CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

Pneumonia is an infection of one or both of the lungs caused by bacteria, viruses, or fungi. There are more than 30 different causes of pneumonia, and they're grouped by the cause. The main types of pneumonia are bacterial, viral, and mycoplasma pneumonia. A cough that produces green, yellow, or bloody mucus is the most common symptom of pneumonia. Other symptoms include fever, shaking chills, shortness of breath, low energy, and extreme tiredness. Pneumonia can often be diagnosed with a thorough history and physical exam. Tests used to look at the lungs, blood tests, and tests done on the sputum you cough up may also be used. Treatment depends on the type of pneumonia you have. Antibiotics are used for bacterial pneumonia. It may also speed recovery from mycoplasma pneumonia and some special cases. Most viral pneumonias don't have a specific treatment and just get better on their own. Other treatment may include a healthy diet, more fluids, rest, oxygen therapy, and medicine for pain, cough, and fever control. Most people with pneumonia respond well to treatment, but pneumonia can cause serious lung and infection problems. It can even be deadly.

This repository contains a self-made Convolutional Neural Networks implemented using Keras, an open source neural network library in Python. A Convolutional Neural Network is a Deep Neural Network, that takes inputs asimages, assigns weights and biases to various aspects of the image and then differentiates among different class of images. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The model is computationally cheaper than other popular and successful implementations of CNNs such as the VGG16 and ResNet50, having only

291,809trainable parameters. Despite having 291,809 number of parameters, it manages a training set accuracy~ 94% and test set accuracy ~ 90%**.** It uses 4 Convolution andPooling layers followed by 2 Dense and Dropout layers to classify the images as having pneumonia or not having pneumonia. The classification occurs after the image is fed through a series of convolutional and max pooling layers that are activated by using the ReLU activation function that is subsequently fed into the neurons present in the dense layers and finally, the output neuron is activated by the sigmoidal function. Bacteria cause most cases of community-acquired pneumonia in adults. You can catch pneumonia when someone who is infected coughs or sneezes. Bacteria-filled droplets get into the air, where you can breathe them into your nose or mouth.

1.2 OVERVIEW

Now, we are passing a great time of science and technology. We can't skip our daily life from this technology era. Our main motive to find out whether the person is affected from pneumonia or not from given X-Ray (lung) images. There are some type of pneumonia where used in our research work. We want to identify those images by deep learning method Convolutional neural network. So here is our goal is given below down there. There are many more thing to discuss but we only focus on those importance thing which is very essential to our paper work.Our main goal is classification of normal or pneumonia in their patients record.To develop a multi-class classification model to perform predictions on the current diagnosis states of specified lungs. Convolutional Neural Networks (CNN) were usually built and trained for image classification tasks as they were commonly applied in the analysis of visual imagery and other backend systems in image classification. The affected states concerned in this project only ranges in between two distinct categories, "Bacterial" and "Viral". Training and evaluation of supervised learning models.

To utilize transfer learning approach on CNNs to achieve desired performance. Transfer learning technique can be utilized when there is limited data source for training or testing. Hence, by leveraging the knowledge or so called the pre-trained weights from the pre-trained models, the performance and accuracy achieved of networks could also be enhanced from any "keras API" networks. With pre-trained weights from existing networks, in spite of the limited training data on hand, the desired performance will still be reachable and the computing resources and time spent could be saved efficiently.

AI is a revolution in recent times in so many fields. With the help of AI we can solve numerous problems. Convolutional neural network (CNN) also known as image classifier can used in many types of fields to solve various problems like -viral agents or can detect disease in leaf. So, we find it useful to solve our problem also. We also find that recent state of art keras API library gives better performance interm of using CNN.In present time, AI is everywhere and we can also see the crisis with the lungs images classification. So we decide to do something new to detect covid pnemonia, viral pneumonia and bacterial pneumonia with the help of AI.

1.2.1 VIRAL PNEUMONIA

Viral pneumonia is defined as a disease entity wherein there is the viral causation of oxygen and carbon dioxide gas exchange abnormalities at the level of the alveoli, secondary to viral-mediated and/or immune response-mediated inflammation. The traditional role of viral pneumonia was as a disease found predominantly in the very young, the elderly, and those exposed to influenza. In the past, the diagnosis of viral pneumonia was predicated on it being somewhat a diagnosis of exclusion. History, physical exam, chest radiography, and available lab work (until recently) lacked sensitivity and specificity. Once bacterial pneumonia has been excluded, then viral pneumonia diagnosis was entertained.The increases in life span and early infant survivability have created

an additional population at greater risk of viral pneumonia. The increased number of those receiving immune-impairing therapy (radiation and /or chemotherapy) modifying hematological/immunological agents in chronic illness, resulting in secondary impaired immunity.The advent of HIV increase in the number of patients with inborn immune impairment serving bacterial infection secondary to antibiotic therapy. The increased incidence of organ transplantation and immuno suppressive therapy.

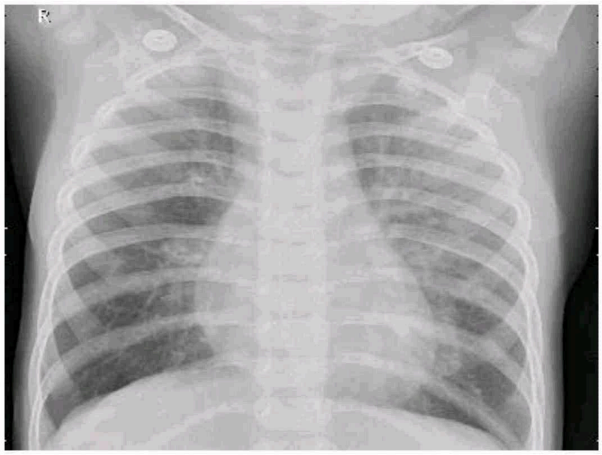


Figure 1.1 Viral Pneumonia

The increasing role of viral pathogens in pneumonia and the increased recognition of bacterial and viral coinfections in patients with pneumonia necessitate a higher clinical index of suspicion and early identification of viral pulmonary pathogens. This activity describes the evaluation and management of viral pneumonia and highlights the role of the interprofessional team in improving care for affected patients. Your doctor’s diagnosis will depend on how severe your infection is. If you have mild symptoms, your doctor may suggest blood tests or a chest X-ray. If your symptoms are serious, and you are 65 or older (or an infant or young child), your doctor might want to test some of your fluids. They may also put a camera down your throat to check your airways.

1.2.2 BACTERIAL PNEUMONIA

Identify the etiology of bacterial pneumonia. Recall the X-ray findings in a patient with bacterial pneumonia. Outline the treatment and management options available for bacterial pneumonia. Employ interprofessional team strategies for improving care coordination and communication to advance the management of patients affected by bacterial pneumonia and improve outcomes. Access free multiple choice questions on this topic. Introduction The word "pneumonia" takes its origin from the ancient Greek word "pneumon," which means "lung," so the word "pneumonia" becomes "lung disease." Medically it is an inflammation of one or both lungs' parenchyma that is more often, but not always, caused by infections. The many causes of pneumonia include bacteria, viruses, fungi, and parasites.

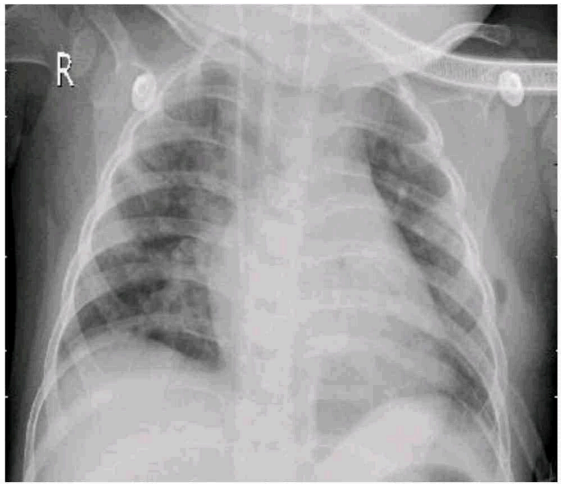


Figure 1.2 Bacterial pneumonia

Community-acquired pneumonia can be caused by an extensive list of agents that include bacteria, viruses, fungi, and parasites, but this article will focus on bacterial pneumonia and its causes. Bacteria have classically been categorized into two divisions based on etiology, "typical" and "atypical" organisms.

1.2.3 COVID PNEUMONIA

Pneumonia occurs when a bacterial or viral infection causes significant damage and inflammation in the lungs. The resulting fluid and debris build-up makes it hard for a person to breathe — sometimes to such an extent that oxygen therapy or ventilator support is required. Regardless of the bacteria or virus causing it, pneumonia can become very serious, life-threatening.In the case of COVID pneumonia, the damage to the lungs is caused by the coronavirus that causes COVID-19. When COVID pneumonia develops, it causes additional symptoms, such as Shortness of breath, Increased heart rate Low blood pressure.

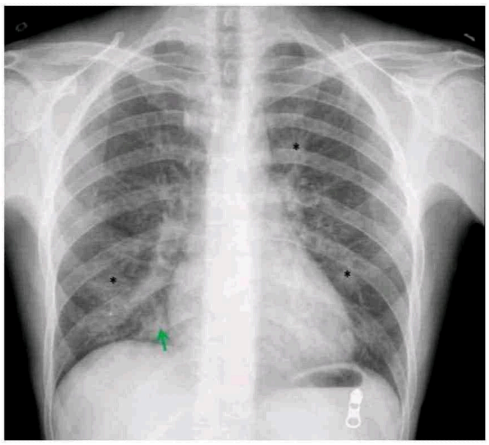


Figure 1.3 Covid Pneumonia

What's more is that COVID pneumonia often occurs in both lungs, rather than just one lung or the other. Additionally, the widespread inflammation that occurs in some people with COVID-19 can lead to acute respiratory distress syndrome (ARDS) — a severe type of lung failure. Like other respiratory infections that cause pneumonia, COVID-19 can cause short-term lung damage. In more severe cases, the damage can last a long time. In fact, early data is showing that up to a third of COVID pneumonia patients have evidence of scarring on X-rays or lung testing a year after the infection.

1.3 DEEP LEARNING

Deep learning(DL) is a type of machine_learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier. Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers. In a word, accuracy. Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images. While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful to Deep learning requires large amounts of labeled data. For example, driverless car

development requires millions of images and thousands of hours of video. Deep learning requires substantial computing power. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.If you are just starting out in the field of deep learning or you had some experience with neural networks some time ago, you may be confused. I know I was confused initially and so were many of my colleagues and friends who learned and used neural networks in the 1990s and early 2000s. The leaders and experts in the field have ideas of what deep learning is and these specific and nuanced perspectives shed a lot of light on what deep learning is all about.

1.3.1Examples of Deep Learning at Work

Deep learning applications are used in industries from automated driving to medical devices.

Automated Driving: Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.

Aerospace and Defense: Deep learning is used to identify objects from satellites that locate areas of interest, and identify safe or unsafe zones for troops.

Medical Research: Cancer researchers are using deep learning to automatically detect cancer cells. Teams at UCLA built an advanced microscope that yields a high-dimensional data set used to train a deep learning application to accurately identify cancer cells.

Industrial Automation: Deep learning is helping to improve worker safety around heavy machinery by automatically detecting when people or objects are within an unsafe distance of machines.

Electronics: Deep learning is being used in automated hearing and speech translation. For example, home assistance devices that respond to your voice and know your preferences are powered by deep learning applications.

1.3.2 Rise of Deep Learning

Machine learning is said to have occurred in the 1950s when Alan Turing, a British mathematician, proposed his artificially intelligent "learning machine." Arthur Samuel wrote the first computer learning program. His program made an IBM computer improve at the game of checkers the longer it played. In the decades that followed, various machine learning techniques came in and out of fashion.

Neural networks were mostly ignored by machine learning researchers, as they were plagued by the 'local minima' problem in which weightings incorrectly appeared to give the fewest errors. However, some machine learning techniques like computer vision and facial recognition moved forward. In 2001, a machine learning algorithm called Adaboost was developed to detect faces within an image in real-time.

1.4 DEEP LEARNING VS MACHINE LEARNING

If deep learning is a subset of machine learning, how do they differ? Deep learning distinguishes itself from classical machine learning by the type of data that it works with and the methods in which it learns. Machine learning(ML) algorithms leverage structured, labeled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. This doesn't necessarily mean that it doesn't use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format. Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data,

like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let's say that we had a set of photos of different pets, and we wanted to categorize by "cat", "dog", "hamster", etc. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert. Then, through the processes of gradient descent and back propagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions about a new photo of an animal with increased precision. Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn't require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

1.5 HOW DEEP LEARNING WORKS

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data. Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is

where the final prediction or classification is made. Another process called back propagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backward through the layers in an effort to train the model. Together, forward propagation and back propagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate. The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. Neural networks are layers of nodes, much like the human brain is made up of neurons. Nodes within individual layers are connected to adjacent layers. The network is said to be deeper based on the number of layers it has. A single neuron in the human brain receives thousands of signals from other neurons. In an artificial neural network, signals travel between nodes and assign corresponding weights. A heavier weighted node will exert more effect on the next layer of nodes. The final layer compiles the weighted inputs to produce an output.

For example,

- Convolutional neural networks (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
- Recurrent neural network (RNNs) are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

CHAPTER 2

LITERATURE SURVEY

This section provides a literature review of previous researches that used Deep Learning(DL) techniques for pneumonia diagnosis.

Julianna Antonchuk ,Benjamin Prescott [1] implemented COVID-19 Pneumonia and Influenza Detection Using Convolutional Neural Networks in 2021 to analyse visual imagery of CXR images using Convolutional Neural Networks.To analyze visual imagery of CXR images with COVID-19 pneumonia, influenza virus pneumonia, and normal biomarkers, we have considered a convolutional neural network topology of deep and shallow architectures. In the initial experimentation phase, we identified a convolutional neural network with 512 units, followed by a pooling operation for two-dimensional spatial data with a size of 2x2, and a flattened layer of 64 nodes resulting in 95% accuracy. We use the model as an indirect baseline while constructing architectures for multi-class image classification.

Luka racic, Tomo popovic [2] has implemented Pneumonia Detection Using Convolutional Neural Networks in2021. To analyse Images of Chest X-Ray, dataset, Preprocessing of Images Using convolutional Neural Networks. Artificial intelligence has found its use in various fields during the course of its development, especially in recent years with the enormous increase in available data. Its main task is to assist making better, faster and more reliable decisions. Artificial intelligence and machine learning are increasingly finding their application in medicine. This is especially true for medical fields that utilize various types of biomedical images and where diagnostic procedures rely on collecting and processing a large number of digital images. The application of machine learning in processing of medical images helps with consistency and boosts accuracy in reporting.

Alexandr A.Kalinin, [3] has implemented the Deep Learning For Automatic Pneumonia Detection in 2020. To analyse Retina Net[33] with implementation on Pytorch[24].Often, the solutions in machine learning competitions are based on large and diverse ensembles, test-time augmentation, and pseudo labelling, which is not always possible and feasible in real-life applications. At test-time, we often want to minimize a memory footprint and inference time. Here, we propose a solution based on a single model, ensembled over several checkpoints and 4 folds. The model utilises an SSD Retina Net with SE-Res Net encoder pre-trained on Image Net. The extra output for global image classification with one of the classes ('No Lung Opacity / Not Normal', 'Normal', 'Lung Opacity') was added to the model. The total loss was combined of this global classification output with regression loss and individual boxes classification loss. For an ablation study, we trained the Retina Net model with the SE-ResNext-101 encoder and fixed augmentations with and without the global classification output

A.Tilve, et al [4] has implemented Pneumonia detection Using deep learning Approaches in 2020 to X-Ray Images into standard Format CNN, RESNET,DENSENET, ANN and KNN. Pneumonia is among the most prevalent diseases, and due to lack of experts it is difficult to detect. This paper focuses on surveying and comparing the detection of lung disease using different computer-aided techniques and suggests a model for detecting pneumonia, which will then be implemented as part of our future research. In this survey, we also tried to familiarize ourselves with the different image pre-processing techniques used to convert raw X-ray images into standard formats for analysis and detection, machine learning techniques which is an important phase in accurate pneumonia detection.

Shah,et,al. [5] has implemented Pneumonia detection using Convolutional neural Networks in 2020 To predict Pneumonia is a fatal disease that majorly affects the elderly and may sometimes prove to be life threatening. Early diagnosis of Pneumonia gains a paramount importance for saving many human lives. This paper aims at the detection and classification of patients affected by Pneumonia based on their chest X-rays. A convolutional neural network is employed from scratch to make the above diagnosis and yield highly accurate results. Deep Learning models automate the process and ensure speedy, adroit, and adept result when provided with X-rays of patients. The classification occurs after the image is fed through a series of convolutional and max pooling layers that are activated by using the ReLU activation function that is subsequently fed into the neurons present in the dense layers and finally, the output neuron is activated by the sigmoidal function. The accuracy increases as the model trains and decreases the loss simultaneously. Over fitting is prevented by implementing data augmentation before fitting the model. Thus, efficient, and cogent results are obtained by the proposed deep learning models to classify the chest X-rays for the detection of pneumonia.

E. ayan et al, [6] has implemented the Diagnosis of Pneumonia from Chest X-Ray Images Using deep learning in that year 2019. Diagnosis from X-Ray Images computer aided diagnosis system are needed Xeption and Vgg16.Pneumonia is a disease which occurs in the lungs caused by a bacterial infection. Early diagnosis is an important factor in terms of the successful treatment process. Generally, the disease can be diagnosed from chest X-ray images by an expert radiologist. The diagnoses can be subjective for some reasons such as the appearance of disease which can be unclear in chest X-ray images or can be confused with other diseases. Therefore, computer-aided diagnosis systems are needed to guide the clinicians. In this study, we used two well-known convolutional neural network models Xception and Vgg16 for

diagnosing of pneumonia. We used transfer learning and fine-tuning in our training stage. The test results showed that Vgg16 network exceed Xception network at the accuracy with 0.87%, 0.82% respectively.

Y Karthick Tharkal, varshini, [7] has implemented Pneumonia Detection using CNN based on Feature Extraction, in that year of 2019. To implementate. Densely Connected Convolutional Neural Networks (DenseNet-169) Pre-processing stages, feature Extraction. This section deals with the detailed description of the applied methodology. The proposed pneumonia detection system using the 'Densely Connected Convolutional Neural Network' (DenseNet-169) is described in Figure . The architecture of the proposed model has been divided into three different stages - the preprocessing stage, the feature extraction stage and the classification stage.

D. Varshni, et al, [8] has implemented that Pneumonia detection Using CNN based Feature Extraction in that year of 2019 to analyse the X-Rays which are used to diagnose to Convolutional Neural networks have gained much attention utilized as feature extractors.Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans commonly caused by bacteria called Streptococcus pneumonia. One in three deaths in India is caused due to pneumonia as reported by World Health Organization (WHO). Chest X-Rays which are used to diagnose pneumonia need expert radiotherapists for evaluation. Thus, developing an automatic system for detecting pneumonia would be beneficial for treating the disease without any delay particularly in remote areas. Due to the success of deep learning algorithms in analyzing medical images, Convolutional Neural Networks (CNNs) have gained much attention for disease classification.

Mohhammed Aledhari, [9] has implemented the Optimized CNN based Diagnosis System to Detect the Pneumonia from Chest Radiographs in that year of 2016. Pre-trained model fed into the fine turning portion of the CNN using Receiver Operating Characteristic (ROC).Pneumonia is a high mortality disease That fighting pneumonia-related infections in that. Early detection and intervention are crucial in treating pneumonia related infections. Since chest x-ray is one of the simplest and cheapest methods to diagnose pneumonia, we propose a deep learning algorithm based on convolutional neural networks to identify and classify pneumonia cases from these images. For all three models implemented, we obtained varying classification results and accuracy. Based on the results, we obtained better prediction with average accuracy of (68%) and average specificity of (69%) in contrast to the current state-of-the-art accuracy that is (51%) using the Visual Geometry Group (VGG16 also called Oxford(Net), which is a convolutional neural network architecture.

Abdullah, etal, [10] has implemented the Preliminary study of Pneumonia symptoms detection method using cellular Neural Networks in that year 2011 to predict that Medical diagnosis is one of the most important procedure in which image processing are usefully applied. In this paper, a pneumonia symptoms detection method based on cellular neural networks (CNNs) is proposed. The CNN design is characterized by a virtual template expansion obtained through a multi step operation. It is based on linear space invariant 3 × 3 templates. The proposed design is capable of performing pneumonia symptoms detection within a short time. The main idea in Cellular Neural Network is that connection is allowed between adjacent units only. There are few rules in Cellular Neural Network that has to be implemented when designing the templates, such as state equation, output equation, boundary equation, and also the initial value.

CHAPTER 3

EXISTING SYSTEM

3.1 OVERVIEW

In the existing system,several algorithm such as Random Forest &KNN helps to detect and predict average accuracy.

- Random Forest Classifier : 67.80%
- KNN Classifier : 43.25%
- Logistic Regression : 25.81%
- No support colour images
- This reach to increase in time to reach the optimal solution

In recent times after undergoing treatment for the new coronavirus (COVID-19),certain elderly patients get affected to post COVID Pneumonia.Now-a days Mostly Pneumonia would be diagnosed by a physician using radiological imaging and determining the infectious agent that causes the disease.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

Such method might require a lot of resources so this method might be time consuming as the presence of pneumonia can only be diagonised once all these resources are ready.

- No flexibility
- Low Accuracy
- Low efficiency
- High complexity

CHAPTER 4

PROPOSED SYSTEM

4.1 OVERVIEW

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image datasets, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.As a solution to this problem statement, we propose a model, which detects the presence of pneumonia using the given X-Ray chest dataset as input.To achieve this, we use a deep convolutional neural network, to detect patterns and features in the images.The human brain is modeled by using design and implementation of neural network. The neural network is mainly used for vector quantization, approximation, data clustering, pattern matching, optimization functions and classification techniques. The neural network is divided into three types based on their inter connections.Three type neural networks are feedback, feed forward and recurrent network. The Feed Forward Neural network is further divided into single layer network and multilayer network.

4.1.1 ADVANTAGE OF PROPOSED SYSTEM

- Augmentation of images is done to increase accuracy
- Datasets will be increased
- Architecture of CNN is used
- Prediction will be done perfectly
- More number of images can be processed
- Efficient at delivering high quality results.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE AND SOFTWARE SPECIFICATION

5.1.1 HARDWARE REQUIREMENTS

- Processor : Intel(R) Celeron(R) CPU 877 @ 1.40GHz
- RAM : 4GB (minimum)
- ROM : 4GB and More
- Keyboard : Standard keyboard
- Monitor : 14.0 inch color monitor

5.1.2 SOFTWARE SPECIFICATION

- Operating System : Windows10 pro
- Software : Google colaboratory
- Language : PYTHON
- Framework : Tensor Flow
- Library : Keras API

5.2 TECHNOLOGIES USED

5.2.1 SOFTWARE: GOOGLE COLABORATORY

Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

5.2.2 LANGUAGE: PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant white space. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

PYTHON WORK

- Python can be used on a server to create web applications.
- Python can be used along side software to create workflows.
- Python can connect to database systems.It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.

5.4.3 FRAMEWORK: TENSORFLOW

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions. The official website of TensorFlow is mentioned below link follow as there is a https://www.tensorflow.org/ Let us now consider the following important as features of TensorFlow: It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multidimension arrays called tensors.It includes a programming support of deep neural networks and machine learning techniques.Pip install tensorflow .

TensorFlow - Convolutional Neural Networks

After understanding machine-learning concepts, we can now shift our focus to deep learning concepts. Deep learning is a division of machine learning and is considered as a crucial step taken by researchers in recent decades. The examples of deep learning implementation include applications like image recognition and speech recognition. Following are the two important types of deep neural networks

- Convolutional Neural Networks
- Recurrent Neural Networks

In this chapter, we will focus on the CNN, Convolutional Neural Networks

5.2.4 LIBRARY: KERAS API

An open source neural network library in Python. It is a high-level, deep learning API developed by Google for implementing neural networks. It also support multiple back-end neural network computation. This makes deep learning models fast and easy. Deep learning is one of the major subfield of machine learning framework.Machine learning is the study of design of algorithms, inspired from the model of human brain. Deep learning is becoming more popular in data science fields like robotics, artificial intelligence(AI), audio & video recognition and image recognition. Artificial neural network is the core of deep learning methodologies. Deep learning is supported by various libraries such as Theano, TensorFlow, Caffe, 32Mxnet etc.,

FEATURES

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features:

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and backends.
- It is user friendly framework which runs on both CPU and GPU.

CHAPTER 6

SYSTEM DESIGN AND METHOD

6.1 DATASET COLLECTION

We used a dataset from kaggle.com for this study. At first, the dataset is (https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia), data set is divided into 4 categories, as follows:

Total Images-10014	Train set	Test Set
Normal	3270	270
Bacterial Pneumonia	3001	200
Viral Pneumonia	1656	256
Covid Pneumonia	1291	100

Table 6.1 Train and Test Datasets

The dataset contains images that were used for training, testing .

- The chest X-ray image dataset is organized into 2 folders (Train, Test).
- There is a total of 10044 images under four categories(normal, Bacterial Pneumonia, Viral Pneumonia, and Covid Pneumonia)
- Training set accuracy ≈ 96% and Test set accuracy ≈ 93%.

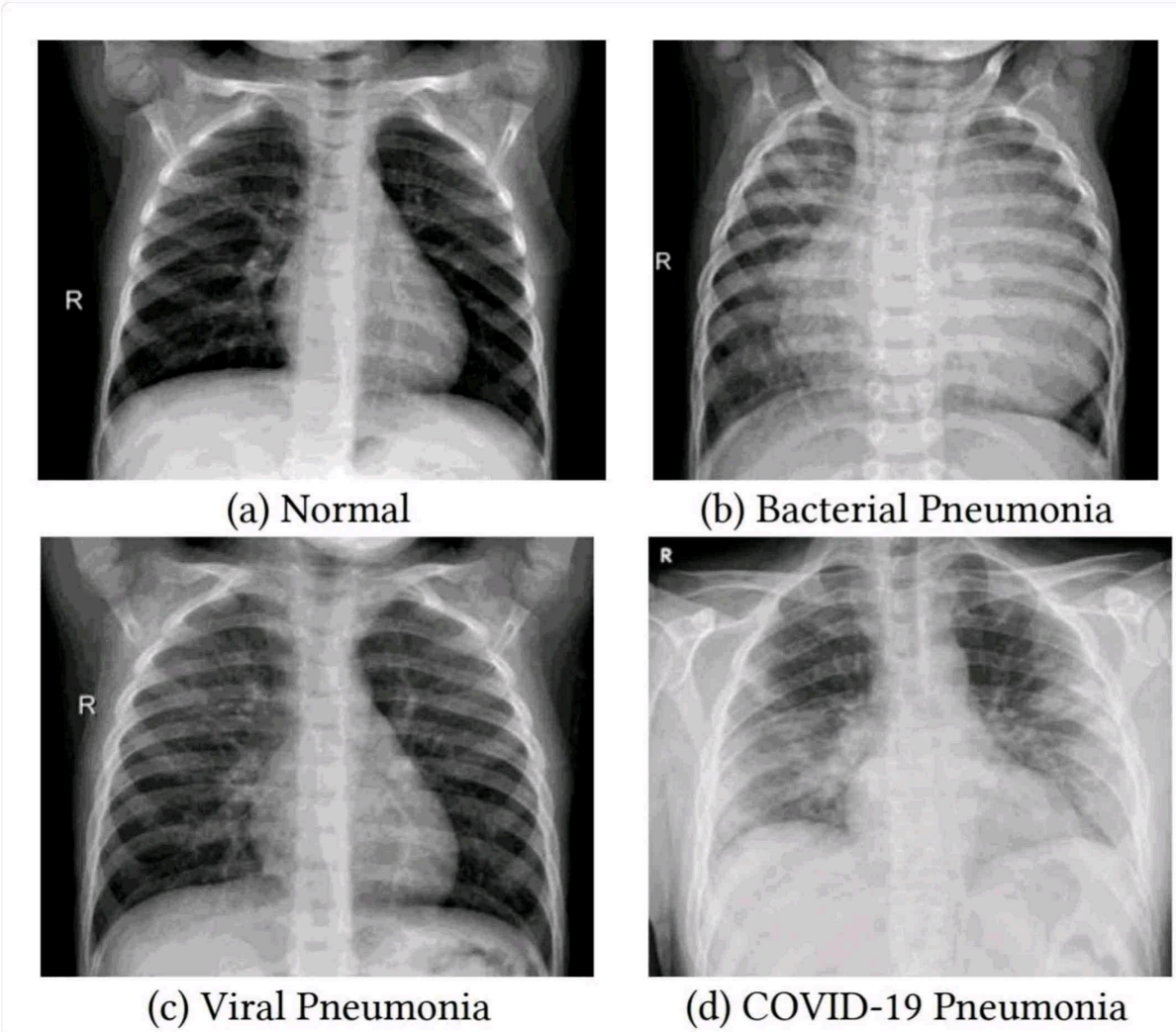


Figure 6.1 X-ray Datasets Images

6.2 DATA PREPROCESSING

The data set was taken from an online platform named Kaggle.The size of the data set was trimmed to 400 images feeding the input directly into the model, the data which is the set of fund us images must undergo some preprocessing steps resizing of images size from 512*512.The input data set is classified into three different categories. They are a. Training dataset, the dataset that is used to train or exercise the model. This data is labeled data set. Testing data set, which is used to test the model. Validation data set, the dataset that is used to validate the model. Validation data set is used to ensure that the model is not

over-fitting whereas training data helps to minimize the loss function. Updating of weights happens accordingly when the training data set is exercised in the model but validation data set does not involve any updating process. Training dataset and validation dataset are labeled but not the testing dataset. Also, one hot encoding is performed on the training labels.

6.3 DATA AUGMENTATION

The images sets are augmented using ImageDataGenerator() in keras to improve efficiency.The images in the dataset are not all the same size, so preprocessing was us needed for this study. Deep learning models require a significant amount of data for training rather than machine learning. We used Keras' ImageDataGenerator tool to resize all of the images to 256 x 256 pixels. We normalized both images after transforming them to 256 X 256. For faster calculation, images are converted to NumPy arrays. The volume of data may be increased by rotating, zooming, shearing, and flipping horizontally. Photos are obtained as well. The photos are then reshaped into 200 x 200 pixels for passing into the second convolution layer, and then down to 64 x 64 pixels for passing into the third convolution layer.

6.4 MODEL TRAINING

Transfer learning is a concept of reusing and transferring the knowledge and dataset from a previously trained model. Starting to train the model with the training dataset. Once the data set is well prepared, the data set is then fed into the model to start training

6.5 ARCHITECTURE OF CNN

The dataset is downloaded from Kaggle, and it is divided into training and testing datasets and stored in the colaboratory's local storage. Initially the model is allowed to train, the training dataset is passed through a series of convolution and maxpooling layers and to detect patterns, texture and feature maps in the images and then the output from the last pooling layer is passed through a flatten layer to convert the 2 Dimensional images to 1 dimensional images and set ready to pass it through the dense layer to classify the images as pneumonia and normal. After training is successful the ,the model is evaluated using the test dataset ,this Evaluation stage just gives the percentage of answers that the trained model got right in the form of accuracy. Once the successful model is trained and evaluated it is set ready for prediction, The images of X-rays is then sent into the final model and then the model predicts whether the given image is pneumonia or normal.

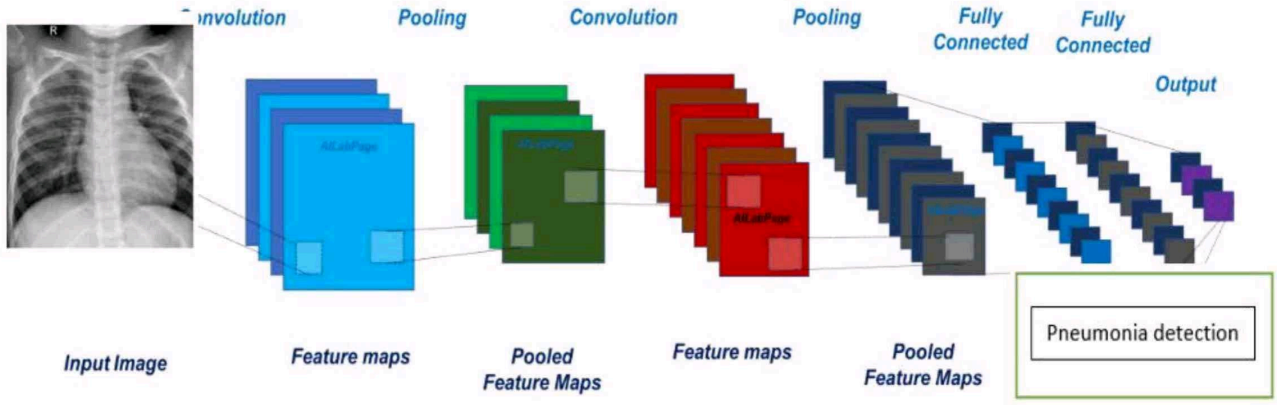


Figure 6.2 Architecture of CNN

6.6 MODEL ARCHITECTURE- CNN LAYERS

Input Layer This layer consists of 786 x 786 neurons which is equal to the count of pixels of each individual image being passed. Here, the pixels values of the training images are sent to the input layer. Convolutional Layer 1 This layer consists of 32 neurons. There is a connection between each of the neurons present

in this layer to all of the neurons in the previous layer. Convolution is performed on the input pixels, which is a process of performing dot product on the pixel values with arbitrary numbers called as filters. So, the layer's output is further passed to the max pooling layer. Max Pooling Layer 1 With the filters provided max pooling operation is performed on the received input which is identification of highest value in each patch of feature map. Convolution Layer 2 The max pooling layer's output is concatenated to a convolution layer (convolution layer 2) with 16 filters, kernel size as 4*4 and activation function 19 as ReLu. This layer is further passed to a max pooling layer.

TABLE 1 DESCRIPTION OF LAYERS CURRENTLY IN USE			
Layers	**Size of the output**	**Size of the Kernel**	**Stride**
Input Image	224 x 224	-	-
Convolution 1	220×220×8	5	1
Max Pooling 1	110×110×8	2	2
Convolution 2	106×106×16	5	1
Max Pooling 2	53×53×16	2	2
Convolution 3	47×47×32	7	1
Max Pooling 3	23×23×32	2	2
Convolution 4	21×21×64	3	1
Max Pooling 4	10×10×64	2	2
Convolution 5	8×8×128	3	1
Max Pooling 5	4×4×128	2	2
Convolution 6	2×2×256	3	1
Flatten	1024		
Dense	512		
Softmax	6		

Table 6.2 Description of Layers

Max Pooling Layer 2 The max pooling layer performs the max pooling operation on the received input. Then the output of the max pooling layer is flattened. Flattening is a process of converting any matrix into one dimensional array. Flatten function is applied on the convolution layer to create a single long feature vector. Output Layer The total amount of neurons existing in this layer is equal to the number of levels the disease is classified into.

The neuron consisting of the maximum value ranging between 0-1 will be the output i.e., the level in which the disease is,This output will be compared with the actual values and the error is determined. Based on the error the model tunes its underline parameters such that the error is as minimum as possible. This operation is performed on each and every training image.

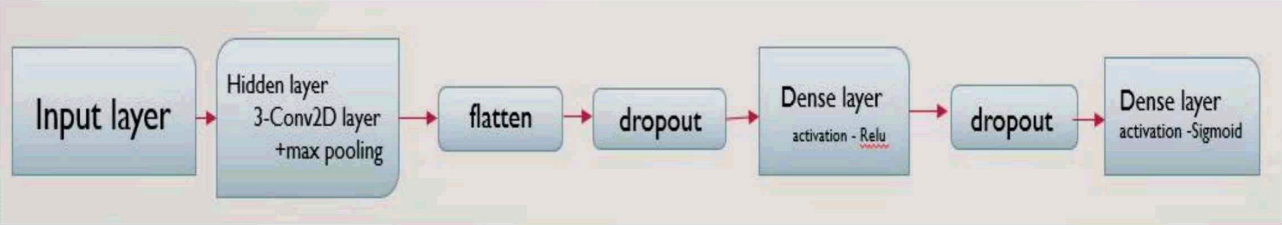


Figure 6.3 CNN layers

The rectified linear activation function or ReLU for short is piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.

6.7 ARCHITECTURE DIAGRAM

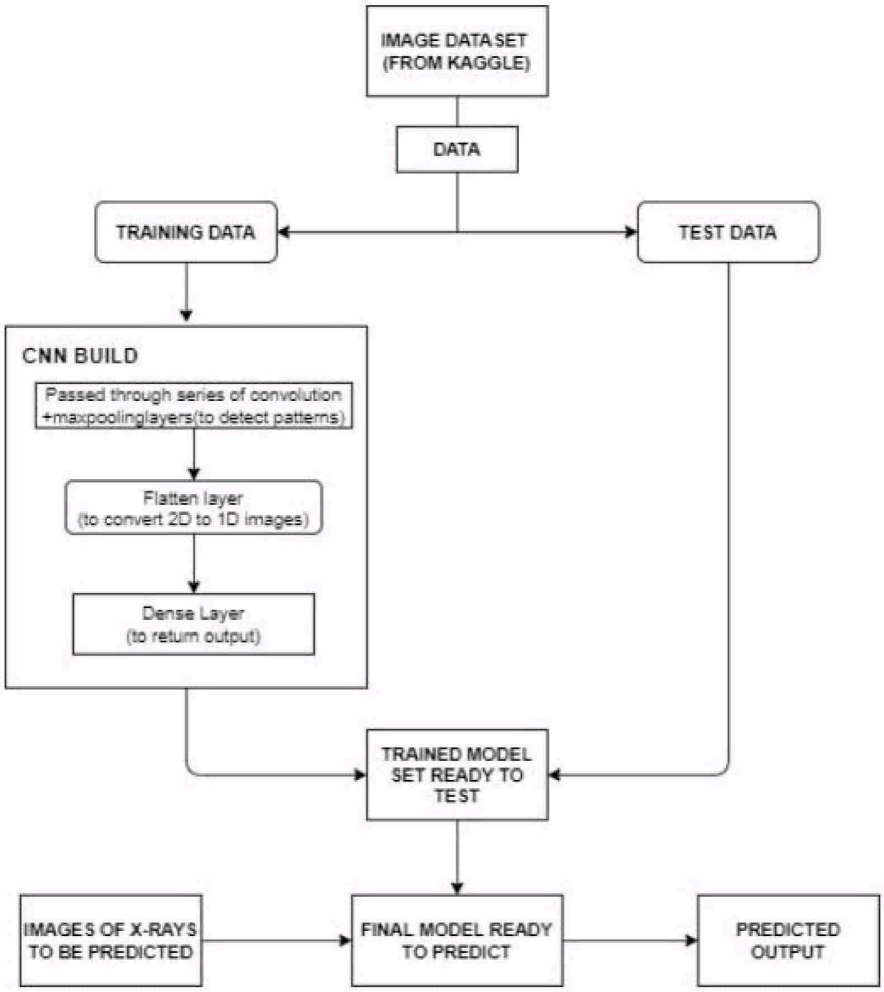


Figure 6.4 Architecture Diagram

6.8 IMAGE CLASSIFICATION

Image classification is a complex process that may be affected by many factors. Because classification results are the basis for many environmental and socioeconomic applications, scientists and practitioners have made great efforts in developing advanced classification approaches and techniques for improving classification accuracy. Image classification is used in a lot in basic fields like medicine, education and security. Correct classification has vital importance, especially in medicine. Therefore, improved methods are needed in this field.

The proposed deep CNNs are an often-used architecture for deep learning and have been widely used in computer vision and audio recognition.

Image classification pipeline:

• **Input:** Our input consists of a set of *N* images, each labeled with one of *K* different classes. We refer to this data as the training *set*.

• **Learning:** Our task is to use the training set to learn what every one of the classes looks like. We refer to this step as training a classifie*r*, or learning a model.

• **Evaluation:** In the end, we evaluate the quality of the classifier by asking it to predict labels for a new set of images that it has never seen before. We will then compare the true labels of these images to the ones predicted by the classifier. Intuitively, we're hoping that a lot of the predictions match up with the true answers.

6.9 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks are very similar to ordinary Neural Networks. They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product, and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the others. CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is a mathematical construct that is typically composed of three types of layers : convolution, pooling, and fully connected layers. The first two, convolution and pooling layers, perform feature 8 extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification.

A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. In digital images, pixel values are stored in a two-dimensional (2D) grid, i.e., an array of numbers and a small grid of parameters called kernel, an optimizable feature extractor, is applied at each image position, which makes CNNs highly efficient for image processing, since a feature may occur anywhere in the image. As one layer feeds its output into the next layer, extracted features can hierarchically and progressively become more complex. The process of optimizing parameters such as kernels is called training, which is performed so as to minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

6.10 CONVOLUTIONAL LAYERS

The neural network was built using a combination of the following layers of the KerasAPI :

- Input layer
- Convo-layer (Convo+ReLU)
- Pooling layer
- Fully connected (FC)layer
- Soft Max / logistic layer
- Output layer

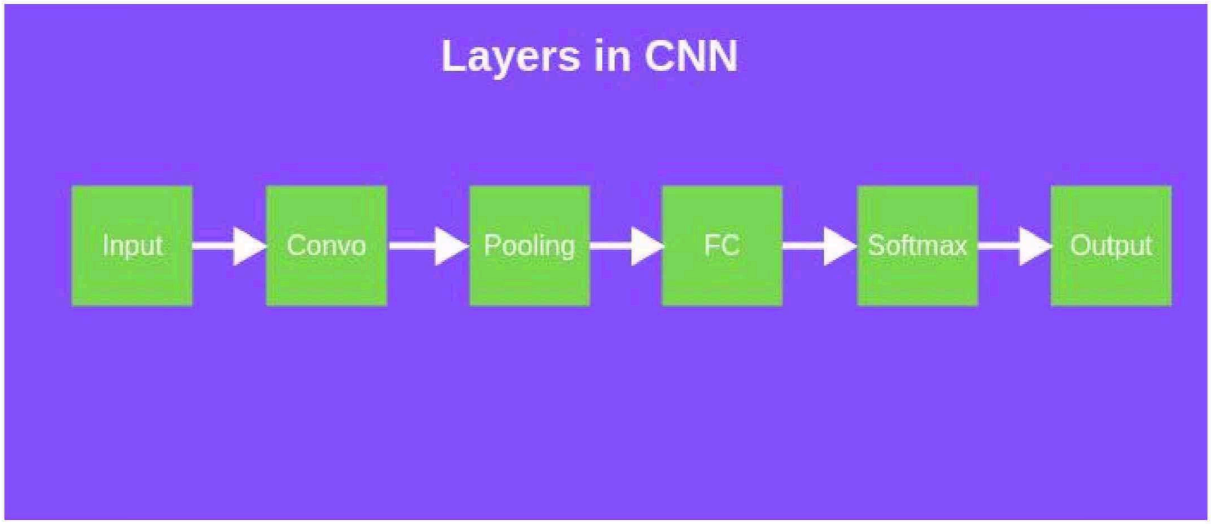


Figure 6.5 Layers In CNN

6.10 DIFFERENT LAYERS OF CNN

6.10.1 INPUT LAYER

Input layer in CNN should contain image data. Image data is represented by three-dimensional matrix as we saw earlier. You need to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, you need to convert it into784 x 1 before feeding into input. If you have “m” training examples then dimension of input will be (784, m).

6.10.2 CONVO LAYER

Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, apart of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume.Then we slide the filter over then extreceptive field of the same input image by a Stride and do the same operation again. We will repeat the same process again and again until we go through the whole image The output will be the input for the next layer. Convo layer also contains ReLU activation to make all negative value to zero.

6.10.3 POOLING LAYERS

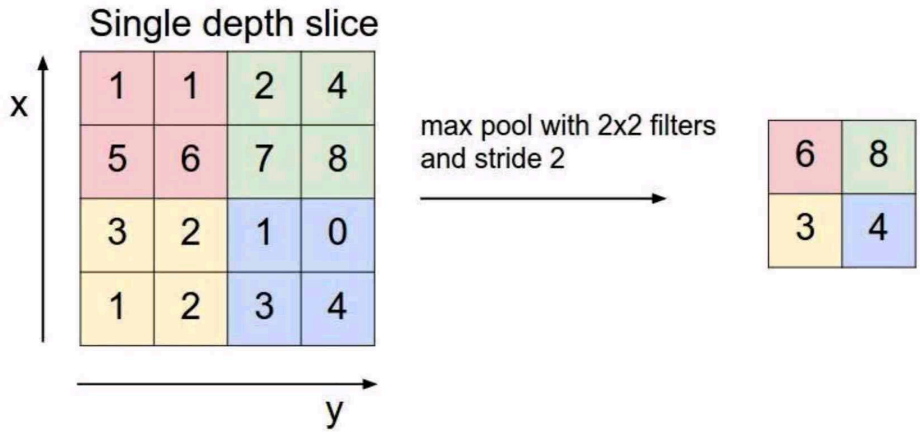


Figure 6.6 Pooling Layers

Pooling layer is used to reduce the spatial volume of input image after with convolution. It is used between two convolution layers. If we apply FC after Convo layer without applying pooling or max pooling, then it will be there acomputationally expensive and we don't want it. So, the max pooling is only way to reduce the spatial volume of input image. In the above example, we have applied max pooling in single depth slice with Stride of 2. You can observe the 4 x4-dimension input is reduce to2x 2dimension. There is no parameter in pooling layer but it has two hyper parameters

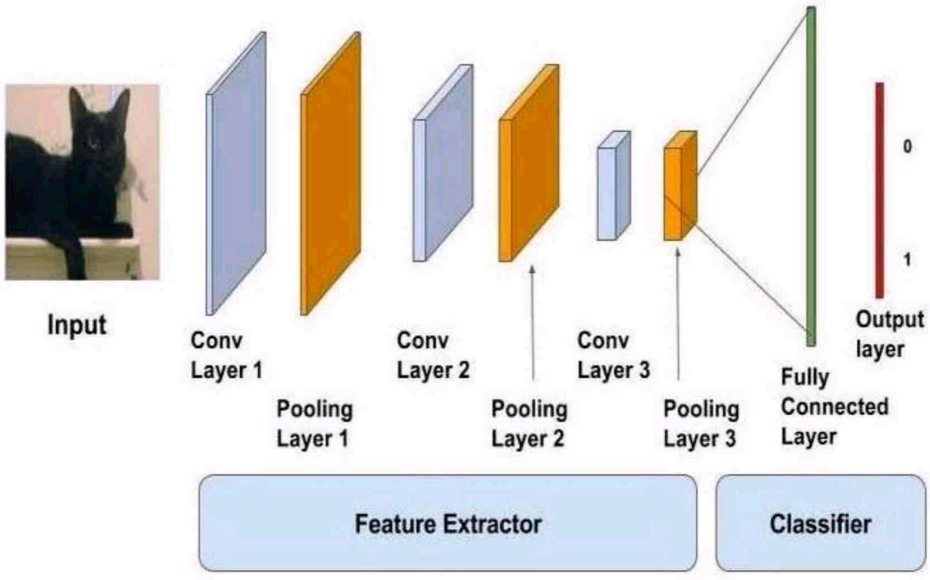


Figure 6.7 CNN Layers

6.10.4 FULLY CONNECTED LAYER

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training.

6.10.5 SOFTMAX /LOGISTIC LAYER:

Soft Max or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and Soft Max axis for multi classification.

6.10.6 OUTPUT LAYER:

Output layer contains the label which is in the form of one-hot encoded.

6.10.7 PARAMETER

Model Parameters are properties of training data that will learn during the learning process .they are often used of how well a model is performing.Epochs refer to how many times the input will used to update the weights while training the model. we are using epochs=7 the model look at the input at seven times ,as we the number of epochs,the weights are updated after each epochs and hence producing better results.

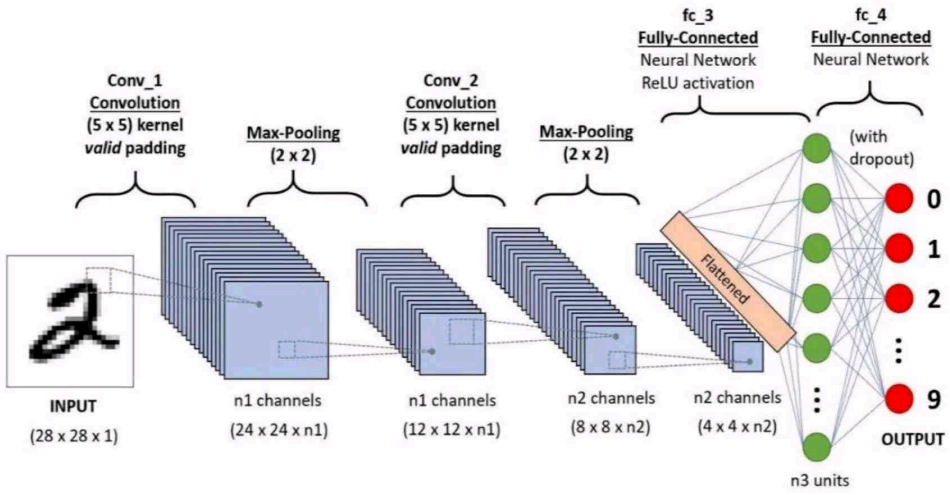


Figure 6.8 CNN Layers

To predict to in a CNN, the input is a tensor with a shape: (number of inputs) x (input height) x (input width) x (input channels). After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation.Although fully connected feed forward neural network scan be used to learn features and classify data, this architecture is generally impractical for larger inputs such as high-resolution images. It would require a very high number of neurons, even in a shallow architecture, due to the large input size of images, where each pixel is a relevant input feature. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for *each* neuron in these condlayer. Instead, convolution reduces the number of free parameters, allowing the network to be deeper. For example, regardless of image size, using a 5x5 tiling region, each with the same shared weights, requires only learnable parameters. Using regularized weights over fewer parameters avoids the Vanishing gradients and exploding gradients problems seen during back propagation in traditional neural networks. Furthermore, convolutional neural networks are ideal for data with a gridlike topology (such as images) as spatial relations between separate features are taken into account during convolution and/or pooling

6.10.8 BATCH SIZE

Accuracy of classification model is affected by the specification of batch size. Bigger batches takes more training time, it also influence the memory requirement and overall performance. Therefore, appropriate batch size selection is must to improve the quality of model. Batch sizes of 8,16,32,64 is evaluated in current study. As the batch size grows from 8 to16,the model's accuracy improves. It drops to and then drops by 64 moretimes. The figure indicates that the model performed best with a batchsize of 16 people.

Fig 6.9 Impact Batch Size

6.10.9 NUMBER OF EPOCHS

The number of runs over the complete training dataset is represented by epochs.The proposed model is trained over a period of 45 epochs.The model is trained at different numbers of epochs 5, 15, 20, and 25 after selecting the batch size learning rate 0.0001, and Adam optimizer. In 25 epochs, a classification accuracy of 90%is achieved.

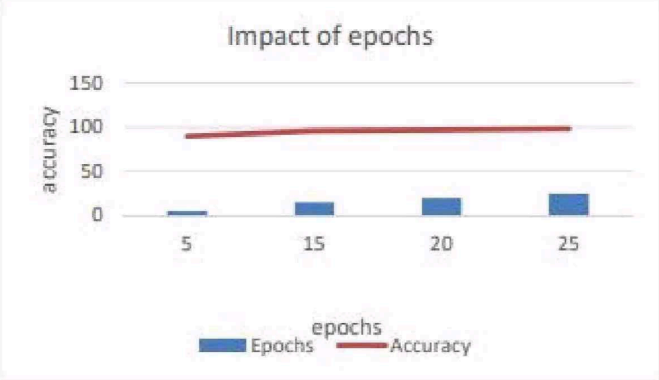


Fig 6.10 Impact of Epochs

6.10.10 OPTIMIZERS

Another important factor is to choose optimizer which optimizes model's performance. It reduces the loss function by updating the weight parameter. By changing the network's parameters, our objective is to reduce the loss of neuralnetworks. The neural network loss function is assessed by comparing

the real and predicted values. Assessment of four optimizers and accuracy comparison is doneto figure out the best optimizer. Four optimizers used in presented work are stochastic gradient descent (SGD), Adam, Adagrad, and RMS prop. Based on analysis best optimizer is found to be Adam optimizer which gives the accuracy of 98.23% where as RMS prop , SGD and Adagrad gives the accuracy of 89.19 %,24.43% and 62.65% respectively.

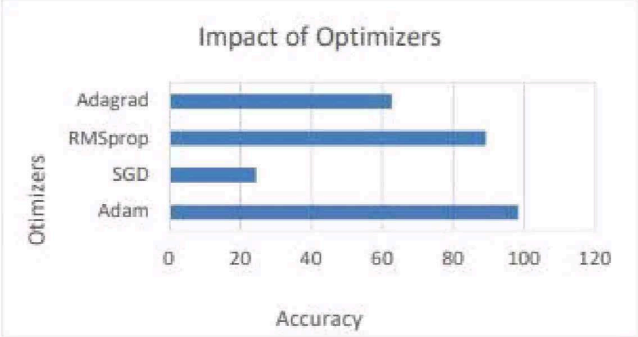


Fig 6.11 Impact of Optimizers

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 EVALUVATION AND ANALYSIS OF CNN

The following table shows how the dataset was managed into different sets. Where are two categories train and Test datasets of X-ray Images are as Given below,

Dataset of X-Ray Images	Quantity/Size
Traning datasets	9218
Testing datasets	826

Fig 7.1 data Splitting Table

The following table shows the important parameters, hyper parameters or information regarding the CNNs that would be used for model training regarding

Parameters / Hyperparameters	Value
Input shape of data/images	(150, 150, 3)
Batch size	30
Training epochs	30 or 100
Output classes	6
Loss function	Categorical Cross Entropy
Optimizer	Adam
Learning rate	0.001

Fig 7.2 Parameters With Respective Values

7.2 DATASET COLLECTIONS

First step is collecting sufficient numbers of images for the research. In the image collection step of system design, images collection is required to get value sufficient number of images for training and testing purposes.

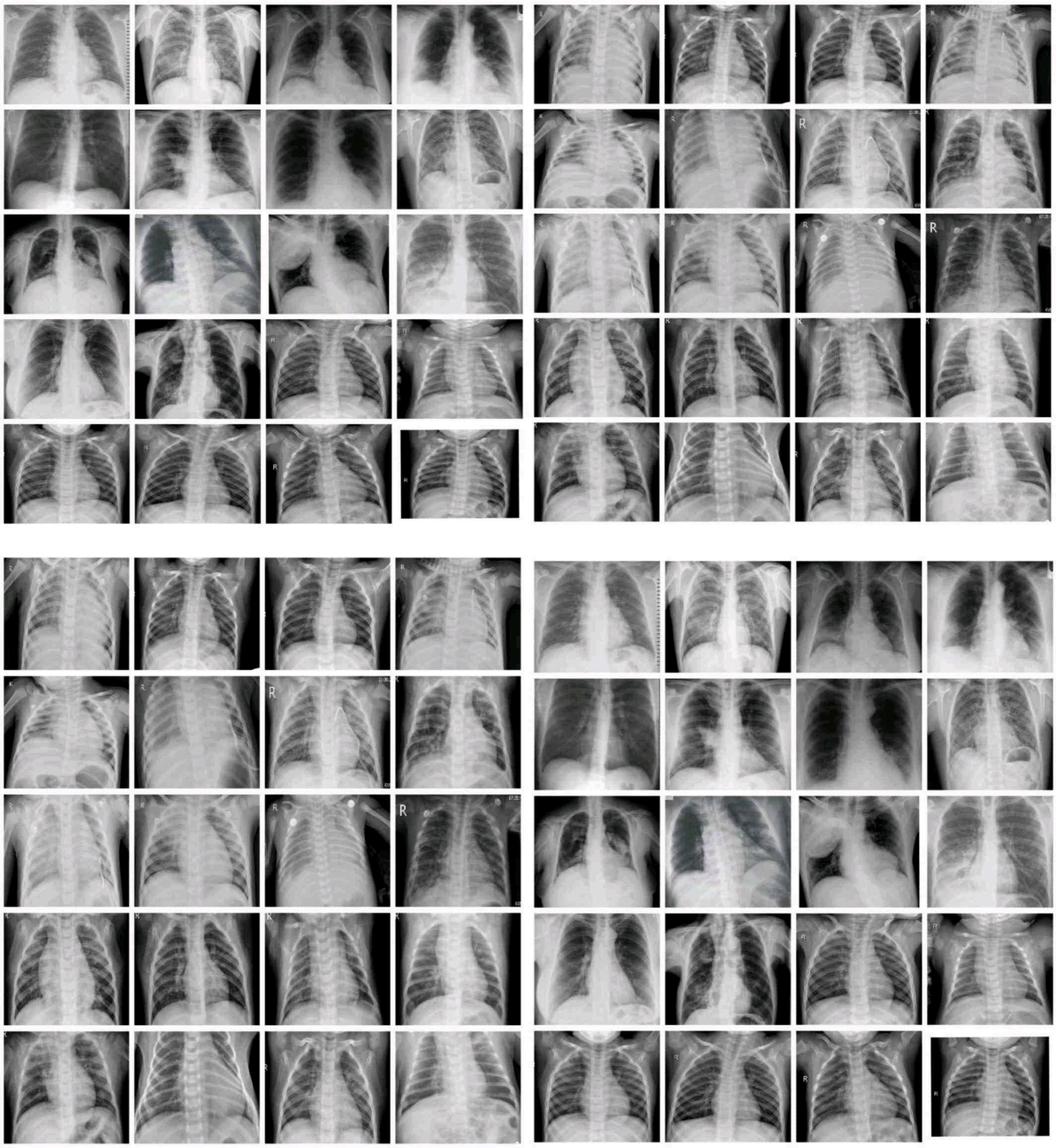


Figure 7.1 Testing And Training Datasets

At first step we have uploaded the zipped file of our dataset to our collaboratory and we have installed kaggle followed by downloading the dataset from zipped file and extract the files for further use.

7.3 BUILDING THE CONVOLUTIONAL NEURAL NETWORK:

Here we are compiling and training our model for that we imported keras layers utilized for our project such as Conv2d, maxpooling2d, dense ,dropout, padding . It uses 4 Convolution and Pooling layers followed by 2 Dense and Dropout layers to classify the images as having pneumonia or not having pneumonia.

The classification occurs after the image is fed through a series of convolutional and max pooling layers that are activated by using the ReLU activation function that is subsequently fed into the neurons present in the dense layers and finally, the output neuron is activated by the sigmoidal function. Then the model is compiled using model.compile() method, in this model,optimizer used during compilation is ‘adam’ which is responsible for manipulations in the weights and Loss function uses ‘binary cross entropy’ and metrics is chosen as ‘accuracy’.

Then proceeding with the training part where we have implemented layers of Convolutional Neural Network for training the extracted images and followed by displaying the summary of the training part. Here we have trained every nook and corner of the images in the dataset for better performance of our model and to attain a decent result prediction.

7.4 MODEL SUMMARY

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 16)      448

 max_pooling2d (MaxPooling2D  (None, 111, 111, 16)     0
 )

 conv2d_1 (Conv2D)           (None, 109, 109, 32)      4640

 max_pooling2d_1 (MaxPooling  (None, 54, 54, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 52, 52, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 26, 26, 64)       0
 2D)

 flatten (Flatten)           (None, 43264)             0

 dropout (Dropout)           (None, 43264)             0

 dense (Dense)               (None, 128)               5537920

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 4)                 516

=================================================================
Total params: 5,562,020
Trainable params: 5,562,020
Non-trainable params: 0
_________________________________________________________________
```

7.5 TRAINING THE MODEL

Then we have imported imagegenerator() for training the train dataset and test dataset by splitting them into batches and finally the accuracy and loss accuracyis obtained by running upto 25 epochs.

```
Epoch 1/25
229/229 [==============================] - 631s 3s/step - loss: 1.0443 - accuracy: 0.5761 - val_loss: 0.7525 - val_accuracy: 0.7111
Epoch 2/25
229/229 [==============================] - 626s 3s/step - loss: 0.7751 - accuracy: 0.6912 - val_loss: 0.7093 - val_accuracy: 0.7188
Epoch 3/25
229/229 [==============================] - 615s 3s/step - loss: 0.7465 - accuracy: 0.7050 - val_loss: 0.6836 - val_accuracy: 0.7346
Epoch 4/25
229/229 [==============================] - 594s 3s/step - loss: 0.7036 - accuracy: 0.7185 - val_loss: 0.6235 - val_accuracy: 0.7692
Epoch 5/25
229/229 [==============================] - 595s 3s/step - loss: 0.6743 - accuracy: 0.7289 - val_loss: 0.6194 - val_accuracy: 0.7593
Epoch 6/25
229/229 [==============================] - 617s 3s/step - loss: 0.6529 - accuracy: 0.7413 - val_loss: 0.6087 - val_accuracy: 0.7659
Epoch 7/25
229/229 [==============================] - 619s 3s/step - loss: 0.6382 - accuracy: 0.7397 - val_loss: 0.5827 - val_accuracy: 0.7769
Epoch 8/25
229/229 [==============================] - 616s 3s/step - loss: 0.6440 - accuracy: 0.7457 - val_loss: 0.5672 - val_accuracy: 0.7577
Epoch 9/25
229/229 [==============================] - 618s 3s/step - loss: 0.6275 - accuracy: 0.7455 - val_loss: 0.5876 - val_accuracy: 0.7588
Epoch 10/25
229/229 [==============================] - 616s 3s/step - loss: 0.6121 - accuracy: 0.7565 - val_loss: 0.5655 - val_accuracy: 0.7796
Epoch 11/25
229/229 [==============================] - 615s 3s/step - loss: 0.6110 - accuracy: 0.7563 - val_loss: 0.5353 - val_accuracy: 0.8043
Epoch 12/25
229/229 [==============================] - 612s 3s/step - loss: 0.5921 - accuracy: 0.7651 - val_loss: 0.5309 - val_accuracy: 0.7807
Epoch 13/25
229/229 [==============================] - 613s 3s/step - loss: 0.5859 - accuracy: 0.7609 - val_loss: 0.5454 - val_accuracy: 0.7621
Epoch 14/25
229/229 [==============================] - 608s 3s/step - loss: 0.5864 - accuracy: 0.7564 - val_loss: 0.5387 - val_accuracy: 0.7664
Epoch 15/25
229/229 [==============================] - 607s 3s/step - loss: 0.5825 - accuracy: 0.7623 - val_loss: 0.5415 - val_accuracy: 0.7900
Epoch 16/25
229/229 [==============================] - 605s 3s/step - loss: 0.5666 - accuracy: 0.7721 - val_loss: 0.5318 - val_accuracy: 0.7856
Epoch 17/25
229/229 [==============================] - 621s 3s/step - loss: 0.5552 - accuracy: 0.7717 - val_loss: 0.5248 - val_accuracy: 0.7752
Epoch 18/25
229/229 [==============================] - 635s 3s/step - loss: 0.5434 - accuracy: 0.7787 - val_loss: 0.4851 - val_accuracy: 0.8273
Epoch 19/25
229/229 [==============================] - 634s 3s/step - loss: 0.5381 - accuracy: 0.7760 - val_loss: 0.4745 - val_accuracy: 0.8081
Epoch 20/25
229/229 [==============================] - 606s 3s/step - loss: 0.5409 - accuracy: 0.7796 - val_loss: 0.5155 - val_accuracy: 0.8032
Epoch 21/25
229/229 [==============================] - 612s 3s/step - loss: 0.5323 - accuracy: 0.7794 - val_loss: 0.4738 - val_accuracy: 0.8207
Epoch 22/25
229/229 [==============================] - 605s 3s/step - loss: 0.5305 - accuracy: 0.7796 - val_loss: 0.4519 - val_accuracy: 0.8235
Epoch 23/25
229/229 [==============================] - 610s 3s/step - loss: 0.5142 - accuracy: 0.7898 - val_loss: 0.4565 - val_accuracy: 0.8048
Epoch 24/25
229/229 [==============================] - 618s 3s/step - loss: 0.5256 - accuracy: 0.7835 - val_loss: 0.4778 - val_accuracy: 0.7939
Epoch 25/25
229/229 [==============================] - 612s 3s/step - loss: 0.5194 - accuracy: 0.7863 - val_loss: 0.4852 - val_accuracy: 0.8070
```

7.6 EXPERIMENTAL SETUP

In this section, we will usually analyze the result of the proposed model of ourwork and explain the result of our work with proper logic. We trained our modelusing Tensor flow 2.0.1. Setting hyper parameters to our model was the first step.Define Batch size as 45. The ratio of our training and testing data as follow 80%,20%.After that we compiled our model using adam[6]optimizer and our learning rate was 0.0001.Then fit ()used for start the training.

7.7 PERFORMANCE EVALUATION

After finishing training our model got an accuracy of 95.04% with 50 epoch. The accuracy curve of our model where we observed our training accuracy 95.04% and validation accuracy 96.21%.

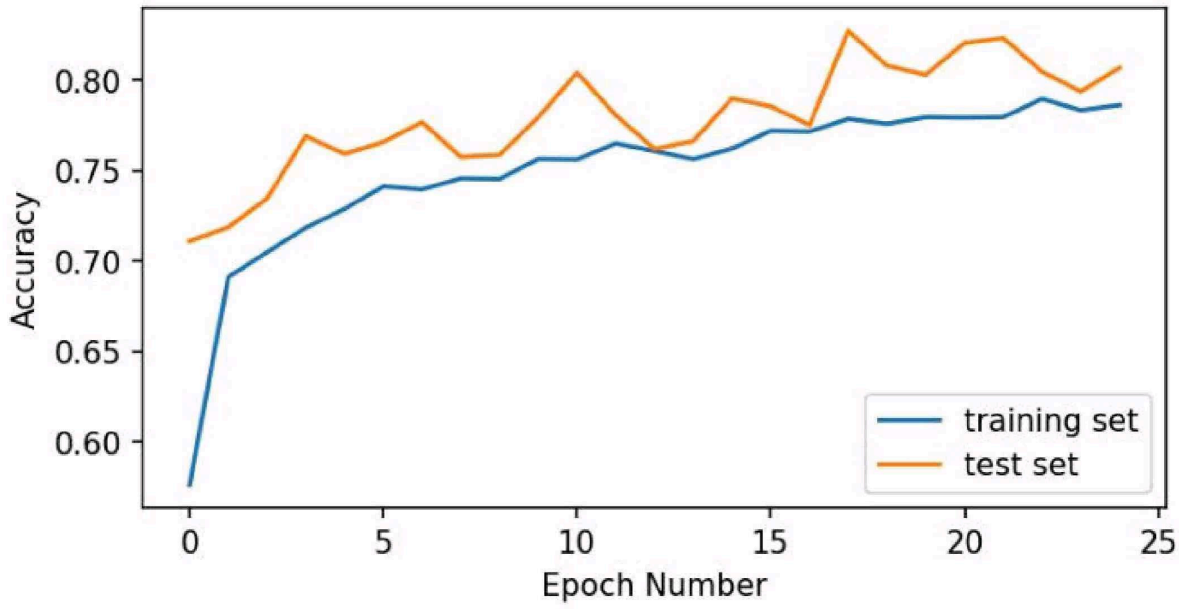


Figure 7.4 Training And Test Accuracy

The Figure represents the loss curve of our model. Where we observed our training loss 0.0545 and validation loss is 0.0797 .

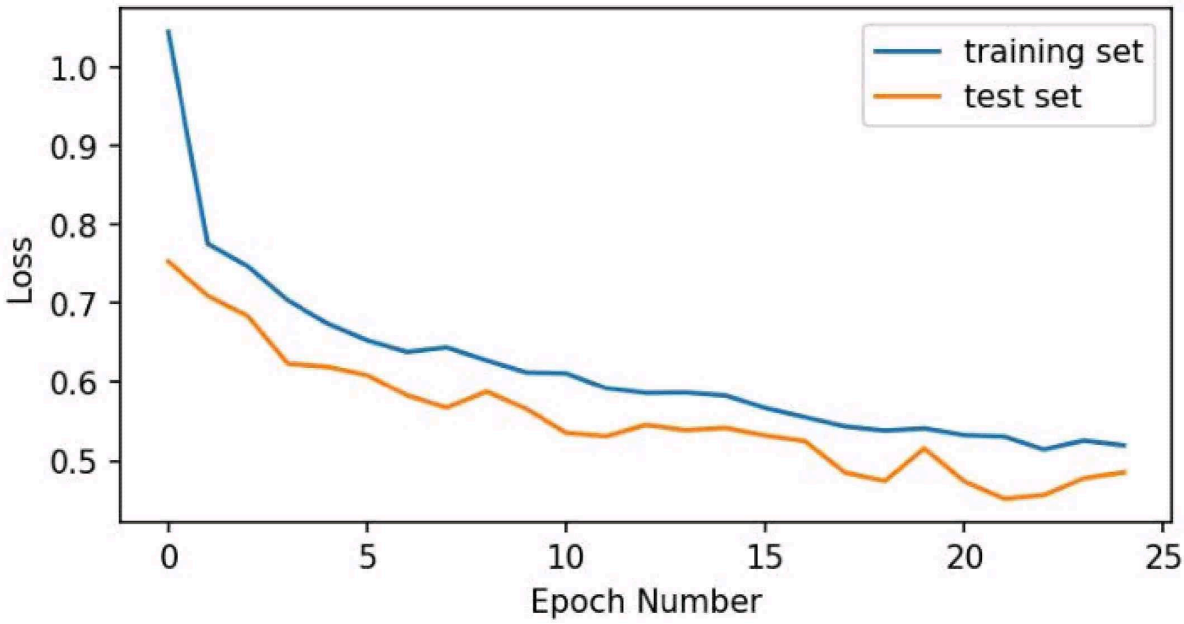


FIG 5.3 Training vs Test Loss

7.8 CNN WITH IMAGE AUGMENTATION

Image augmentation is also a regularization method to improve the proficiency of CNN models. It artificially re-processes and creates training images through different processing methods in order to increase the amount used of data for model training as neural networks with larger training data tends to achieve desirable performance. Hence, Image Data Generator API in Keras was used as the configurations were easy to implement for training data. The configurations applied.

CHAPTER 8

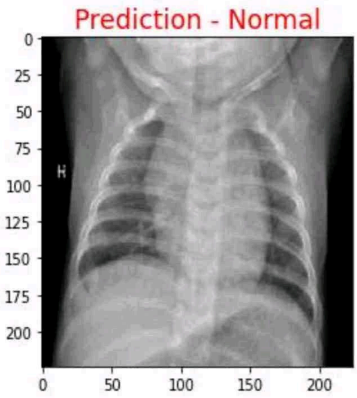
EXPERIMENTAL RESULTS

8.1 TESTING AND PERFORMANCE OF CLASSIFICATION MODEL

Images were found randomly from internet sources to test on the final into a classification model, the pneumonia or not were expected prediction outcome. In this case, four images were normal, viral, bacterial and covid-19 chosen randomly for testing.

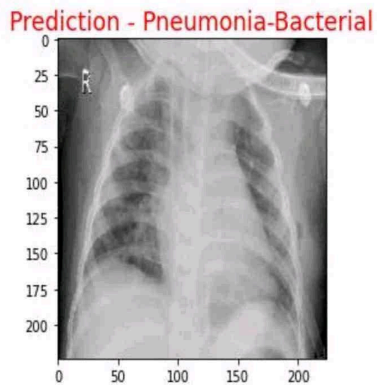
8.2 PREDICTION NORMAL

```
[ ] predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Training/Normal/Normal (1).jpg', model)

    [[0.00729293 0.50804627 0.23139498 0.25326586]]
```



8.3 PREDICTION BACTERIA-PNEUMONIA

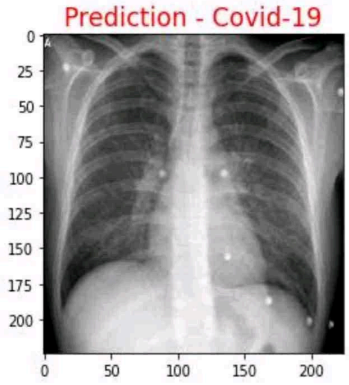
```
[ ] predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Training/Pneumonia-Bacterial/Pneumonia-Bacterial (1).jpg',

    [[0.18119171 0.05354968 0.600051   0.16520762]]
```



8.4 PREDICTION COVID-19 PNEUMONIA

```
[ ] predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Training/COVID-19/COVID-19 (1).jpg', model)
```

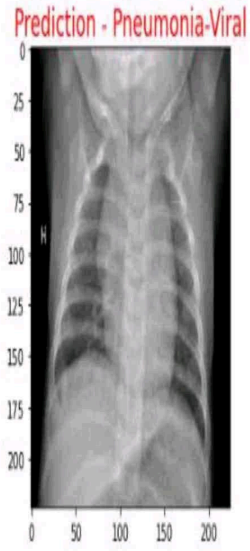
```
[[0.72684735 0.11593869 0.11508192 0.04213204]]
```



8.5 PREDICTION VIRAL PNEUMONIA

```
[ ] predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Training/Normal/Normal (1).jpg', model)
```

```
[[0.00058491 0.13002497 0.36929202 0.50009805]]
```



CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

The main purpose of our research was to find out three types of pneumonia from a given image. We have seen that CNN classify images very accurately in our problem. For this study, we first collected data from different places. Then we do preprocessing and then our job done by dividing the training and testing. Later we got our desire results. Performance analysis of automated pneumonia detection from X-Ray imaging using basic image processing techniques based on various hard and soft computing has been performed in our work. Then we applied CNN for lung pneumonia detection to include deep learning method in our work. We compared the result of the traditional one having the best accuracy with the result of CNN. In the context of the full dataset, it is necessary to parallelize and utilize high-performance computing platform for maximum efficiency. We tried our best to detect the pneumonia accurately but, nevertheless we faced some problems in our work where pneumonia could not be detected or falsely detected. So, we will try to work on those images and on the complete dataset. Hence, we will try to apply other deep learning methods in the future so that we can get a more precise and better result.

9.2 FUTURE WORK

Primary destinations for is to improve the exactness of the neural organization. To build up a more robust and exact organic product newness appraisal profound learning model, as a typical profound learning practice, an enormous volume of source information is required. As CNN requires a vast amount of data I will add more data to make this model more effective. That will use more class with better accuracy, that will make a web app & android application in future.

REFERENCES

1. COVID-19 Pneumonia and Influenza Detection Using CNN convolutional Neural Networks, Julianna Antonchuk.,, 2021.

2. Pneumonia Detection Using Convolutional Neural Networks, Luka racic., Tomo popovic, 2021.

3. Pneumonia Detection Using Convolutional Neural Networks, S. Shah, H et al, 2020.

4. Deep Learning For Automatic Pneumonia, Alexandr A.Kalin, 2020

5. Pneumonia Detection Using Deep Learning Approaches math A.Tilve, S. Nayak et al, 2020.

6. Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning, E. Ayan et al, 2019.

7. Pneumonia Detection Using CNN based on Feature Extraction, Y.Karthick Tharkal, 2019.

8. CNN based Pneumonia Detection Using Feature Extraction, Varshni,. 2019.

9. Optimized CNN based Diagnosis System to Detect Pneumonia from Chest Radiographs, Mohhammed Aledhari., 2016.

10. Preliminary study of Pneumonia symptoms detection method using cellular Neural Networks. A. Abdullah, et al., 2011.

APPENDIX

SAMPLE CODE

```
!pip install -q visualkeras
!pip install -q ann_visualizer
!pip install -q dtreeviz


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


import tensorflow as tf
from tensorflow import keras


from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Activation
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from keras import regularizers

from tensorflow.keras.preprocessing import image
import visualkeras
from ann_visualizer.visualize import ann_viz
from dtreeviz.trees import *
from tensorflow.keras.utils import plot_model

import warnings
warnings.filterwarnings('ignore')

from google.colab import drive
drive.mount('/content/drive')
```

```
data_dir = '/content/drive/MyDrive/Colab Notebooks/pneumonia/Training'
data = tf.keras.preprocessing.image_dataset_from_directory(data_dir)


datagen = ImageDataGenerator(
        rescale = 1./255,
        rotation_range = 40,
        width_shift_range = 0.2,
        height_shift_range = 0.2,
        shear_range = 0.2,
        zoom_range = 0.2,
        horizontal_flip = True,
        fill_mode = 'nearest',
        validation_split = 0.2)

height = 224
width = 224
channels = 3
batch_size = 32
img_shape = (height, width, channels)
img_size = (height, width)


train_data = datagen.flow_from_directory(
    data_dir,
    target_size = img_size,
    batch_size = batch_size,
    class_mode = 'categorical',
    subset = 'training')

val_data = datagen.flow_from_directory(
    data_dir,
    target_size = img_size,
    batch_size = batch_size,
    class_mode='categorical',
    subset = 'validation')


def plotImages(image_arr):
    fig,axes = plt.subplots(1, 5, figsize=(20,20))
    axes = axes.flatten()
    for img,ax in zip(image_arr,axes):
        ax.imshow(img)
    plt.tight_layout()
    plt.show()

img_array = [train_data[0][0][0] for i in range(6)]
plotImages(img_array)
```

```
num_classes = len(data.class_names)
print('.... Number of Classes : {0} ....'.format(num_classes))
model = Sequential()
model.add(Conv2D(16, (3,3), input_shape=(224,224,3), activation="relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Conv2D(32, (3,3), activation="relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Conv2D(64, (3,3), activation="relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dropout(0.2))
model.add(Dense(128,activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(4, activation="softmax"))

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

model.summary()


STEP_SIZE_TRAIN = train_data.n // train_data.batch_size
STEP_SIZE_VALID = val_data.n // val_data.batch_size

history = model.fit_generator(train_data,
                    steps_per_epoch = STEP_SIZE_TRAIN,
                    validation_data = val_data,
                    validation_steps = STEP_SIZE_VALID,
                    epochs = 2,
                    verbose = 1)


model.save('model.h5')


plot_model(model, to_file='model.png', show_shapes=True, show_layer_nam
es=True)


plt.figure(figsize=(6, 3), dpi=150)
plt.xlabel('Epoch Number')
plt.ylabel('Loss')
plt.plot(history.history['loss'], label='training set')
plt.plot(history.history['val_loss'], label= 'test set')
plt.legend()
plt.figure(figsize=(6, 3), dpi=150)
```

```
plt.xlabel('Epoch Number')
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'], label= 'training set')
plt.plot(history.history['val_accuracy'], label='test set')
plt.legend()


ann_viz(model, view=True, title='Custom model')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import image

import warnings
warnings.filterwarnings('ignore')

classes = ['COVID-19','Normal','Pneumonia-Bacterial','Pneumonia-Viral']

from tensorflow.keras.models import load_model
model=load_model('model.h5')

def predict_image(filename, model):
    img_ = image.load_img(filename, target_size=(224, 224))
    img_array = image.img_to_array(img_)
    img_processed = np.expand_dims(img_array, axis=0)
    img_processed /= 255.
    prediction = model.predict(img_processed)
    print(prediction)
    index = np.argmax(prediction)
    plt.title("Prediction -
 {}".format(str(classes[index]).title()), size=18, color='red')
    plt.imshow(img_array)


predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Trainin
g/COVID-19/COVID-19 (1).jpg', model)


predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Trainin
g/Normal/Normal (1).jpg', model)
predict_image('/content/drive/MyDrive/Colab Notebooks/pneumonia/Trainin
g/Pneumonia-Bacterial/Pneumonia-Bacterial (1).jpg', model)
```