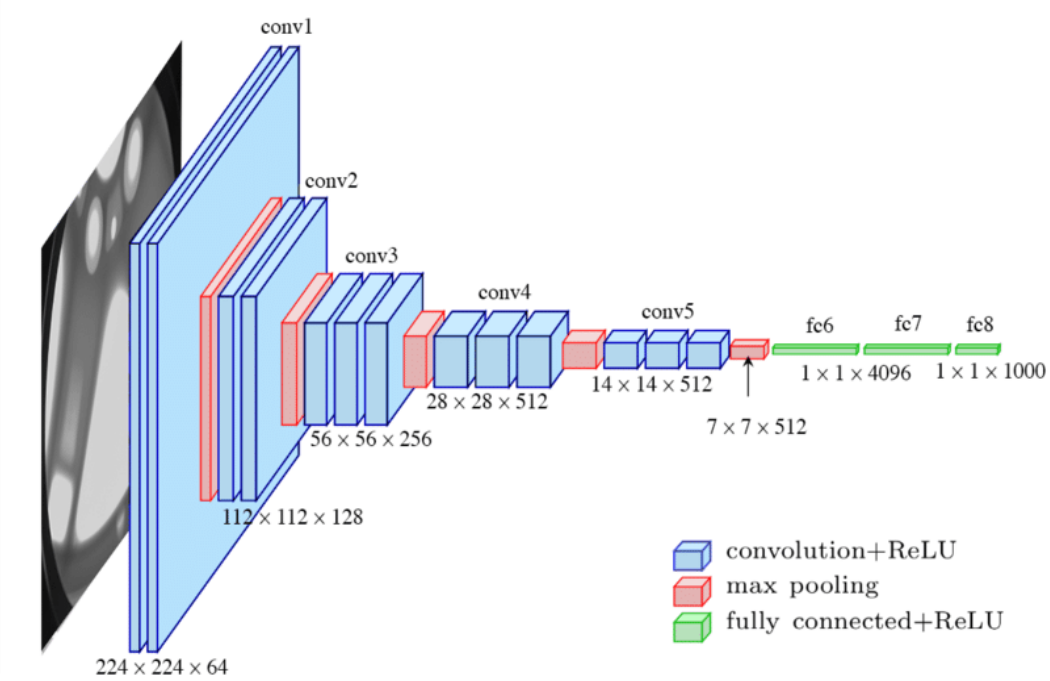


Figure 4.1 Architecture of VGG 16

4.4 Working Model

- We deployed Keras open-source deep learning framework with TensorFlow backend to build and train the convolutional neural network.
- Relu Activation Function is used in the convolutional operations
- Softmax Activation Function is used at dense and fully connected layers to fit the model accurately
- Adam Optimizer is used with different iterating learning rates to optimize the model
- The VGG 16 model consists of a combination of pre-trained weights, convolutions, poolings and fully connected layers
- **Convolution Operation:** Convolution operation basically uses different filters with different values to detect different aspects of the image like vertical edge detection, horizontal edge detection, face edges etc .A visual representation of this operation can be seen in Figure 4.2.

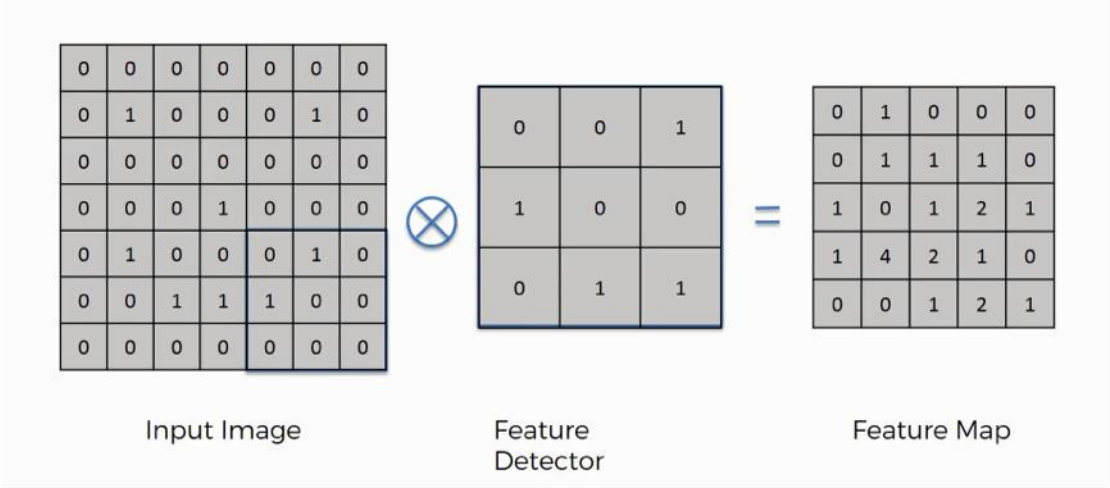


Figure 4.2 Convolutional Operation

- **Max Pooling Layer** : Max pooling layer is used to select the high-intensity values from the previously trained image. Max pooling operation visual representation can be seen in Figure 4.3

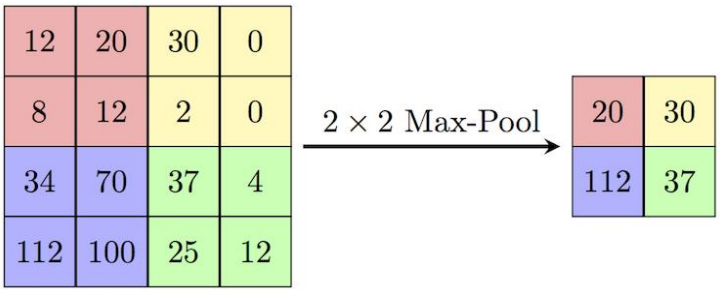


Figure 4.3 Max Pooling Operation

The stack of convolution and max-pooling layers is the core of the deep neural convolutional network. Finally, it is stacked with fully connected layers and dense layers. As presented in Table 4.1, there are different operations involved in a sequential manner which ultimately produces the required output result. The stack of convolutional and max-pooling layers is the core of the model.

Table 4.1 VGG 16 Working Description

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

4.5 Flow Chart

As shown in Figure 4.4, the flow chart is presented to know the functionalities of the proposed system in depth and with ease.

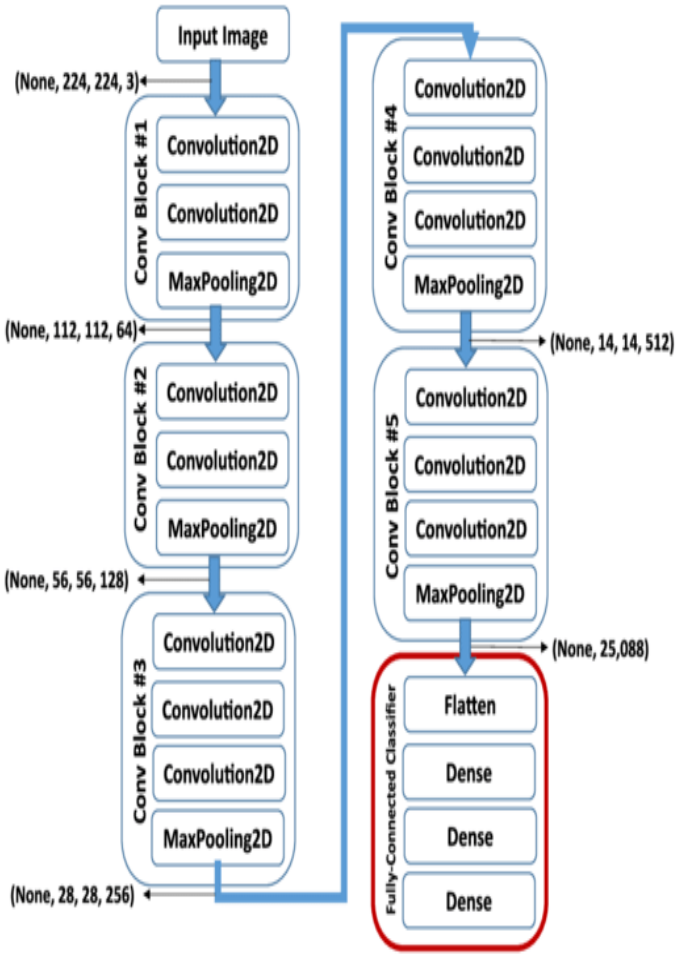


Figure 4.4 Flowchart for proposed system

4.6 Model Implementation

4.6.1 Network Parameters

1. Data Sets

The original dataset mainly consists of three folders they are - training, testing and validation. In addition to these it contains two subfolders which contain pneumonia (P) and normal (N) chest X-ray images, respectively. Approximately 5000 X-ray images of anterior-posterior chests were chosen from retrospective pediatric patients aged between 1 and 5 years old. The chest X-ray imaging was done as a part of patients' routine health checkup. In order to balance the data assigned to the training and validation set, the original data category is slightly tweaked to suit the requirements. The entire data is rearranged into only two sets they are the training set and validation set. About 3,722 images were assigned to the training set and 2,134 images were kept in the validation set. This addition of images to the validation set improved the validation accuracy. The sample images of both x-rays with and without pneumonia can be seen in Figure 4.4

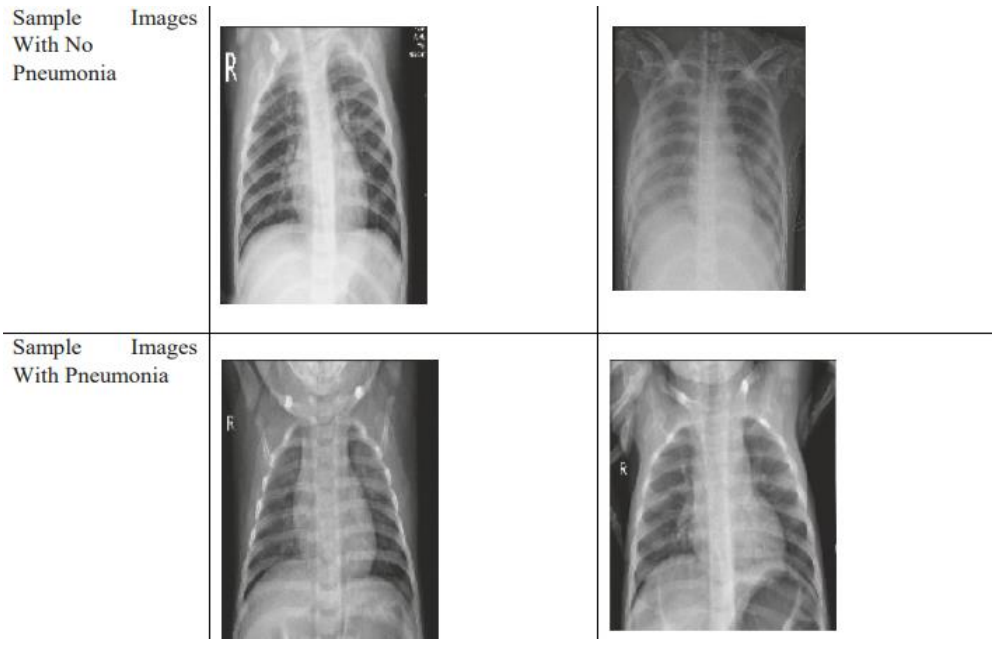


Figure 4.4 Sample images from dataset

2. **Importing all required libraries available in keras**

Dense and Flatten layers are used to design the output base on the required number of classes. Next VGG 16 and Pre_process input is imported. ImageData Generator is imported which is used in the data augmentation (Data Augmentation is a technique which significantly increases the amount of training data samples without actually collecting a new set of data. Data Augmentation techniques such as cropping, flipping, padding are commonly used to train large neural networks). Even sequential layers are imported.

3. **Providing Image size and path**

Initially, we should resize the images to (224,224) because the VGG 16 was designed to handle that size of images. Then we should provide training paths and test paths

4. **Importing the VGG 16 model**

We import the model and use the weights of the model itself. Later we

add the pre-processing layer to the front of VGG.

5. **Don’t train existing weights**

As we are using state of the art algorithm there is no need to train existing weights.

6. **We need to get the categories(output classes) as folders**

We use the glob function and find out the number of output classes.

7. **Flatten the layer before adding the last layer**

After flattening, append folders as a dense layer with the activation function

“softmax” while creating the model object.

8. Compile the model

To compile the model we use the following :

- Loss Function : ‘categorical_crossentropy’
- Optimizer : ‘Adam’
- Metrics : ‘accuracy’

9. After compiling the model, we should upload the dataset

Use ImageDataGenerator with features including rescaling, zoom_range, horizontal_flip etc. The settings in the data augmentation can be seen in the Table 4.2

Table 4.2 Data Augmentation

Method	Setting
Rescale	1/255
Rotation range	40
Width shift	0.2
Height shift	0.2
Shear range	0.2
Zoom range	0.2
Horizontal flip	True

10. Finally fit the model

Use fit.generator to fit the model and set the epochs, steps per epoch(training set) and validation steps(testing set)

11. Plot the loss function and check validation data

Finally, we plot the loss function and check the validation data.

4.6.1 The resulting neural network

The resulting neural network performs well as it is equipped with some good parameters. Figure 4.5 shows the outcome of the model. It describes the utilization of the entire architecture of the VGG 16 model with modified outputs which suits the predictions of our proposed system. The number of outputs depend upon the number of distinct classes to identify or classify using our model. The model mainly consists of feature extractors and classifiers. The final layer is a dense layer from which the output layer arises. The dense layer is fully connected. In our model, as we are utilizing the architecture of a pre-trained model, we need not change any weights but should modify the last layer as our model is a binary classifier. Figure 4.5 demonstrates the typical working of a neural network. In our model, we use the VGG 16 pre-trained network in the centre of Figure 4.5.

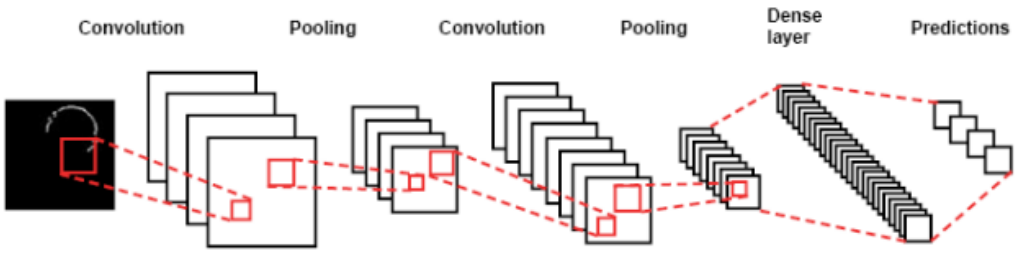


Figure 4.4 Working of Convolutional Neural Network

The input data sets are passes in a series of batches to the model. The model accepts the data only when it is reshaped into the tensor format. In this, we are considering images of size 224 x 224 with 3 colour channels namely RGB(Red, Blue, Green). Here we are considering 2D images with various network parameters. The format of our data which is in tensor format modifies and changes into different forms during the course of the process. We can consider different batches of images to study the model in more depth.

There is no particular way in which network parameters are chosen and the resulting model can have different parameters in different test cases. The only way in which one can find that the parameters are good is to try a different set of values and test cases. The resulting model has resulted by utilizing the parameters listed in the following tables. These layers and operations with network parameters found to be precisely good and accurate.

The parameters and the description of the model that were evaluate are given in the following tables :

- The first convolutional operation is performed two times with the following parameters i.e 2 convolutional layers which can be seen in Table 5.2. Here 2 convolution operations are performed.

Table 4.3 First Convolutional Layer

Input Data Shape	**[224,224,3]**
Number of filters	64
Kernel size	3x3
Padding	None
Strides	1
Activation	ReLu
Resulting Data shape	**[224,224,64]**

- The first max-pooling layer has the following parameters as seen in Table 4.4

Table 4.4 First Max Pooling Layer

Input Data Shape	**[224,224,64]**
Kernel size	2x2
Strides	2
Resulting Data Shape	**[112,112,64]**

- Next, another 2 convolution layers are added whose network parameters can be seen in Table 4.5. Here 2 convolutional operations are performed.

Table 4.5 Second Convolutional Layer

Input Data Shape	**[112,112,64]**
Number of filters	128
Kernel size	3x3
Padding	None
Strides	1
Activation	ReLu
Resulting Data shape	**[112,112,128]**

- The second max-pooling layer is added and has the following network parameters which can be seen in Table 4.6

Table 4.6 Second Max Pooling Layer

Input Data Shape	**[112,112,128]**
Kernel size	2x2
Strides	2
Resulting Data Shape	**[56,56,128]**

- Next, another 3 convolution layers are added whose network parameters can be seen in Table 4.7. Here 3 convolutional operations are performed.

Table 4.7 Third Convolutional Layer

Input Data Shape	**[56,56,128]**
Number of filters	256
Kernel size	3x3
Padding	None
Strides	1
Activation	ReLu
Resulting Data shape	**[56,56,256]**

- The third max-pooling layer is added and has the following network parameters which can be seen in Table 4.8

Table 4.8 Third Max Pooling Layer

Input Data Shape	**[56,56,256]**
Kernel size	2x2
Strides	2
Resulting Data Shape	**[28,28,256]**

- Next, another 3 convolution layers are added whose network parameters can be seen in Table 4.9. Here 3 convolution operations are performed.

Table 4.9 Fourth Convolutional Layer

Input Data Shape	**[28,28,256]**
Number of filters	512
Kernel size	3x3
Padding	None
Strides	1
Activation	ReLu
Resulting Data shape	**[28,28,512]**

- The fourth max-pooling layer is added and has the following network parameters which can be seen in Table 4.10

Table 4.10 Fourth Max Pooling Layer

Input Data Shape	**[28,28,512]**
Kernel size	2x2
Strides	2
Resulting Data Shape	**[14,14,512]**

- Next, another 3 convolution layers are added whose network parameters can be seen in Table 4.11. Here 3 convolution operations are performed.

Table 4.11 Fifth Convolutional Layer

Input Data Shape	**[14,14,512]**
Number of filters	512
Kernel size	3x3
Padding	None
Strides	1
Activation	ReLu
Resulting Data shape	**[14,14,512]**

- The fifth max-pooling layer is added and has the following network parameters which can be seen in Table 4.12

Table 4.12 Fifth Max Pooling Layer

Input Data Shape	**[14,14,512]**
Kernel size	2x2
Strides	2
Resulting Data Shape	**[7,7,512]**

- Finally, a fully connected layer is added along with dense layer at the last which classifies two categories of classes i.e Normal or Pneumonia classes represented by 0 and 1 for Normal and Pneumonia respectively The dense layer parameters can be seen in Table 4.13

Table 4.13 Dense Layer

Number of classes	**Binary**
Loss function	categorical_crossentropy
Activation	Softmax
Resulting Output	**1 or 0**

We use the softmax activation function to do classification in our model and output will be the number of categorical classes in our supplied data. In our data, we have two classes i.e normal and pneumonia. So, we should add the Dense layer such that it generates an output with a shape of (None,2). Here our model demonstrates a binary classifier. Finally, one could use cross_entropy function for the calculation of loss. In this model, the loss function uses is softmax cross entropy. This cross entropy is commonly used in various kinds of CNN architectures. Here we didn't use any modification parameters which can be drawn from TensorFlow. There could be some possible improvements related to loss function and we shall discuss this in discussion section of this report. The learning rate chosen was 0.001 and data augmentation was utilized.

This is all about the model involving the accurate pre-trained neural networks. We shall discuss the results and performance in the upcoming sections of this report.

4.7 Algorithm

Step 1: Importing the VGG 16 model and other necessary libraries from Keras.

Step 2: Resizing the images in the dataset to [224,224] with 3 colour channels.

Step 3: Providing the training path and validation path and storing them

Step 4: Adding the Pre-processing layer to the front of VGG after removing the top layer

```
Vgg = VGG16(input_shape=IMAGE_SIZE+[3],weights='imagenet',
        include_top=False)
```

Step 5: Don't train existing weights

```
for layer in vgg.layers:
  layer.trainable = False
```

Step 6: Flattening and adding the dense layer

```
x = Flatten()(vgg.output)
prediction = Dense(2, activation='softmax')(x)
```

Step 7 : Create a model Object

```
model = Model(inputs= vgg.input, outputs=prediction)
```

Step 8: Apply cost optimization and loss function to the model

```
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy'] )
```

Step 9: Create ImageDataGenerator

Step 10: Fit the model

```
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=5,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

Step 11: Plot the loss function and accuracy

Chapter 5
Results and Discussions

5.1 Network performance

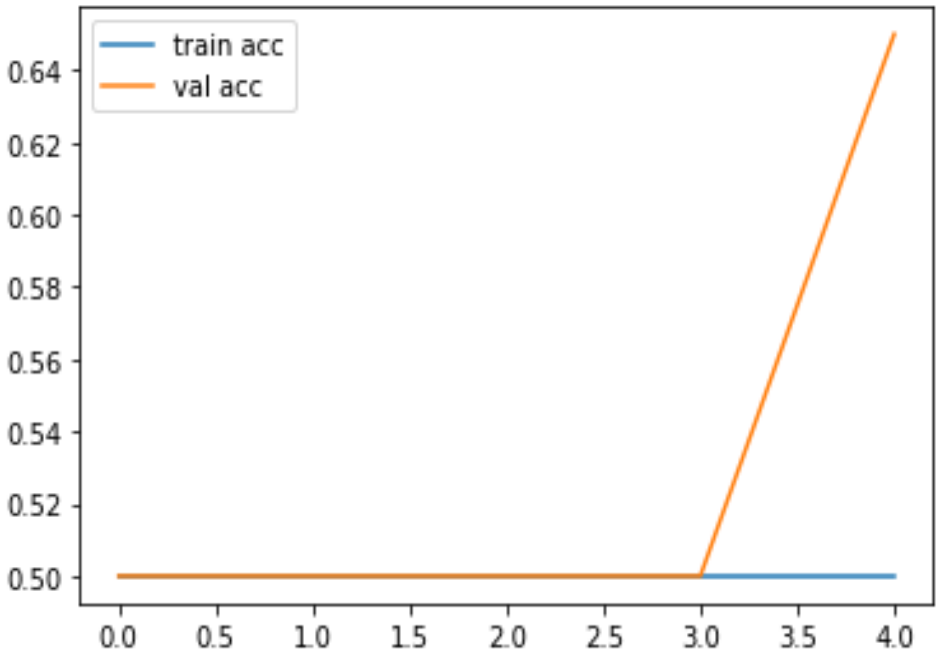
To evaluate and analyse the performance of the proposed model , we conducted the experiments several times with different sizes of datasets. However there were different parameters which were used to study the performance. In this report , we only present the most valid and better approach. Even large set of images require more training time as well as computational cost. Usually CNN models requires images of same sizes during training and that's the reason why we consider data set of similar sizes and resize them to suit the constraints.

In this way our model predicts the outcome based on the data. The input is set of images of fixed size (224,224) as the VGG 16 architecture is designed to accept the data in those parameters.

5.2 Accuracy results

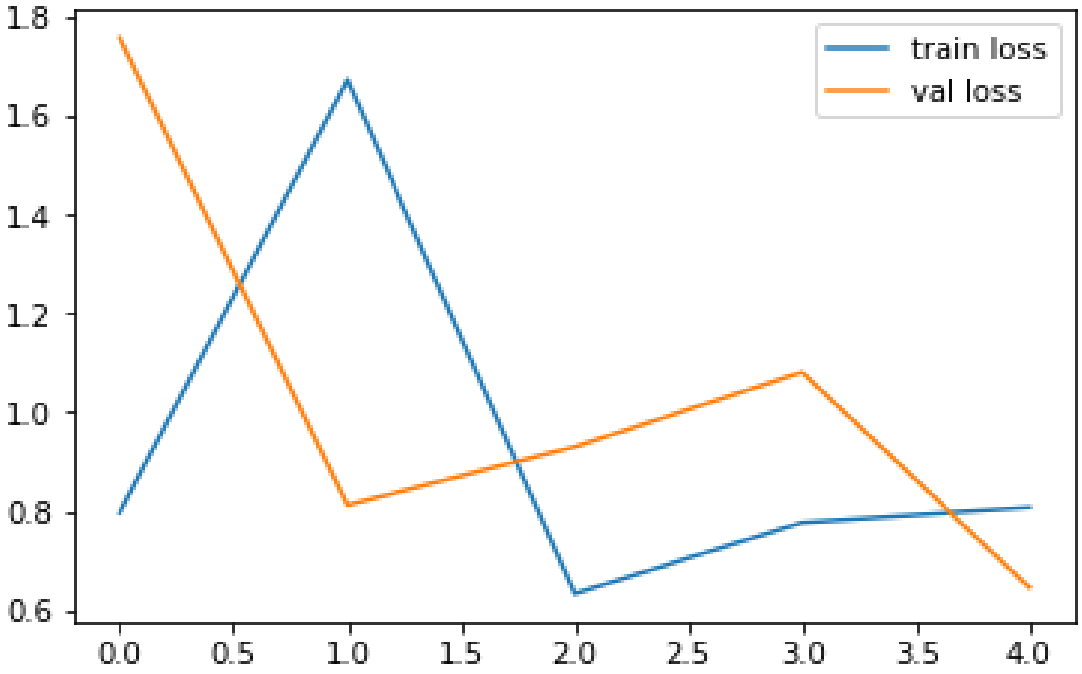
In the first graph in Figure 5.1, the accuracy of the proposed network is plotted.The accuracy is plotted against training period. The dataset used consists of set of images from both normal and pneumonia classes with fixed sizes of (224,224) with 3 channels (Red, Green, Blue) which are actually 2D images.In this graph , the accuracy for the network is shown when it is evaluating data from both test dataset and training dataset.

Figure 5.1 Accuracy



The graph in the Figure 5.2 projects the variation in the loss function over the training data . It almost got sync in with the Figure 5.1. Finally, the loss function levels out.

Figure 5.2 Plot for Loss Function



Various parameters and tool were utilized like data augmentation, different learning rate to fit the datasets perfectly into the deep neural network architectures. The final results obtained are loss=, training accuracy=, validation loss=, validation accuracy=.

Chapter 6
Conclusion and Future Scope of Study

Deep learning frameworks are playing a vital role in the field of medicine. The medical field is advancing at a faster pace and developments due to the utilization of technology particularly those based on the classification and data science. Our proposed system has shown better accuracy and validation than existing systems as we used the novel concept of transfer learning VGG 16 framework in modelling our data sets. The proposed model is highly effective in the diagnosis of pneumonia through chest x-rays and can be implemented easily with some good computing requirements showing a good amount of accuracy. In future, the model can be upgraded to even predict the type of virus and kind of various infections with the greater availability of open-source data sets of x-rays.

Chapter 7

References

[1] Vijendran, Sriram, and Rahul Dubey. "Deep Online Sequential Extreme Learning Machines and its Application in Pneumonia Detection." 2019 8th International Conference on Industrial Technology and Management (ICITM). IEEE, 2019.

Vijendran, Sriram, and Rahul Dubey. "Deep Online Sequential Extreme Learning Machines and its Application in Pneumonia Detection." 2019 8th International Conference on Industrial Technology and Management (ICITM). IEEE, 2019.