

DIAGNOSING PNEUMONIA USING CNN

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering degree in Computer Science and Engineering

by

C SAI SARAN (38110096)

P AJITH (38110434)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI - 600 119**

APRIL – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHANDAVARAPU SAI SARAN(38110096),PURAM AJITH(38110434)** who carried out the project entitled "**DIAGNOSING PNEUMONIA USING CNN**" under my supervision from NOV 2021 to APRIL 2022.

Internal Guide

Mrs.N.S.Usha,B.E.,M.E.,(Ph.D)

Head of the Department

**Dr.L.LAKSHMANAN,M.E.,PhD.,
Dr.S.VIGNESHWARI,M.E.,PhD.**

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I **C.SAI SARAN (Reg No:38110096)** and **P. AJITH(Reg No: 38110434)** hereby declare that the Project Report entitled “**DIAGNOSING PNEUMONIA USING CNN**” done by us under the guidance of **Mrs. N.S. USHA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D, Dean**, School of Computing **Dr.L.Lakshmanan M.E., Ph.D** and **Dr.S.Vigneshwari M.E., Ph.D** Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Mrs.N.S.Usha,B.E,M.E,(Ph.D),CSE** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Pneumonia is one of the serious diseases that is caused by a bacterial or viral infection of the lungs and has the potential to result in severe consequences within a short period. Therefore, early diagnosis is a key factor in terms of the successful treatment process. Thus, there is a need for an intelligent and automatic system that has the capability of diagnosing chest X-rays and to simplify the pneumonia detection process for experts as well as for novices. This study aims to develop a model that will help with the classification of chest x-ray medical images into normal(healthy) vs abnormal(sick). To achieve this, seven existing state-of-the-art machine learning techniques and well-known convolutional neural network models have been employed to increase efficiency and accuracy. In this study, we propose our machine learning for the classification task, which is trained with modified images, through multiple steps of preprocessing. Experimentally, it demonstrated that the machine learning technique for the classification task performs the best, compared to the other seven machine learning techniques. In this study, we successfully classified chest infection in chest predict using on CNN with an overall accuracy of 98.46%. It achieved a more successful result in detecting pneumonia cases. Machine learning models. In this study, we propose a method for image-based diagnosis for Pneumonia leveraging deep learning techniques and interpretability of explanation models such as Local Interpretable Model-agnostic Explanations and Saliency maps. We experiment on a variety of sizes and Convolutional neural network architecture to evaluate the efficiency of the proposed method on the set of Chest x-ray images. The work is expected to provide an approach to distinguish between healthy individuals and patients who are affected by Pneumonia as well as differentiate between viral Pneumonia and bacteria Pneumonia by providing signals supporting image-based disease diagnosis approaches. . This study aims to develop a model that will help with the classification of chest x-ray medical images into normal(healthy) vs abnormal(sick). To achieve this, seven existing state-of-the-art machine learning techniques and well-known convolutional neural network models have been employed to increase efficiency and accuracy. In this study, we propose our deep learning for the classification task, which is trained with modified images, through multiple steps of preprocessing. Experimentally, it demonstrated that the deep learning technique for the classification task performs the best, compared to the other seven

machine learning techniques. In this study, we successfully classified chest infection in chest Xray images using deep learning based on CNN with an overall accuracy of 98.46%. It achieved a more successful result in detecting pneumonia cases.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
1	INTRODUCTION	10
2	LITERATURE SURVEY	13
3	AIM AND SCOPE	16
4	IMPLEMENTATION AND ALGORITHM USED	17
	4.1 EXISTING SYSTEM	17
	4.2 PROPOSED SYSTEM	17
	4.3 BLOCK DIAGRAM	18
	4.4 FLOW DIAGRAM	18
	4.5 MODULES	19
	4.5.1 GATHERING DATA	19
	4.5.2 DATA PREPROCESSING	20
	4.5.3 RESEARCHING	22
	4.5.4 TRAINING AND TESTING	26
	4.5.5 EVALUTION	28
	4.6 ALGORITHM USED	29
	4.6.1 CNN ALGORITHM	29
	4.6.2 CONVOLUTION LAYER	30
	4.6.3 STRIDES	30

	4.6.4 PADDING	30
	4.6.5 POOLING LAYER	30
	4.6.6 MAX POOLING	30
	4.6.7 AVERAGE POOLING	31
	4.6.8 FULLY CONNECTED LAYER	31
5	RESULTS & DISCUSSION	32
6	SUMMARY & CONCLUSION	33
	A.REFERENCES	34
	B.OUTPUT SCREENSHOTS	36

LIST OF FIGURES

FIGURE NO.	NAME	PAGE NO.
4.3	BLOCK DIAGRAM	18
4.4	FLOW DIAGRAM	18
4.5.3.1	SUPERVISED LEARNING	22
4.5.3.2	UNSUPERVISED LEARNING	22
4.5.3.3	CLASSIFICATION	23
4.5.3.4	GRADIENT SEARCH	24
4.5.3.5	POINTS AND LINE	24
4.5.3.6	UNCLUSTERED DATA	25
4.5.3.7	CLUSTERED DATA	25
4.5.3.8	MODULE CATEGORIES	26
4.5.4.1	TESTING CLASSIFIER	26
4.5.4.2	CLASSIFIER PARAMETERS	27
4.5.4.3	PREDICTION TABLE	28
4.6.1	CLASSIFICATION USING CNN	30
4.6.8	CONVOLUTION NEURAL NETWORK	32
B.1	PNEUMONIA POSITIVE	36
B.2	PNEUMONIA NEGATIVE	36

CHAPTER 1

INTRODUCTION

According to the World Health Organization (WHO), pneumonia is one of the most infectious causes of death worldwide, it affects children and families everywhere and causes 50 thousand deaths each year. Recently, the Situation Report - 150 from WHO about COVID-19 presented the number of infected active cases is up to 8, 2 million, the number of deaths is 445, 535. The patients can get pneumonia as a complication of viral infections such as COVID-19 or the common flu. Besides, the bacteria, fungi, and other microorganisms can also be the primary infectious agents of pneumonia, causing cough with phlegm or pus, fever, chills, and difficulty breathing. Pneumonia is the infection of one or both lungs and filled up with fluid and pus. Based on , up to 60% of the cases are related to respiratory virus infections. The study in indicated the difference between viral and bacterial pneumonia in children is based on the serum C reactive protein (CRP) but the sensitivity is not enough for use in clinical practice. In the field of medicine, diagnostic radiology is significant and used for disease assessment.

To detect the pneumoconiosis early and accurately after the clinical analysis, performing a chest X-ray is important and necessary. It is critical to preventing complications including death. The expert radiologists assessed the X-ray images for the pneumonia fluid in the lungs during the diagnosis. Specifically, contains images of viral (left image), bacterial (middle image) and normal chest (right image) pneumonia. Viral pneumonia presents the diffuse interstitial pattern in both lungs, whereas, bacterial pneumonia typically exhibits a focal lobar consolidation. Furthermore, chest X-ray (CXR or chest radiography) can reveal the abnormalities areas and not only produce images of the chest but also the nearby structures. Nevertheless, the X-ray images consist of black and white colors, it is quite difficult for detecting the infected areas in the images. Additionally, the technical level of radiologists is also important to make the diagnosis correctly. A study in conducted an education in person training for improving chest radiograph interpretation accuracy among non-radiologists clinicians. In recent years, the complexity of medical data makes it more difficult for analyzing and diagnosing the disease. In parallel, the improvements in Machine Learning and Deep Learning have a

certain influence on image processing in general and medicine in particular. The diagnosis process performs with Machine Learning or Deep Learning can help physicians investigate the medical images conveniently and reduce the analysis time. Several studies have resolved the challenging tasks such as medical image classification , skin cancer detection using images , or 3D image biomedical segmentation.

Moreover, Deep Learning-based technologies have successfully demonstrated in clinical practice including clinical decision support systems (CDSS), diagnosis prediction, and predicting the invasiveness of lung adenocarcinoma manifesting based on radionics and clinical features. Though, there are still several challenges with Machine Learning. Selecting a dataset, creating a predictive model, and evaluating and refining the model, the most important thing is data. The implementations of Machine Learning or Deep Learning in health care are influenced by the accuracy of medical data. Specifically, the annotation progress in the medical image is based on medical professional knowledge, medical industry standard, and medical system Pneumonia is one of the serious diseases which cause most of the deaths in adults globally. According to Health Metrics and Evaluation (IHME), the highest pneumonia mortality rates in 2017 were among people aged 70 and older. More than 1.13 million pneumonia-related deaths are reported every year were in this age group . Pneumonia is a disease of infectious origin that causes inflammation in the air sacs or alveoli of one or both lungs The air sacs get filled with fluid which can cause difficulty in breathing and an over-generation of mucus and sputum as shown in . Pneumonia is most commonly caused by viruses or bacteria and, less commonly, other microorganisms]. Nowadays, chest X-ray (CXR) imaging is used commonly for health intensive care and analysis of many lung diseases such as pneumonia, cancer because of relatively low prices. Therefore, Chest X-rays are currently the best available tool for diagnosing pneumonia which has played a huge role in clinical care and epidemiological research However, detecting pneumonia in chest X-rays is a challenging task that relies on the availability of expert radiologists. Accurate image analysis and image interpretation are very crucial for a better diagnosis. Though image interpretation by conventional machine learning algorithms depends mostly on expertly crafted features, computer vision is a very effective machine learning application. However, not all doctors have good diagnosing tools to diagnose patients. As a result, sometimes their diagnosing is not very accurate. It is also much more difficult to judge pneumonia just by looking at

the chest X-rays images.

Therefore, more specific diagnostic tools must be developed which are cheap and accurate in diagnosing pneumonia. With the advancements of information technology, machine learning algorithms have been successfully applied to many healthcare problems and have explained complex relationships, and improved clinical predictions. In recent years, several researchers have proposed different artificial intelligence (AI)- based solutions for different medical problems. techniques based on Convolutional neural networks (CNNs) have allowed researchers to obtain successful results in wide medical applications, including classifying skin cancer from skin photographs, detecting metastasis on pathological images and disease classification in X-ray images all of which demonstrated expert-level diagnostic accuracy. As a result of this trend, numerous studies have emerged that analyzes the influence of deep learning and convolutional neural networks in the field of health and imaging medical diagnosis. Specifically, due to successes in other fields especially in medical problems, deep learning algorithms such as Convolutional neural networks (CNNs) have recently been applied for image classification tasks. However, algorithms in those studies have not yet been fully validated in various data sets, limiting the generalization of the results. Classification methods are among the most commonly used techniques in medical imaging, where the goal is building classifiers that are capable of predicting whether x-ray images are normal or show the presence of Pneumonia. Especially, this study tries to answer the following research question: Could machine learning algorithms be used to assist in the diagnosis of whether a patient has pneumonia or not by looking at chest X-ray images? This inspires us to use machine learning algorithms to predict pneumonia based on the x-ray images. Moreover, it would provide doctors with immediate information about the patient's condition and risk level to recommend more diagnostic tests without delay. Thus, the investigative and predictive methods, such as machine learning algorithms, are vital to make smart decisions that will help doctors and radiologists to get more information to prevent themselves from misdiagnosing a patient. The primary objective of this study was to examine and evaluate the performance of machine learning techniques and convolutional neural network (CNN) for the classification of pneumonia, which is based on Chest X-ray Images to achieve high accuracy. Other objective is to provide radiologists and medical experts with a tool which is a lower cost. This tool will be helping radiologists and medical experts to identify the slit and slow changes among.

CHAPTER 2

LITERATURE SURVEY (change the contents)

The recent studies on diagnosing Pneumonia by use of deep learning processing model based on CNN by different authors has been listed in this section.

Fawole et. al and ResNet-50 have concentrated inside and out X-X imaging and changed undetectable articles like patient age, vision, and perceivability. also the vector related with the picture. Up to 15 percent of beneficial things require one look.

.Ruuskanen O, et. al The report incorporates 247 radiograms from the Japan Society for Radiology (JSRT). The first picture was diminished to 96 96 and the Log channel was utilized prior to preparing or testing. The presentation of the inside and out preparing technique is surveyed in light of the ID of the knobs on the radiograph, and the outcomes are analyzed.

Juan G.; et. al. Recent work has shown that a round network is profound, finished, and proficient in preparing assuming it contains a short circle between the inbound and outbound lines. In this article, we will investigate the DenseNet network that associates every classification to another level. Conventional L-broken pipes have a L association between each stage and the following stage, while our organization has a straight line $(L + 1)/2$. For each stage, the guide of the multitude of past areas is utilized as an info, and the information picture is utilized as an info. DenseNets has various

significant benefits: it works with the absence of slope issues, fortifies execution conveyance, upholds execution improvement, and altogether lessens execution. We are assessing our development for four capacities (CIFAR- 10, CIFAR-100, SVHN, and ImageNet) to make the data more exact. DenseNets requires a ton of memory and figuring to accomplish superior execution while accomplishing extraordinary turn of events and exceptional innovation across numerous gadgets.

IMAGE QUALITY AFFECTS DEEP LEARNING RECONSTRUCTION OF MRI

Jilani, et. al Magnetic reverberation imaging (MRI) will in general recognize a wide scope of articles from various points. Now and again, the outcomes can be mistaken for pathology. What's more, present day MR recuperation calculations have a repetitive nature that builds recuperation time. As of late, a top to bottom review technique has been utilized in MRI reproduction to give excellent rapid imaging. Inside and out study is generally subject to learning data, so the nature of learning ought not be neglected. This article talks about what noisy pictures in media practice mean for the construction of MR reproduction. The normal technique utilizes a weighty misfortune to change the capacity of the line to make the picture more appealing. This page utilizes a similar blunder free stretch; however, this technique can be reached out to other misfortune making exercises. Utilizing fixed heart MRI, this technique is contrasted with the current strategy for eliminating clamor delicate exercise data or overlooking the positive distinction. Indeed, even essential systems will work on the nature of profound gaining from this methodology.

A DEEP CASCADE OF CONVOLUTIONAL NEURAL NETWORKS FOR DYNAMIC MR IMAGE RECONSTRUCTION

Hlemper et. al First of all, we expect that the normal technique for reproducing each 2D picture autonomously is superior to the cutting edge strategy for disentangling the 2D picture, in particular by concentrating on the jargon in view of the MR picture of the design, by mistake and blunder .. Second, When we reproduce a progression of edges together, CNN shows that we can actually learn space-time designs by organizing ends and trading data. We show that the arranged technique is more present day than the current one and that the physical construction can be dependably kept up with for as

long as 11 days. Also, the recuperation is extremely quick: each full casing can be reproduced in under 10 seconds, and in 2-D, each picture can be reproduced in ms 23, which helps continuously use.

CHAPTER 3

AIM & SCOPE OF THE PROJECT

Pneumonia is one of the serious diseases that is caused by a bacterial or viral infection of the lungs and has the potential to result in severe consequences within a short period. There is a need for an intelligent and automatic system that has the capability of diagnosing chest X-rays and to simplify the pneumonia detection process for experts as well as for novices.

This project aims to develop a model that will help with the classification of chest x-ray medical images into normal healthy vs abnormal sick. To achieve this, seven existing state-of-the-art machine learning techniques and well-known convolutional neural network models have been employed to increase efficiency and accuracy. In this project, we propose our machine learning for the classification task, which is trained with modified images, through multiple steps of preprocessing.

CHAPTER 4

IMPLEMENTATION AND ALGORITHMS USED

4.1 EXISTING SYSTEM

- Data Processing. In 2018, the Radiological Society of North America (RSNA) [b20] released a dataset on the detection and localization of pneumonia in chest X-rays .
- In the Det Net network, the first four phases of the backbone are consistent with the original ResNet50 phase,
- To enhance the pictorial information output of the Det Net network, this paper combines FPN and Det Net to enhance the feature-extraction mode of the network.

4.1.1Disadvantages

- low processing method.
- Result of the system is not perfect.
- Less efficiency.
- Slow process to predict the pneumonia.
- Using some other algorithm

4.2 PROPOSED SYSTEM

- We introduce in detail our proposed Deep Conv-Dilated Net method, including the data processing, the architecture of our network, and the effective enhancement effect of Soft-NMS.
- The Chest X-Ray images which include normal chest X-Ray and Pneumonia chest X-Ray. Here, we have taken 5,856 sample images of the dataset to use later on for

recognition and classification of pneumonia. 1,583 of which are normal X-Ray lung images, and 4,273 of which are X-Ray lung images with Pneumonia from patients.

- The most important part of machine learning is data and data must be clean for models to process it. When it comes to image data, there are some preparation methods. Various types of pre-processing tasks such as dimension reduction, image resize, and image cropping is applied.
- Then it will process CNN algorithm to identify the pneumonia.

4.2.1 Advantage

- Using CNN algorithm to identify the pneumonia by using the x-ray.
- This system predicts the pneumonia accurately.
- We have to collect the previous data to predict the future data by using of CNN algorithm so the result is perfect to predict.
- To train the data we have using decision tree.
- Then apply the CNN algorithm for result.

4.3 SYSTEM ARCHITECTURE

The architectural diagram of the proposed model is given in Figure 4.1



Fig 4.1 System Architecture

The process flow carried out by the proposed model is given in Figure 4.2

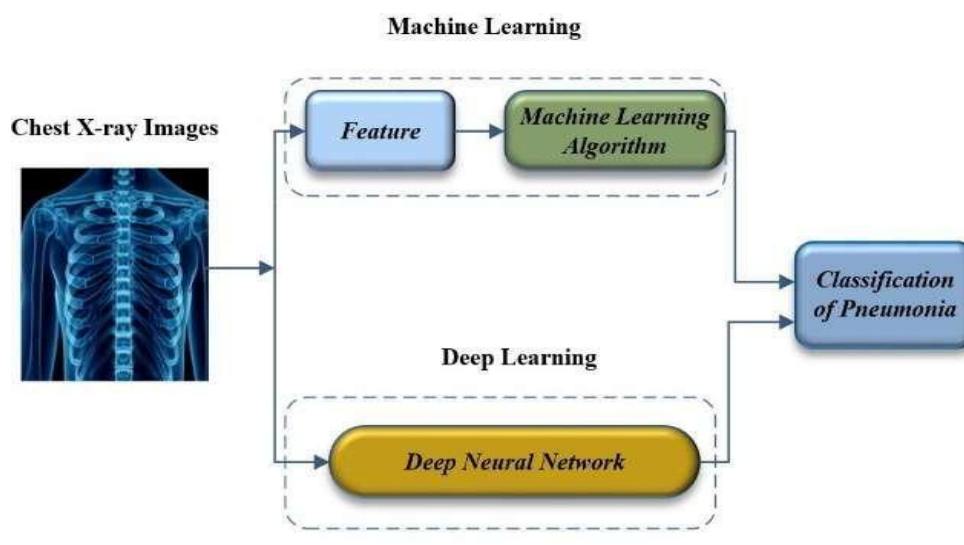


Fig 4.2 Flow Diagram

4.4 MODULES

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

4.4.1 Gathering Data

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected

data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done. Some free data sets which are present on the internet are Kaggle and UCI Machine learning Repository, the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

4.4.2 Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.

Data Preprocessing

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and it cannot be analysed. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects. Most of the real-world data is messy, some of these types of data are:

1. ***Missing data:*** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

2. **Noisy data:** This type of data is also called outliers, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

Three Types of Data

1. Numeric e.g. income, age
2. Categorical e.g. gender, nationality
3. Ordinal e.g. low/medium/high

These are some of the basic pre — processing techniques that can be used to convert raw data.

1. **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.

2. **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn't be performed if there are a lot of missing values in the dataset.

3. **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.

4. **Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.

5. **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0]

4.4.3 Researching the model that will be best for the type of data

Our main goal is to train the best performing model possible, using the pre-processed data.

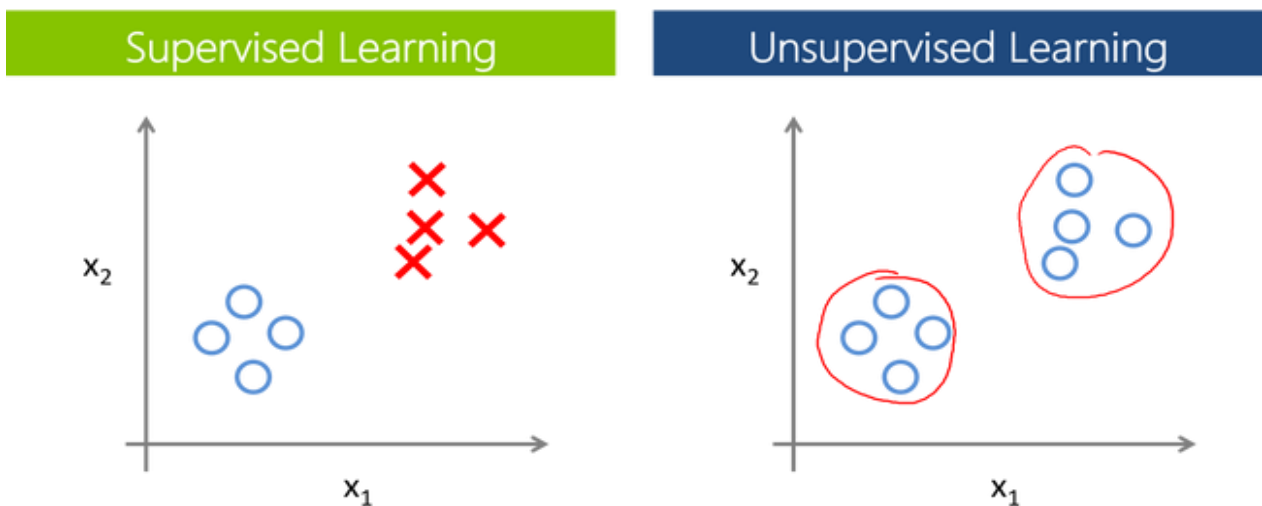


Fig 4.3 Supervised and Unsupervised Learning

In Supervised learning, an AI system is presented with data which is labelled, which means that each data is tagged with the correct label. The supervised learning is categorized into 2 other categories which are “**Classification**” and “**Regression**”.

Classification

Classification problem is when the target variable is **categorical** (i.e. the output could be classified into classes — it belongs to either Class A or B or something else). A classification problem is when the output variable is a category, such as “red” or “blue”, “disease” or “no disease” or “spam” or “not spam”.

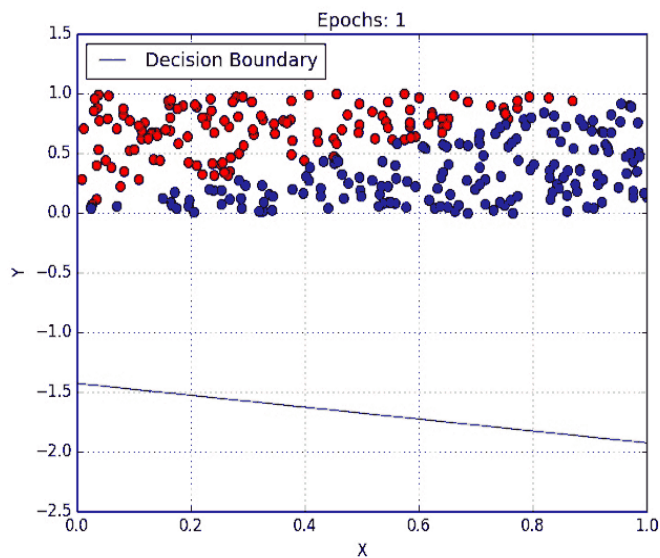


Fig 4.4 *Classification of data given in dataset*

As shown in the above representation, we have 2 classes which are plotted on the graph i.e. red and blue which can be represented as 'setosa flower' and 'versicolor flower', we can image the X-axis as ther 'Sepal Width' and the Y-axis as the 'Sepal Length', so we try to create the best fit line that separates both classes of flowers.

These some most used classification algorithms.

- K-Nearest Neighbor
- Naive Bayes
- Decision Trees/Random Forest
- Support Vector Machine
- Logistic Regression

Regression

While a **Regression** problem is when the target variable is **continuous** (i.e. the output is numeric).

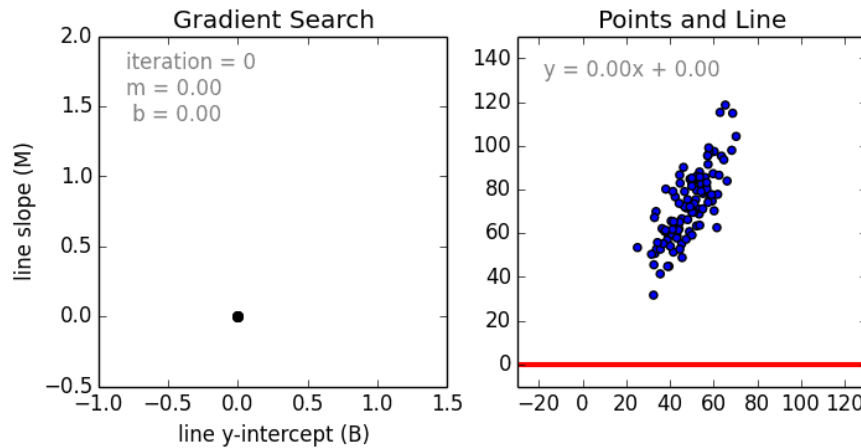


Fig 4.5 Comparison of gradient search

As shown in the above representation, we can imagine that the graph's X-axis is the 'Test scores' and the Y-axis represents 'IQ'. So we try to create the best fit line in the given graph so that we can use that line to predict any approximate IQ that isn't present in the given data.

These some most used regression algorithms.

- Linear Regression
- Support Vector Regression
- Decision Tress/Random Forest
- Gaussian Progresses Regression
- Ensemble Methods

Unsupervised Learning

The unsupervised learning is categorized into 2 other categories which are "Clustering" and "Association".

Clustering

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

FIG : 4.5.3.6

FIG : 4.5.3.7

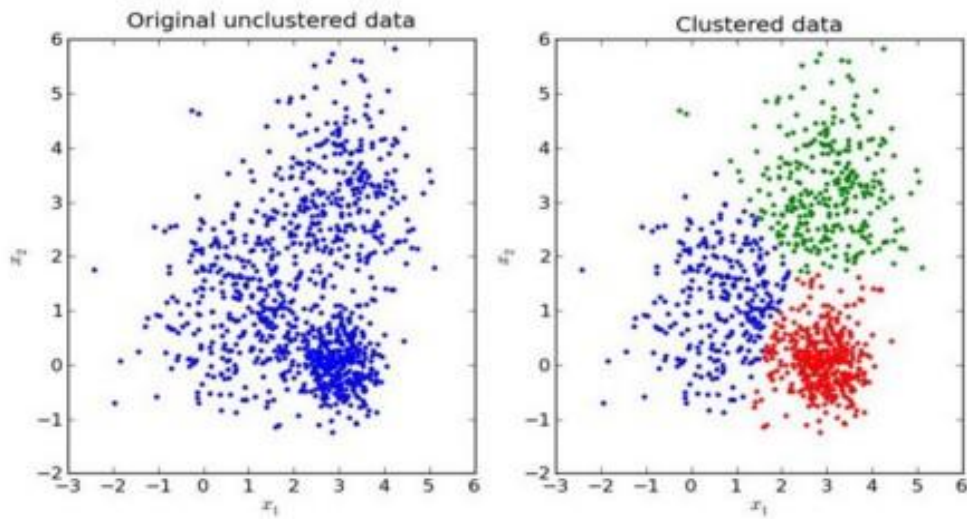


Fig 4.6 Clustered and Unclustered data

Methods used for clustering are:

- Gaussian mixtures
- K-Means Clustering
- Boosting
- Hierarchical Clustering
- K-Means Clustering
- Spectral Clustering

Overview of models under categories

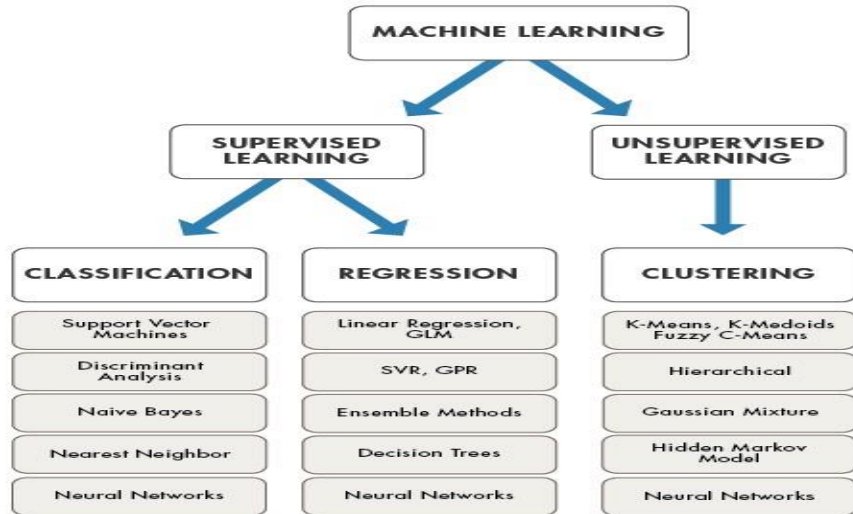


Fig 4.7 Model categories

4.4.4 Training and testing the model on data

For training a model we initially split the model into 3 three sections which are ‘Training data’, ‘Validation data’ and ‘Testing data’. train the classifier using ‘training data set’, tune the parameters using ‘validation set’ and then test the performance of your classifier on unseen ‘test data set’. An important point to note is that during training the classifier only the training and/or validation set is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

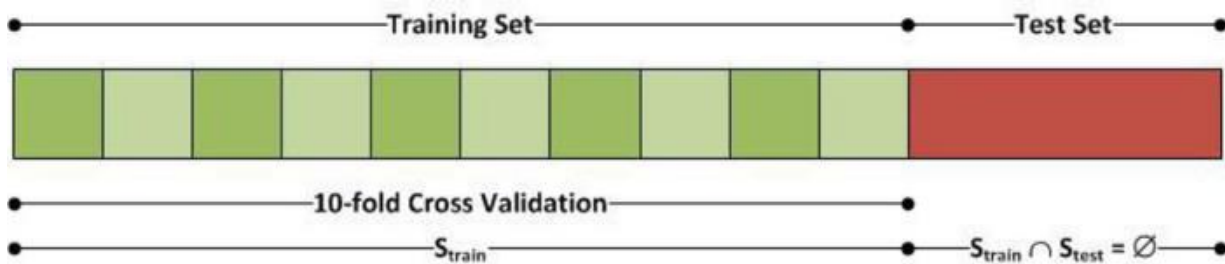


Fig 4.8 Testing Classifier

Training set

The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

Validation set

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. A set of unseen data is used from the training data to tune the parameters of a classifier.

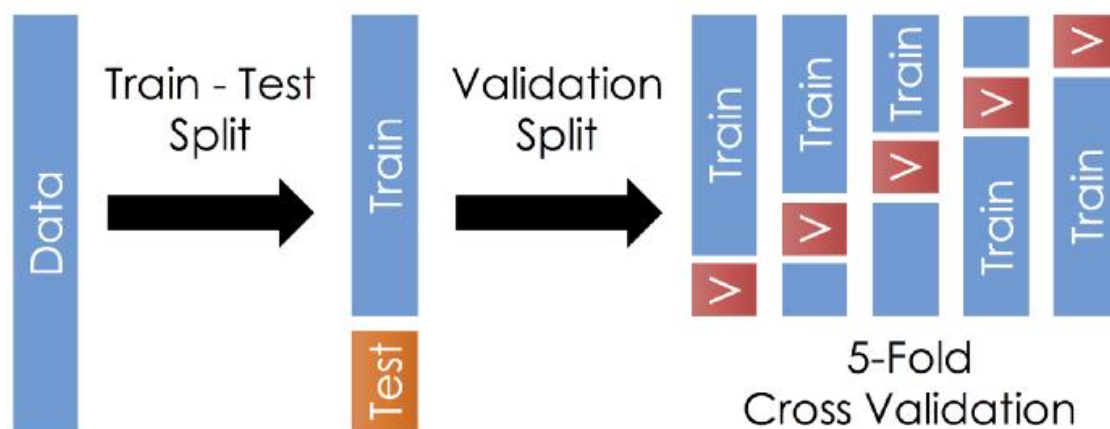


Fig 4.9 Classifier Parameters

Once the data is divided into the 3 given segments we can start the training process.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Data points in the training set are excluded from the test (validation) set. Usually, a data set is divided into a training set, a validation set (some people use 'test set' instead) in each iteration, or divided into a training set, a validation set and a test set in each iteration.

The model uses any one of the models that we had chosen in step 3/ point 3. Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data. Once this is done we can develop a confusion matrix, this tells us how well our model is trained. A confusion matrix has 4 parameters, which are '**True positives**', '**True Negatives**', '**False Positives**' and '**False Negative**'. We prefer that we get more values in the True negatives and true positives to get a more accurate model. The size of the Confusion matrix completely depends upon the number of classes.

Table 4.1 Prediction table

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

True positives : These are cases in which we predicted TRUE and our predicted output is correct.

True negatives : We predicted FALSE and our predicted output is correct.

False positives : We predicted TRUE, but the actual predicted output is FALSE.

False negatives : We predicted FALSE, but the actual predicted output is TRUE.

We can also find out the accuracy of the model using the confusion matrix.

Accuracy = (True Positives + True Negatives) / (Total number of classes)

i.e. for the above example:

Accuracy = $(100 + 50) / 165 = 0.9090$ (90.9% accuracy)

4.4.5 Evaluation

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we might tune the hyper-parameters of the model and try to improve the accuracy and also looking at the confusion matrix to try to increase the number of true positives and true negatives.

4.5 ALGORITHM USED

4.5.1 Convolutional Neural Network:

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as $h * w * d$, where h = height w = width and d = dimension. For example, An RGB image is $6 * 6 * 3$ array of the matrix, and the grayscale image is $4 * 4 * 1$ array of the matrix.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

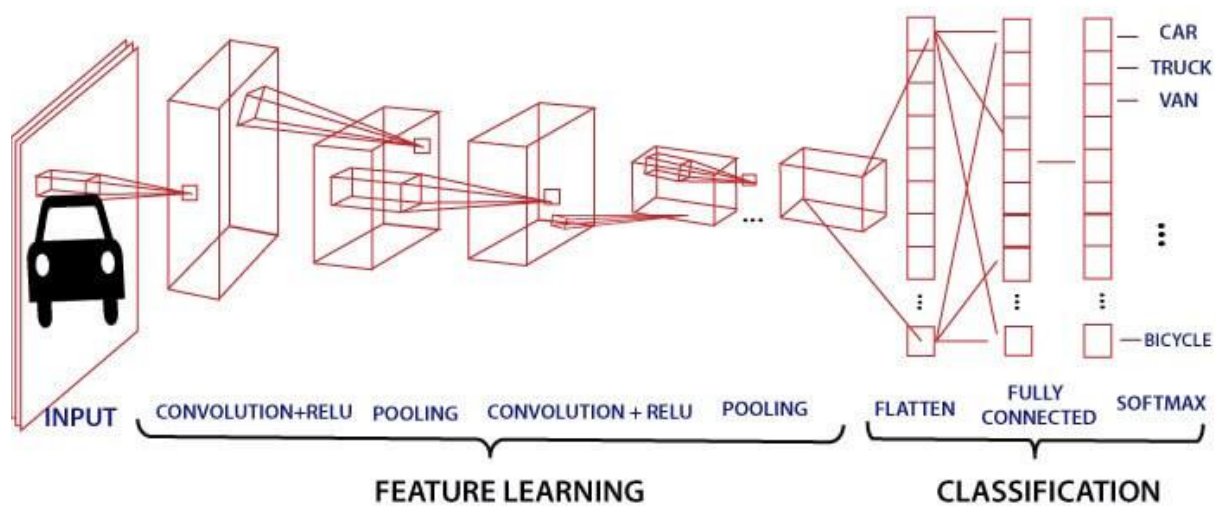


Fig 4.10 Classification using CNN

4.5.2 Convolution Layer

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

4.5.3 Strides

Stride is the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the convolution would work with a stride of 2.

4.5.4 Padding

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end.

4.5.5 Pooling Layer

Pooling layer plays an important role in pre-processing of an image. Pooling layer

reduces the number of parameters when the images are too large. Pooling is "downscaling" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called downsampling or subsampling, which reduces the dimensionality of each map but retains the important information. There are the following types of spatial pooling:

4.5.6 Max Pooling

Max pooling is a sample-based discretization process. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned.

Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

4.5.7 Average Pooling

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

4.5.8 Fully Connected Layer

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

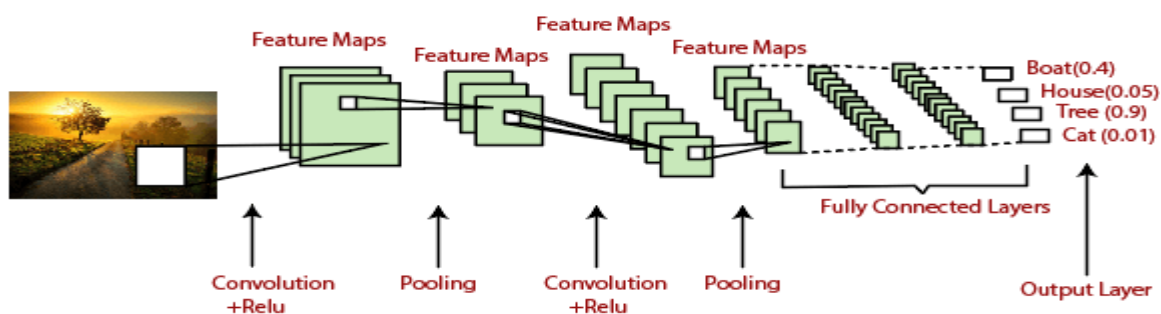


Fig 4.11 Convolution Neural Network

CHAPTER 5

RESULT AND DISCUSSION

Our proposed model is designed and development to detect and classify pneumonia from chest X-ray images. It contains both image processing and convolutional neural network. We developed a model The algorithm begins by transforming chest X-ray images into sizes smaller than the original. The next step involves the identification and classification of images by the convectional neural network framework, which extracts features from the images and classify them. This work has presented the X-Ray images for Pneumonia detection based on convolutional neural networks and different machine learning. By training a set of strong CNNs on a large scale dataset, we built a model that can accurately predict Pneumonia. During each epoch data is trained over and over again to learn the feature of data. The performance evaluation of the model is estimated by using classification accuracy and cross-validation. We performed 5-fold and 10- fold cross-validation and presented the results in terms of mean and standard deviation(SD). Performances of seven Machine learning classification models are presented in The results presented in Figure 9, show that accuracy of Pneumonia Image Classification. It can be seen that all machine-learning models, except the decision tree. The highest accuracy was 98.46% to deep learning . On the other hand, the decision tree classifier achieved the lowest performance of 85.64%. An F-test carried out over these results confirms that the performance differences between predictive models are statistically significant. As shown in Figure 10, F-score of Pneumonia Image Classification. The average F- score of the classifier is all-more than 90%, which is all an excellent result. Therefore, in these results of the f-score, the classification accuracy of Deep Learning (DNN) with Feed Forward and Random Forest (RF) is approximately 98.95%, which is large as compared to other classifiers.

APPENDICES

A SOURCE CODE:

Pneumonia Detection:

```
import os
import numpy as np
import cv2
from numpy import random
from sklearn.metrics import accuracy_score, confusion_matrix
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
from tqdm import tqdm
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from tensorflow.keras.models import Model
from matplotlib import pyplot
import matplotlib.pyplot as plt

input_path = './chest_xray/'
#defined some constants for later usage
img_dims = 160
epochs = 15
batch_size = 32
class_names = ['/NORMAL/', '/PNEUMONIA/']

#Fitting the CNN to the images
def process_data(img_dims, batch_size):
    # Data generation objects
    train_datagen = ImageDataGenerator(rescale=1. / 255, zoom_range=0.3,
shear_range=0.2, vertical_flip=True)
```

```

val_datagen = ImageDataGenerator(rescale=1. / 255)

train_gen = train_datagen.flow_from_directory(
    directory=input_path + 'train',
    target_size=(img_dims, img_dims),
    batch_size=batch_size,
    class_mode='binary',
    shuffle=True)

val_gen = val_datagen.flow_from_directory(
    directory=input_path + 'val',
    target_size=(img_dims, img_dims),
    batch_size=batch_size,
    class_mode='binary',
    shuffle=True)

# I will be making predictions off of the test set in one batch size
# This is useful to be able to get the confusion matrix
test_data = []
test_labels = []
label = 0
for cond in ['/NORMAL/', '/PNEUMONIA/']:
    for img in tqdm(os.listdir(input_path + 'test' + cond)):
        img = plt.imread(input_path + 'test' + cond + img)
        img = cv2.resize(img, (img_dims, img_dims))
        if len(img.shape) == 1:
            img = np.dstack([img, img, img])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = img.astype(np.float32) / 255
        if cond == '/NORMAL/':
            label = 0

```

```

        elif cond == '/PNEUMONIA/':
            label = 1
            test_data.append(img)
            test_labels.append(label)

test_data = np.array(test_data)
test_labels = np.array(test_labels)

return train_gen, val_gen, test_data, test_labels

train_gen, val_gen, test_data, test_labels = process_data(img_dims, batch_size)

#Initialising the CNN
model = models.Sequential()

model.add(layers.Conv2D(16,(3,3),activation = 'relu',
padding='same',input_shape=(img_dims,img_dims,3)))
model.add(layers.MaxPooling2D((2,2)))

model.add(layers.Conv2D(32,(3,3),activation = 'relu', padding='same'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D((2,2)))

model.add(layers.Conv2D(64,(3,3),activation = 'relu', padding='same'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D((2,2)))

model.add(layers.Conv2D(128,(3,3),activation = 'relu', padding='same'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D((2,2)))

```

```

model.add(layers.Conv2D(256,(3,3),activation = 'relu', padding='same'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D((2,2)))
# model.add(layers.Dropout(0.3))

model.add(layers.Flatten())
# Creating 1 Dense Layer
model.add(layers.Dense(units=128,activation='relu'))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(units=1, activation='sigmoid'))

#Compiling the CNN
model.compile(optimizer = 'adam',loss='binary_crossentropy',metrics=['accuracy'])

#model.load_weights('best_weights.hdf5')

checkpoint = ModelCheckpoint(filepath='best_weights.hdf5', save_best_only=True)
lr_reduce   =   ReduceLROnPlateau(monitor='val_loss',   factor=0.2,   patience=2,
verbose=2, mode='max')
#   early_stop   =   EarlyStopping(monitor='val_loss',   min_delta=0.1,   patience=1,
mode='min')

hist       =       model.fit(train_gen,       epochs=epochs,       validation_data=val_gen,
callbacks=[checkpoint, lr_reduce])

test_loss, test_acc = model.evaluate(test_data, test_labels)

##Predictions
predictions = model.predict(test_data)

```

```

predictions = np.array(predictions).reshape(-1)

##Accuracy
print("Untrained model, accuracy: {:.2f}%".format(100 * test_acc))

acc = accuracy_score(test_labels, np.round(predictions))*100
cm = confusion_matrix(test_labels, np.round(predictions))
tn, fp, fn, tp = cm.ravel()

print('CONFUSION MATRIX -----')
print(cm)

print('\nTEST METRICS -----')
precision = tp/(tp+fp)*100
recall = tp/(tp+fn)*100
print('Accuracy: {}'.format(acc))
print('Precision: {}'.format(precision))
print('Recall: {}'.format(recall))
print('F1-score: {}'.format(2*precision*recall/(precision+recall)))

#Visualize the accuracy plots and the model loss
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

for i, met in enumerate(['accuracy', 'loss']):
    ax[i].plot(hist.history[met])
    ax[i].plot(hist.history['val_' + met])
    ax[i].set_title('Model {}'.format(met))
    ax[i].set_xlabel('epochs')
    ax[i].set_ylabel(met)
    ax[i].legend(['train', 'val'])

```

```
plt.show()
```

```
##Visualize the predicted x-ray images from the test
```

```
def plot_image(i, predictions_array, true_label, img):
```

```
    true_label, img = true_label[i], img[i]
```

```
    plt.grid(False)
```

```
    plt.xticks([])
```

```
    plt.yticks([])
```

```
    plt.imshow(img, cmap=plt.cm.binary)
```

```
    predicted_label = (0 if (predictions_array[i] < 0.5) else 1)
```

```
    if int(predicted_label) == int(true_label):
```

```
        color = 'blue'
```

```
    else:
```

```
        color = 'red'
```

```
    plt.xlabel("{} {:.3f} {}".format(class_names[predicted_label],
```

```
                                     predictions_array[i],
```

```
                                     true_label), color=color)
```

```
def plot_value_array(i, predictions_array, true_label):
```

```
    true_label = true_label[i]
```

```
    plt.grid(False)
```

```
    plt.xticks(range(2))
```

```
    plt.yticks([])
```

```
    thisplot = plt.bar(range(2), [1 - predictions_array[i], predictions_array[i]],  
color="#777777")
```

```
    plt.ylim([0, 1])
```

```
    predicted_label = (0 if (predictions_array[i] < 0.5) else 1)
```

```

if predicted_label == int(true_label):
    thisplot[int(true_label)].set_color('blue')
else:
    thisplot[predicted_label].set_color('red')

def data_test_opt(pred):
    pred[pred > 0.5] = int(1)
    pred[pred <= 0.5] = int(0)

    return np.array(pred).reshape(-1)

##Shuffling the x-ray images
def shuffle_tow_array(a, b):
    indices = np.random.permutation(len(a))
    A = a[indices]
    B = b[indices]
    return A, B

test_data, test_labels = shuffle_tow_array(test_data, test_labels)

#####

#####

num_rows = 5
num_cols = 3
num_images = num_rows * num_cols
plt.figure(figsize=(2 * 2 * num_cols, 2 * num_rows))
ind = random.randint(len(predictions), size=(num_images))

```

```

for i in range(num_images):
    plt.subplot(num_rows, 2 * num_cols, 2 * i + 1)
    plot_image(ind[i], predictions, test_labels, test_data)
    plt.subplot(num_rows, 2 * num_cols, 2 * i + 2)
    plot_value_array(ind[i], predictions, test_labels)
plt.tight_layout()
plt.show()

```

##print the names of the layers from Feature map and from the filters

```

def showFiltersBlocks():
    indexFilter = []
    for index in range(len(model.layers)):
        layer = model.layers[index]
        # check for convolutional layer
        if 'conv' not in layer.name:
            continue
        # get filter weights
        filters, biases = layer.get_weights()
        indexFilter.append(index)
        print(layer.name, filters.shape)
    print(indexFilter)
    return indexFilter

```

```

print("_____ Visualize           Feature           Map
Start_____")

```

##Visualize Feature Map

```

def showFeatureMap(indexs):
    outputs = [model.layers[i].output for i in indexs]
    model2 = Model(inputs=model.inputs, outputs=outputs)

```

```

# get feature map for first hidden layer
feature_maps = model2.predict(test_data[1:6])
# plot the output from each block
# plot all 64 maps in an 8x8 squares
plt.figure(figsize=(15, 15))
j = 0
for fmap in feature_maps:
    square = 4
    ix = 1
    fig = plt.figure(figsize=(5, 5))
    filters, biases = model.layers[indexs[j]].get_weights()
    print(model.layers[indexs[j]].name)
    text = fig.text(0.5, 1., str(model.layers[indexs[j]].name) + " " + str(filters.shape),
                    ha='center', va='center', size=20)
    text.set_path_effects([path_effects.Normal()])
    for _ in range(square):
        for _ in range(square):
            # specify subplot and turn of axis
            ax = plt.subplot(square, square, ix)
            ax.set_xticks([])
            ax.set_yticks([])
            # plot filter channel in grayscale
            plt.imshow(fmap[0, :, :, ix - 1])
            ix += 1
        j += 1
# show the figure
plt.show()

```

```

showFeatureMap(showFiltersBlocks())

```

```
print("_____Visualize Filters Start_____")
```

```
##Visualize Filters
```

```
def showFilters(indexs):
```

```
    for index in indexs:
```

```
        fig = plt.figure(figsize=(5, 5))
```

```
        filters, biases = model.layers[index].get_weights()
```

```
        text = fig.text(0.5, 1., str(model.layers[index].name) + " " + str(filters.shape),
```

```
                        ha='center', va='center', size=20)
```

```
        text.set_path_effects([path_effects.Normal()])
```

```
        # normalize filter values to 0-1 so we can visualize them
```

```
        f_min, f_max = filters.min(), filters.max()
```

```
        filters = (filters - f_min) / (f_max - f_min)
```

```
        # plot first few filters
```

```
        n_filters, ix = 6, 1
```

```
        for i in range(n_filters):
```

```
            # get the filter
```

```
            f = filters[:, :, :, i]
```

```
            # plot each channel separately
```

```
            for j in range(3):
```

```
                # specify subplot and turn of axis
```

```
                ax = pyplot.subplot(n_filters, 3, ix)
```

```
                ax.set_xticks([])
```

```
                ax.set_yticks([])
```

```
                # plot filter channel in grayscale
```

```
                pyplot.imshow(f[:, :, j])
```

```
                ix += 1
```

```
        # show the figure
```

```
        pyplot.show()
```

```
showFilters(showFiltersBlocks())
```

Test App:

```
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
import cv2 as cv
import numpy as np
import os
```

```
app = Flask(__name__)
```

```
model = load_model('model.h5')
Dict = {0: 'Normal', 1: "Pneumonia"}
```

```
def predict_label(imgPath):
    imgSize = (224, 224)
    img = cv.imread(imgPath)
    img = cv.resize(img, imgSize)
    img = img.astype(np.float32)/255.
    finalImg = np.expand_dims(img, axis=0)
    result = np.argmax(model.predict(finalImg), axis=-1)[0]
    re = Dict[result]

    if re == "Pneumonia":
        return "Pneumonia Positive"
    return re
```

```
@app.route('/', methods=['GET'])
```

```
def index():  
    return render_template('index.html')  
  
@app.route('/submit', methods=['POST'])  
def get_output():  
    if request.method == 'POST':  
        img = request.files['my_image']  
        img_path = "static/" + img.filename  
        img.save(img_path)  
        p = predict_label(img_path)  
        os.remove(img_path)  
        return render_template("index.html", prediction=p)  
  
# main driver function  
if __name__ == '__main__':  
    app.run()
```

Screen Shots:

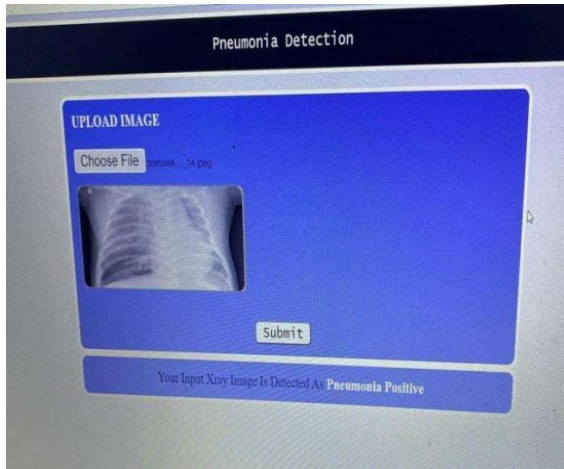


Fig B.1 Pneumonia positive

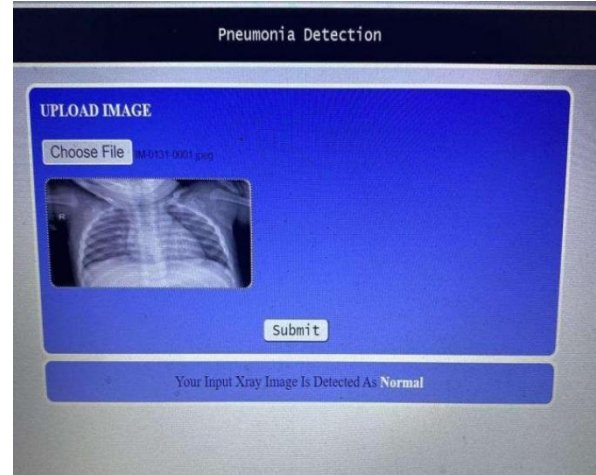


Fig B.2 Normal

SUMMARY AND CONCLUSIONS

In this project, we offered a model for detecting and classifying pneumonia from chest X-ray images using machine learning methods based on Convolutional Neural Networks (CNN). In particular, comparing all seven models based on test accuracy score, F-score and ROC curve, it appears that CNN outperforms all other models by a small margin with test accuracy score 98.46%. Random forest performs surprisingly well with test accuracy score 97.61%. It should note that the CNN model was run with only 100 epochs due to its long running and waiting time. We believe, with proper parameter tuning and giving a little more time, the CNN model could achieve an even better result. The result of our model indicates that our model outperforms other methods, which use no extra training data on chest radiography and shows that image training may be sufficient for general medical image recognition tasks. Moreover, the cooperation between a machine learning based medical system and the detection of Pneumonia will improve the outcomes and bring benefits to clinicians and their patients. This study may use in similar research which will help to develop a diagnostic tool such as COVID-19 diagnostic. future, we will collect X-ray images from hospitals with big data to train and test the system to predict better results. Also, we have plan to work with more complex medical data.

REFERENCES:

- 1 M. Roser, H. Ritchie, and B. Dadonaite, "Child & infant mortality," Our World in Data, 2013.
2. C. Turkington and B. Ashby, The encyclopedia of infectious diseases. Infobase Publishing, 2007.
- [3]R. T. Sousa, O. Marques, F. A. A. Soares,. Sene Jr, L. L. de Oliveira, and E. S. Spoto, "Comparative performance analysis of machine learning classifiers in detection of childhood pneumonia using chest radiographs," Procedia Computer Science, vol. 18, pp. 2579–2582, 2013.
- [4] Step to healthy, "Symptoms of pneumonia and remedies," 2018. [Online]. Available: <https://steptohealth.com/symptoms-of-pneumoniacomplementary-natural-remedies/>
- [5]W. H. Organization et al., "Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children," World Health Organization, Tech. Rep., 2001.
6. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
7. Milletari, F.; Navab, N.; Ahmadi, S.A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proceedings of the 2016 Fourth International Conference on 3DVision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
8. Jeelani, H.; Martin, J.; Vasquez, F.; Salerno, M.; Weller, D. Image quality affects deep learning reconstruction of MRI. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 357–360.
9. Chlemper, J.S.; Caballero, J.; Hajnal, J.; Price, A.N.; Rueckert, D. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans. Med. Imaging 2017, 37, 491–503. [CrossRef] [PubMed]
10. Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning.arXiv 2017, arXiv:1711.05225.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.

11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deepresidual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp.770–778.
12. Yu, F.; Koltun, V.; Funkhouser, T. Dilated Residual Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 25–30 June 2017; pp. 636–644.
13. Liang, G.; Zheng, L. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Comput. Methods Programs Biomed.* 2020, 187, 104964. [CrossRef] [PubMed]
14. Jain, R.; Nagrath, P.; Kataria, G.; Kaushik, V.S.; Hemanth, D.J. Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurment* 2020, 165. [CrossRef]



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in



National Conference on Computational Intelligence and Communication Networks



NCCICN

24TH – 25TH MARCH 2022



Certificate of Presentation

*This is to certify that Dr./Mr./Ms. CHANDAVARAPU SAI SARAN, of Sathyabama Institute of Science and Technology, has presented a paper entitled **Diagnosing Pneumonia by use of Deep Learning Processing Model based on CNN**, in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala

Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan

Convener
Professor & Head, CSE

Dr. S. Vigneshwari

Convener
Professor & Head, CSE

