SMART MEDICAL DIAGNOSIS SYSTEM – PNEUMONIA DETECTION

An Engineering Project in Community Service

A PROJECT REPORT FOR PHASE – II

Submitted by

ARYAN OMKAR ASHAR-19BAI10094
SHREYA SHETYE-19BAI10028
PRANAV SHARMA: - 19BAI10154
AISWARYA NAIR – 19BOE10019
SHIREEN SOPOREE – 19BOE10049
VAIDEHI CHAUDHARI – 19BOE10051
KHUSHII GUPTAA – 19BCE10339
KUSHAGRA – 19BCY10074

in partial fulfillment of the requirements for the degree of

Bachelor of Technology

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

SCHOOL OF BIOENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE

MADHYA PRADESH - 466114

APRIL 2022.

VIT BHOPAL UNIVERSITY,KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **“SMART MEDICAL DIAGNOSIS SYSTEM”** is the bonafide work of “**Aryan Omkar Ashar(19BAI10094), Shreya Shetye (19BAI10028), Pranav Sharma(19BAI10154), Aiswarya Nair(19BOE10019), Shireen Soporee (19BOE10049), Vaidehi Chaudhari(19BOE10051), Khushii Guptaa(19BCE10339) and Kushagra(19BCY10074)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROJECT GUIDE
Dr. Kanchan Lata Kashyap
School of Computer Science and Engineering
VIT Bhopal University

Comments & Signature (Reviewer 1)

Comments & Signature (Reviewer 2)

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work. We would like to thank VIT Bhopal University for giving us this opportunity to work on this project.

We wish to express our heartfelt gratitude to Dr. S. Poonkuntran, Dean of School of Computing Science and Engineering and Dr. Harihara Padhy, Head of Department, School of Bioengineering for much of their valuable support and encouragement in carrying out this work.

We would also like to thank our internal guide Dr. Kanchan Lata Kashyap, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work. Along we that, we would deeply like to thank our reviewers Dr. Pallabi Sarkar as well as Dr. Atul Aman who always pushed us to include the best functionalities in this project.

We would like to thank all the technical and teaching staff of the School of Computing Science and Engineering and School of Bioengineering, who extended directly or indirectly all support.

Last, but not the least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

AI – Artificial Intelligence
ML- Machine Learning
API- Application Programming Interface
OpenCV- Open-Source Computer Vision Library
SQL – Structured Query Language
GPL: - General Public License
MTV – Model Template Views
BCI – Brain Computer Interface

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the new era of machines and artificial intelligence, no field is lacking behind. Medical diagnosis has also found its way in including the techniques of machine learning and making intelligent decisions and is thus helping the community by accurate decisions. Disease diagnosis is one of the major fields of the medical industry. Disease diagnosis is the identification of a health issue, disease, disorder, or other condition that a person may have. Pneumonia is a lung disease that is causing deaths all around the world from several decades now. Also, in the pandemic era, pneumonia has taken lives of millions of people as pneumonia was an after-effect of COVID-19. This project mainly focuses on a smart medical diagnosis system that will be able to identify and detect whether the person is suffering from pneumonia or not. The patient just has to enter his/her chest x-ray scans and within seconds the patient will be able to know whether he/she is suffering from pneumonia. The system encompasses advanced python programming, IBM Watson Studio, Spyder, HTML, CSS and JavaScript. AI is bringing a huge transformation the field of medicine. It's helping doctors diagnose patients more accurately, make predictions about patients' future health, and recommend better treatments. There are two models in the project. One can be used without internet connectivity and another can be used with internet connectivity. The testing accuracy of both these models in 99.99974%

Our project will be of a great help to the community in many ways. Currently, medical diagnosis is a tedious task and costly as well. People living in rural areas thus do not get themselves diagnosed and then face a tough time with this disease. Our project will not cost anything and the patient can perform a self-diagnosis and can get to know how severe pneumonia the person is suffering from. In this way, the patient can decide whether he needs to go to the hospital or not and thus this can prevent unnecessary visits to the hospital. The model will mainly use supervised learning algorithm and convolution neural networks and many other techniques in order to predict the disease. We aim to create a creative project which will help the society by detecting pneumonia in its early stages and thus saving lives of many people. We further discuss various methodologies along with corresponding techniques of AI, including Fuzzy Logic, Machine Learning, and Deep Learning. We will also compare our model with the existing models and state the advantages and disadvantages of both these models. Our project is proven to be feasible to use, economical, independent of demography and a ready to use project due to the accuracy obtained in the project. Thus, in order to promote development in the society and help the community, we propose this model in order to detect pneumonia and save lives of many people by giving them another chance to live

TABLE OF CONTENTS

1. INTRODUCTION

1.1: Introduction

Medical diagnosis is one of the major aspects that is being researched day by day to make it better and more efficient daily. Medical diagnosis can be defined as the process of determining which disease or condition explains a person's symptoms and signs. Pneumonia is a type of respiratory infection that primarily affects the lungs. In these acute respiratory diseases, human lungs are made up of small sacs called alveoli that are filled with air in normal and healthy people, but in pneumonia these alveoli become filled with fluid or "pus." Obtaining a chest X-ray is a major step in the detection and treatment of the phenomenon (CXR).

Pneumonia has remained a threat to human lives for a long time. The pneumonia cases of those affected by the deadly COVID19 commonly known as Coronavirus have been increasing at an alarming rate. The increase in imaging orders, combined with the availability of high-resolution scanners, has resulted in heavy workloads for radiologists. They contribute to overcalls or missed findings by radiologists when combined with lower resolution imaging, cognitive biases, and system factors. With today's technological advances, it is now possible to detect pneumonia based on chest x-rays using tools based on deep learning frameworks. The challenge here is to support the diagnostic process, which enables accelerated treatment and better clinical outcomes. We have given the test and training images labels, which makes it easier to classify if the user has Pneumonia or not. The main aim of the project is to facilitate pneumonia detection for all people at a faster rate.

ML algorithms allows us to build models that associate many types of variables with a disease. Machine learning and artificial intelligence technologies that are embedded into the diagnostic system helps us increase the efficiency of the results, helps us calculate the risk factors and also reduce the unnecessary visits to hospitals and doctors. Machine learning promises and aims to revolutionize decision making in clinic and diagnosis decisions.

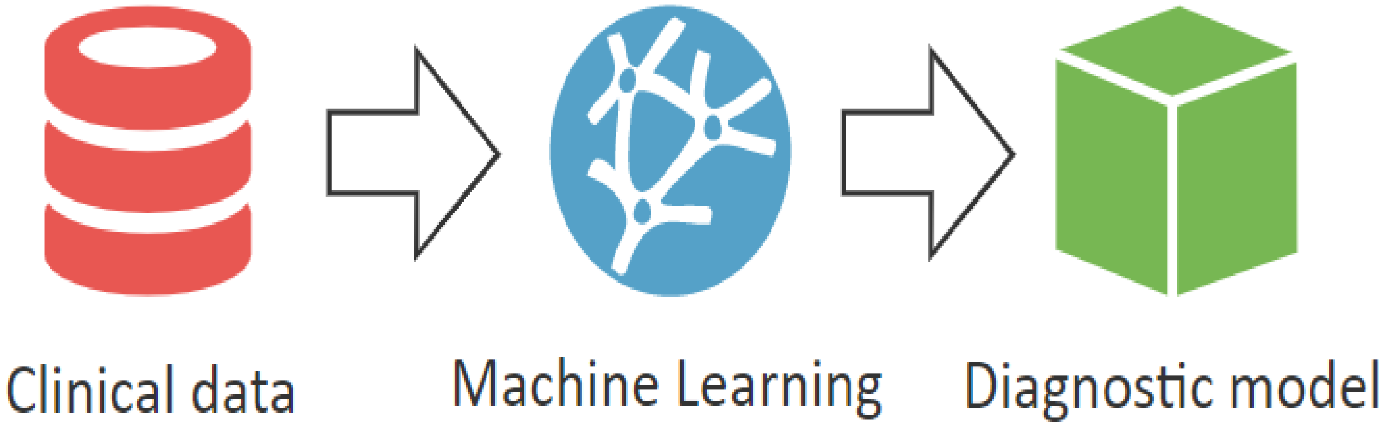


Fig. 1.1.1: - Basic Diagnostic Model

The project uses various python libraries in order to train the model. First, we download the dataset from the help of Kaggle and some of the images are real-time scans taken directly from the hospital database as well as from one patient. On this data, first we analyse the data and then we perform exploratory data analysis and make the data suitable for training. We divide the total images into 3 different forders for training, testing and validation dataset. Next, we train the model by use of certain algorithms by the help of advanced CNN algorithm and we train it on jupyter notebook. Next, we transfer the downloaded model on spyder and we make the frontend of the model to be used as a website. By the help of HTML and CSS we develop the frontend and train the downloaded model and use it on spyder. After all this, we transfer the model and train it on IBM Watson Studio. We perform IBM deployment and after training the model on IBM, we download the IBM trained model and we deploy the new model as well on Flask. These steps have been explained in detail in the further part of the report.

1.2: Motivation for the work

Every day we hear many people who die or reach a last stage of their health implication/disease due to late diagnosis. Smart medical diagnostic system aims to perform early diagnosis of the disease by calculating the risk factors and then the patient can decide how and when he/she should visit the hospital. Normally in today's world, it takes 1-2 day for a report to come. A patient suffering from Pneumonia takes an X-ray image to the doctor; with them he predicts pneumonia. The results are not just based on seeing the X-ray images, furthermore, tests will be conducted on the patient. The process was time-consuming, but in recent days artificial intelligence helps in predicting pneumonia bypassing the X-ray image. The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to precisely predict pneumonia. Pneumonia affects a large number of individuals, especially children, in developing countries where risk factors include poor sanitary conditions, malnutrition, and a lack of appropriate medical services. By the end of this project, we aim to develop a smart medical diagnostic system that can give quick and accurate results and thus can perform more diagnosis than the existing techniques thus helping the patients. The quick detection of pneumonia can help to save the life of the patient. This can help the doctor to decide the type of disease the patient needs. Also, the diagnosis of pneumonia is very costly and thus the poor people cannot afford it. For them this test is almost free thus making our system cost effective. People with disabilities and old people can not visit the doctor very frequently and thus this system helps them detect if they are at risk or not and thus, they can take a decision whether they should see a doctor or not.

Around three lakh children in India lose their lives due to pneumonia and it is a sad fact which states that half who die in one year are due to pneumonia in India. Most of the time the disease is being ignored. We can protect many people's lives if pneumonia is detected on time and quick action is taken on it. For a successful recovery, pneumonia must be detected early. X-ray scan examination is the most common way of diagnosis, albeit it is based on the radiologist's interpretation ability. An autonomous system with generalising capabilities is required to detect the condition. Some studies have shown that discrepancies in the interpretation of chest x-rays by radiologists are common. Chest X-rays are currently the most common imaging modality read by radiologists in hospitals and teleradiology practices. These lower resolution modalities, despite their widespread use, are not the easiest to interpret. To read them correctly, you must first understand the viewing constraints caused by patient positions, image quality, and tissue overlays. This disadvantage of human-based observation motivated to intervene in this area where the machine classifies normal and abnormal chest x-rays.

This is complicated by the fact that there's been no major initiative in the radiology community to catalogue the wide range of possible findings seen among chest radiographs. Second, because many of the methods were trained on single hospital source datasets, their generalizability to different demographics around the world is dubious. Third, the evaluation methodology for these algorithms employs well-known AI metrics such as area under the curve (AUC) or F-scores for label-based precision and recall evaluations.

1.3: Problem Statement

- The pre-existing diagnostic systems take a lot of time and are many times incorrect. Also, the accuracy with which pneumonia is detected is very less.
- People in rural areas do not have access to proper medical facility and thus these people do not get themselves diagnosed on time for pneumonia.
- Many times, we need to analyse the data of the patient and check whether the health of the patient is improving or not. This is tough without the help of graphs. Our system will aim to provide data visualisation with the help of graphs.
- Till now, we always needed an expert doctor in order to perform the diagnosis on the body. But our smart medical diagnostic system will give the answers on the basis of a picture entered into the system.
- Pneumonia was one of the major side - effect that we observed in COVID-19 patients and this system helps to detect pneumonia easily and quickly without the help of a doctor. Also, a doctor may make a mistake in analysing the X-RAY but our project will not make a mistake in the same.
- Local existing models cannot perform accurately.

1.4: Objective of the work

- The first major objective of out project is to perform diagnosis on the human body and predict if it is infected with any pneumonia or not. The doctors can start the medication in that way.
- We aim to perform early diagnosis of pneumonia in the patient so that the recovery process can begin as soon as possible.
- Rural people and people living in areas where medical facility is less can use this system and perform self-diagnosis and can get themselves treated for pneumonia.
- Medical diagnosis can be performed quickly and efficiently and with high amount of accuracy.
- The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to precisely predict pneumonia. Pneumonia affects a large number of individuals, especially children, in developing countries where risk factors include poor sanitary conditions, malnutrition, and a lack of appropriate medical services.
- This project helps in development of 2 different models. The first model can be run without internet on the local system. Another model is trained by the help of IBM Watson Studio and this model requires an internet connectivity. The accuracy of the model trained by the help of IBM Watson Studio is greater than the locally deployed model.

1.5: About Introduction to the project including techniques

Many times, we have heard that the doctors cannot identify the disease in the early stages. This happens in the cases of pneumonia where it is very difficult to treat when it reaches the last stage. Treating the disease in its early stages is easier when compared to treating it in the later stages. Poor people do not go to the doctor thinking that they may have minor cough/cold and then later they come to know that they are a prey of a more dangerous disease which can be pneumonia and thus this affects their health more and affects their chances of living. For all this we aim to make a smart diagnostic system that can be done without a doctor and can detect pneumonia in the patient. We have made the use of Python, JavaScript, CSS, HTML and Spyder application. Along with these applications, we have taken help of IBM Watson Studio in order to increase our accuracy and take it to the higher level so that this model can be relied upon. The testing accuracy on the IBM Watson Studio model is recorded to be 99.99974%. The locally trained model is observed to have an accuracy of 93.52%

The techniques used in this project are: -

1. The first page that is shown up to the user is the home page that contains the details regarding the project and what all the project can perform. The user is given a choice in order to proceed ahead by clicking the button
2. The model asks the user to click on a button. Through this, the user will be able to select an image and put it in on the webpage. This image will be an X-RAY scan of the lungs.
3. Based on the responses of the person, we will perform feature selection on the data. We also first clean our dataset as medical data is always fragmented.
4. We will use python libraries along with JavaScript, CSS, HTML and IBM Watson Studio in order to accurately predict the disease. The webpage consists of static as well as dynamic content.
5. After logging in we will be able perform data analysis and predict the disease.
6. We have used feature extraction and feature engineering in order to predict the disease based on the symptoms. This part is explained in depth when we see the proposed system and the overall architecture diagram of the system.
7. IBM Watson Studio has been used in order to develop this project. This model requires internet connectivity and can be used in city hospitals. But in rural areas, we can use our local model which can work without the help of internet.

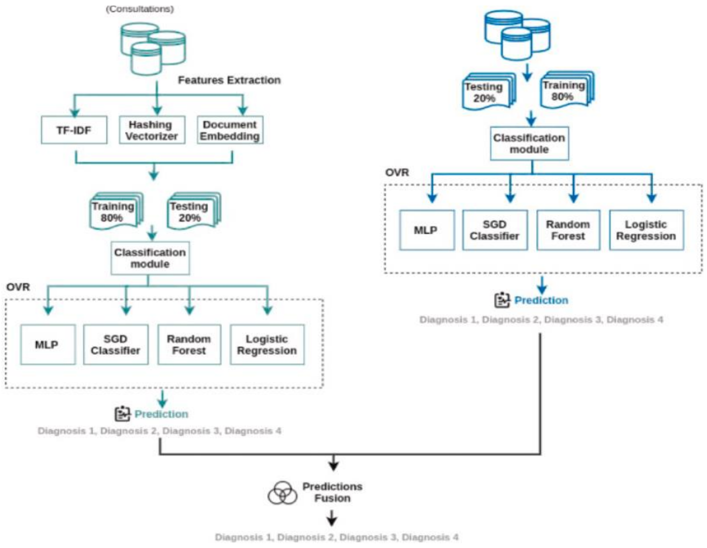


Fig. 1.5.1: -   Techniques used in the project

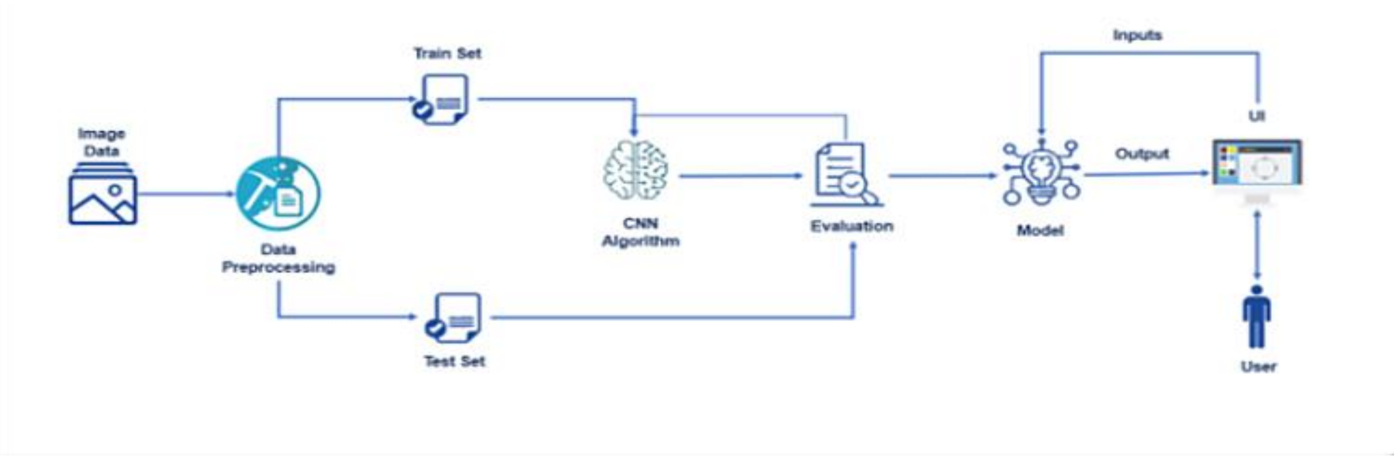


Fig. 1.5.2: -   Technical Architecture Diagram

1.6 Artificial Intelligence Aiding Medical Diagnosis

A Virtual Diagnosis does not constitute or replace medical diagnosis or medical advice, but helps the User in making a feasible decision regarding their health condition. In artificial intelligence, diagnosis of pneumonia is concerned with the development of algorithms and techniques that are able to determine whether the behavior of a human body is correct. If the human is not functioning correctly, the algorithm should be able to determine, the person having pneumonia, which part of the body is failing, and which kind of fault it is facing. The computation is based on observations, which provide information on the current behaviour of the body

Healthcare artificial intelligence is a general term used to describe the use of machine learning algorithms and software, or artificial intelligence (AI), to mimic human perception in analyzing, presenting, and understanding complex medical and healthcare. AI performs this by the help of ML and deep learning algorithms. These algorithms can recognize patterns in behaviour and logic generation.

The primary goal of health-related AI applications is to analyze the relationship between prevention or treatment techniques and patient outcomes. AI programs have been applied to the diagnosis and treatment of pneumonia, the development of a drug for the disease, as well as the monitoring and care of patients. AI algorithms can also be used to analyze large amounts of data through electronic health records for disease prevention and diagnosis. Artificial intelligence has led to significant improvements in medical fields such as medical imaging, automated clinical decision-making, diagnosis, and prognosis that can be used as X-rays to detect inflammation. AI is a field rich in data, algorithms, analytics, deep learning, neural networks, and ideas that are constantly evolving and adapting to the needs of the healthcare industry and of patients. Artificial intelligence in medical diagnostics and healthcare provides overworked doctors and facilities with reliable support, helping to less the workload pressure while increasing the productivity. The use of AI is expected to reduce medical costs as there will be greater accuracy in diagnosis and better prediction in better disease prevention and treatment.

1.7: Introducing Machine Learning in the Medical field

Machine learning (ML) disciplines offer computational algorithms and learning processes for extracting new information from big databases. Applications of machine learning are beneficial for developing methods to issues like as classification, prediction, pattern recognition, and knowledge extraction, where the input is a set of examples and the output is the prediction of future examples. In this respect, machine learning can give approaches and tools to aid in the diagnosis and prognosis of medical issues, where the input is a dataset containing subject information and the output is a diagnosis or prediction of a specific condition. Despite the fact that diagnosis and prognosis are relatively simple ML problems, clinical decision-making using ML applications is not yet widely used by the medical community, because such a complex task necessitates not only accuracy, but also physician specialists' confidence in the functional use of ML approaches in medicine.

Some unique needs must be considered in order to properly apply an ML application in clinical decision-making situations. The evolution of some risk variables, for example, is often connected with the prediction of illness progression; in the case of some chronic diseases (e.g., cancer, cardiovascular diseases, and diabetes), the risk factors include non-changeable features like age or gender. Because there is presently no medical treatment for correcting these biological properties, using such non-changeable attributes to forecast the development of a disease may not be as beneficial for averting disease evolution. As a result, most machine learning systems focus on variable features, making the prognosis task more difficult and complex. Another key factor to consider is the requirement to get interpretable estimates in order to convey meaningful information to medical personnel regarding the condition. Symbolic learning approaches (e.g., decision trees and rules systems) are commonly used to do this, as they allow decisions to be presented in a simple and

understandable manner. However, using a symbolic learning method to create a more understandable model typically means sacrificing prediction accuracy.
Another issue that frequently obstructs high overall performance in the analysis of medical datasets is that they typically have an unbalanced class distribution, with a majority or negative class of healthy people (normal data) and a minority or positive class of sick people (the important class) with higher erroneous classification costs.

Because the performance of standard ML algorithms is often overwhelmed by the majority class, ignoring the minority class examples and obtaining results with acceptable accuracy and specificity (healthy subjects diagnosed correctly), but low sensitivity, the latter has a higher rate of misclassification (sick subjects diagnosed correctly). In addition to creating ML techniques that achieve acceptable overall performance and give interpretable prognostic information for medical professionals, the capacity to assist judgments and minimize the number of medical tests required for a solid diagnosis is also needed. A measure of diagnostic or prognosis reliability is also vital, since this will offer medical professionals enough confidence to use the new strategy. On the other hand, having a technique that can generate solid predictions based on a minimal amount of information about the patient is desired, because collecting such information is typically expensive, time demanding, and sometimes damaging to the patient.

1.8: Organization of the thesis

1. In Section 1 of the report, we have introduced out project and we have given a rough idea about the techniques we have used in this project. We have talked about the problem statement as well as the surroundings that motivated us to take this project. We have also talked about how AI and ML is introduced in the field of medical diagnosis
2. In the Section 2 of the report, we give a brief idea of the resources and the existing algorithms from which we have derived our idea. We have also spoken about the core area of our project in this section. We discuss about the observations gained from these research papers.
3. In Section 3 of the report, we analyse our system and speak about the existing system and its disadvantages. We propose our idea and explain the methodology we follow in the making of our smart medical diagnostic system. We also speak about the novelty, requirements and the applications of this project.
4. In Section 4 of the report, we implement the system by giving the designs of each of the5 modules involved in making the system and explain each module in depth. Also, in the last sub-section of this section each and everyone highlights the contributions done by him/her in the report.
5. In Section 5 of the report, we analyse the performance of the system based on real-time data and discuss the accuracy. We also perform a comparative study with respect to the existing models and prove our model to be better than others. We also study the optimization techniques used in order to increase the accuracy.
6. In Section 6 of the report, we discuss about the enhancements that are possible in the model in the future. We discuss the current applications and some of the future enhancements.
7. In the Section 7 of the report, we show some of the images that the user will see while using our model
8. In this Section 8, we give some additional information on IBM Watson Studio
9. In this Section 9, we list down some references and research papers that we referred to by acknowledging their work.

2. LITERATURE REVIEW

2.1 Introduction

- Need of medical diagnosis in its early stage is a must in order to quickly start the treatment of the patient. This also increases the chances of the patient getting better treatment and getting healthier without much affect to the body.
- Our smart medical system can be used for identification of pneumonia by the help of X-RAY scans. We perform risk analysis as well and then the patient can decide whether and how urgent he needs to visit a doctor.
- This system can be used by people living in the rural areas which lack medical facility and thus they can perform self-diagnosis rather than waiting without doing anything and thus can start the treatment quickly
- The project consists of development of 2 different models. One of the models can be run without internet and this model can be used in rural areas where internet is not easily available. This model is known as the local model. Another model is the model that can be used only with the help of internet. This model is trained on IBM Watson Studio.
- In this section, we will discuss some of the resources that we have used in order to develop this project. We will also have a brief overview at the existing systems of the same kind as our project.

2.2 Core area of the project

Our project revolves around the idea of creating such a system which can perform diagnosis of pneumonia. This can help the patient predict whether he needs to rush to the doctor or not. Pneumonia can become risky if not detected early and hence can lead to death of the patient. We are planning to use python in order to training and testing of our model. The database will be a mixture of databases from online resources as well as from the surveys that we collect regarding various diseases. The database also uses real time X-RAYs of the patient. According to the X-RAY of the patient, we will perform feature extraction and then we will use those features in order to predict the disease. We also aim to develop a model on IBM Watson Studio which involves python programming for training and testing of data. After this, we make a webpage by the help of JavaScript, CSS and HTML. We also aim to use Spyder for Flask deployment for local models and IBM trained models.

2.3 Some Existing Methodologies

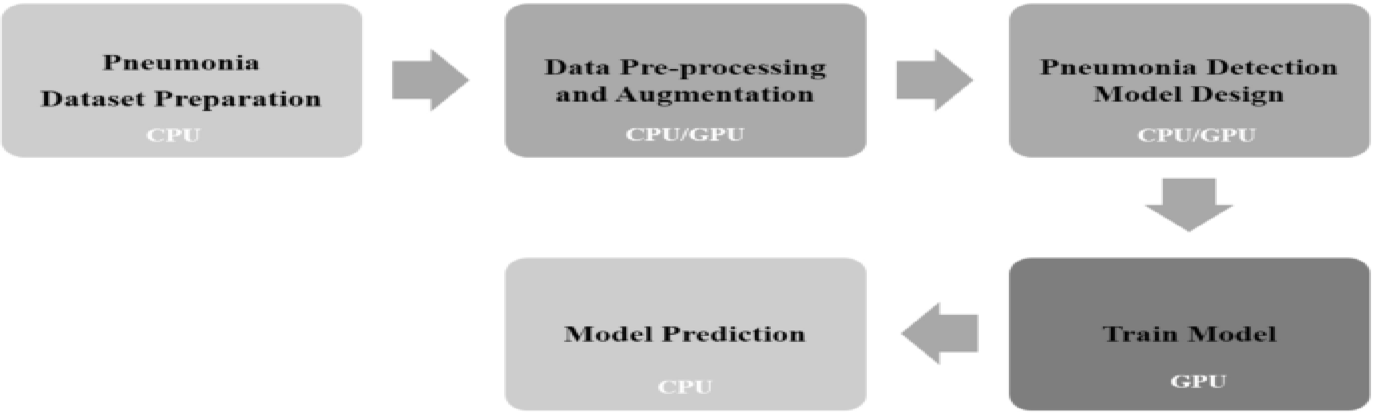


Fig. 2.3.1: - Methodology 1

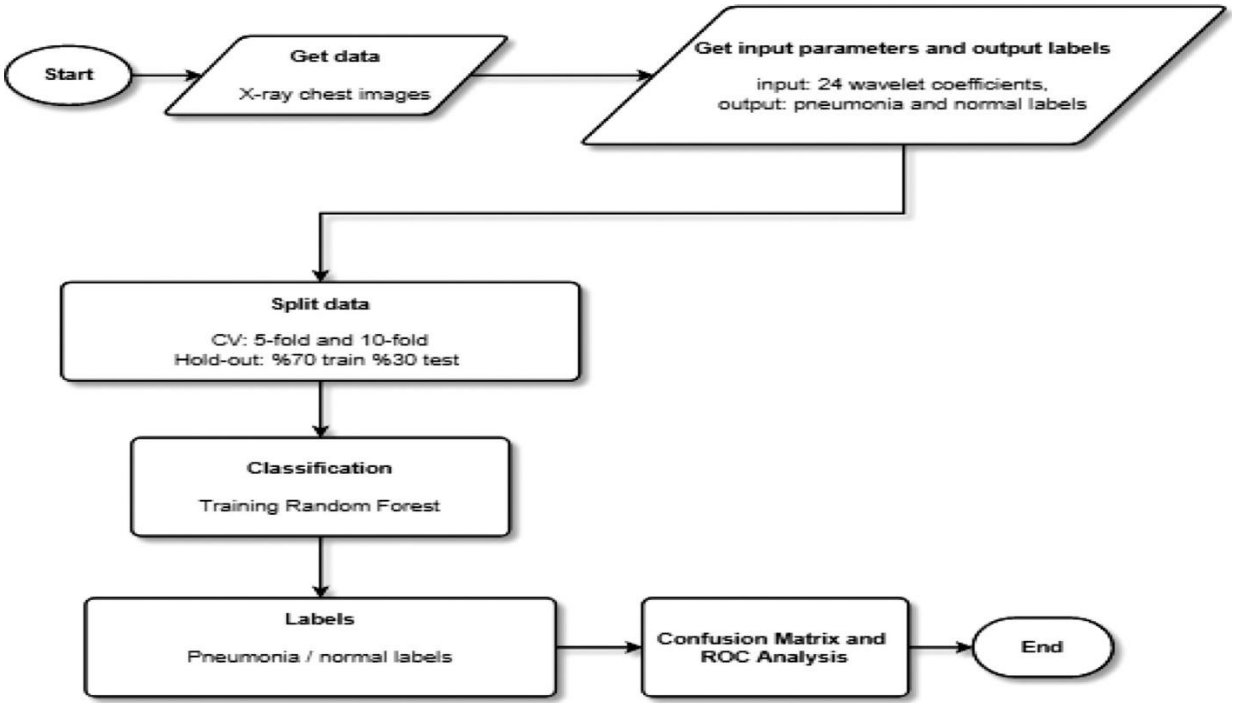

Start
Get data
X-ray chest images
Get input parameters and output labels
input: 24 wavelet coefficients,
output: pneumonia and normal labels
Split data
CV: 5-fold and 10-fold
Hold-out: %70 train %30 test
Classification
Training Random Forest
Labels
Pneumonia / normal labels
Confusion Matrix and
ROC Analysis
End

Fig 2.3.2: - Methodology 2

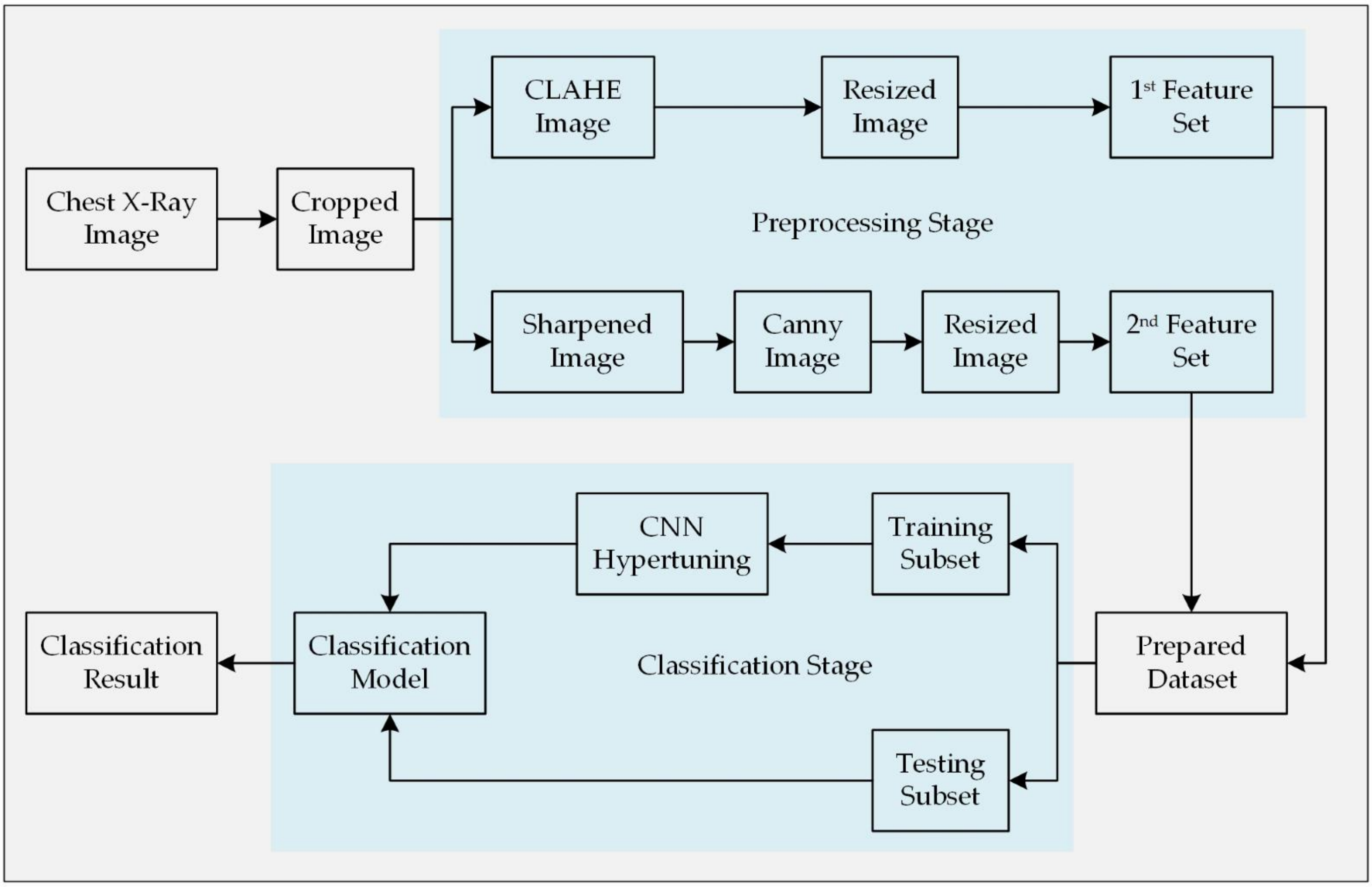


Fig. 2.3.3: - Methodology 3

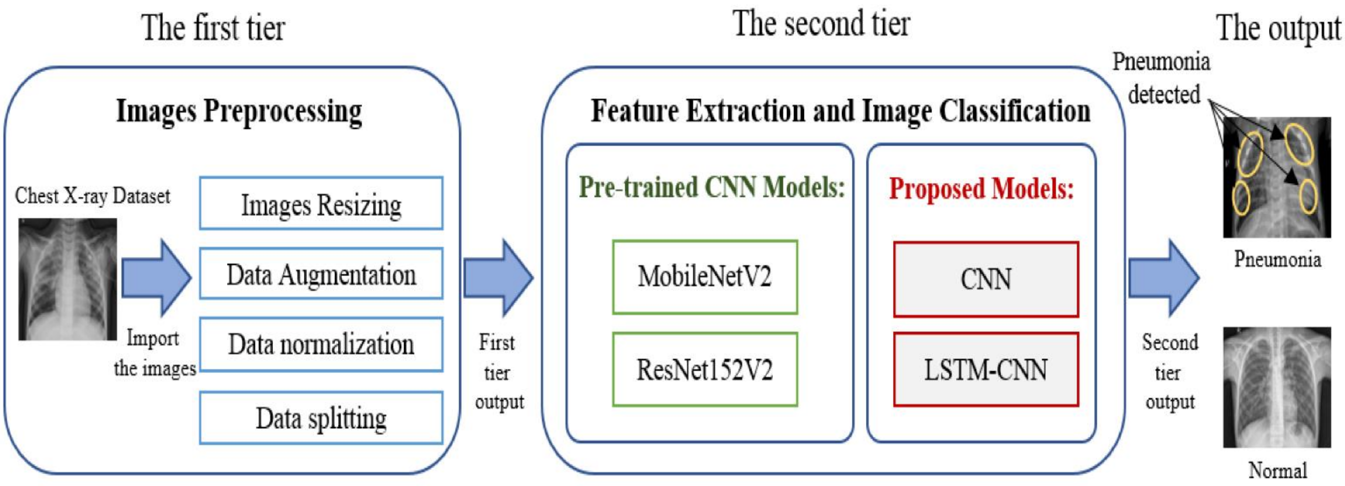


Fig. 2.3.4: - Methodology 4

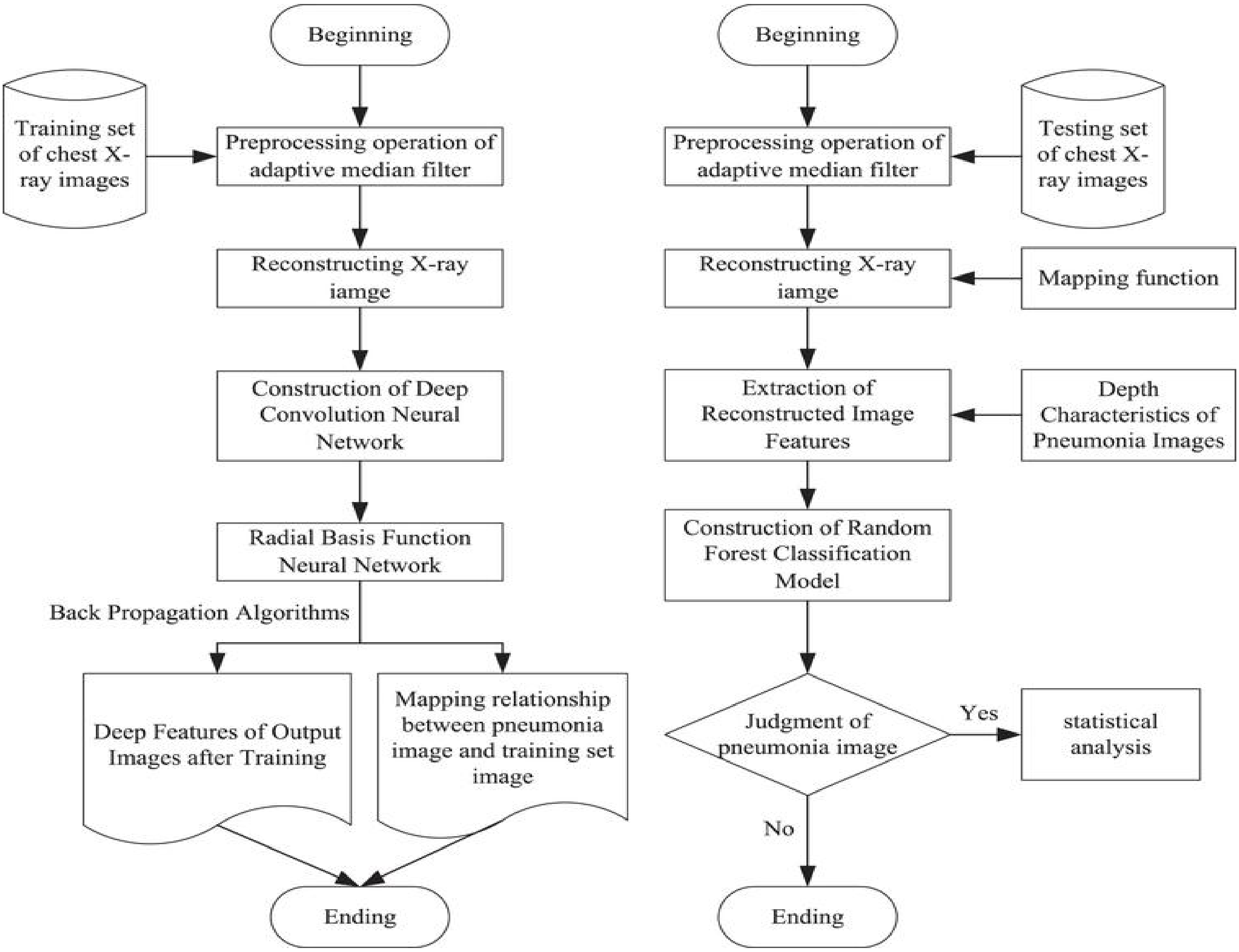

Beginning
Training set of chest X-ray images
Preprocessing operation of adaptive median filter
Reconstructing X-ray iamge
Construction of Deep Convolution Neural Network
Radial Basis Function Neural Network
Back Propagation Algorithms
Deep Features of Output Images after Training
Mapping relationship between pneumonia image and training set image
Ending
Beginning
Testing set of chest X-ray images
Preprocessing operation of adaptive median filter
Reconstructing X-ray iamge
Mapping function
Extraction of Reconstructed Image Features
Depth Characteristics of Pneumonia Images
Construction of Random Forest Classification Model
Judgment of pneumonia image
Yes
statistical analysis
No
Ending

Fig. 2.3.5: - Methodology 5

2.4 Any other method used in the project

Machine learning algorithms have found a way in the medical field and are proving to be highly accurate and successful in predicting many diseases. We are planning to implement various diagnostic libraries that are available in Python. Apart from that we will try to use Custom Vision API if we are not able to increase the accuracy of the model by coding itself. We will try to incorporate the accuracy by the help of coding at first. Also, we aim to perform feature extraction on the entered symptoms by the user and thus print the predicted results on the basis of risk analysis. We have also used IBM Watson Studio where we perform advanced python programming in order to train the model in a more accurate way. Convolution Neural Networks (CNN) is used in the complete training algorithm.

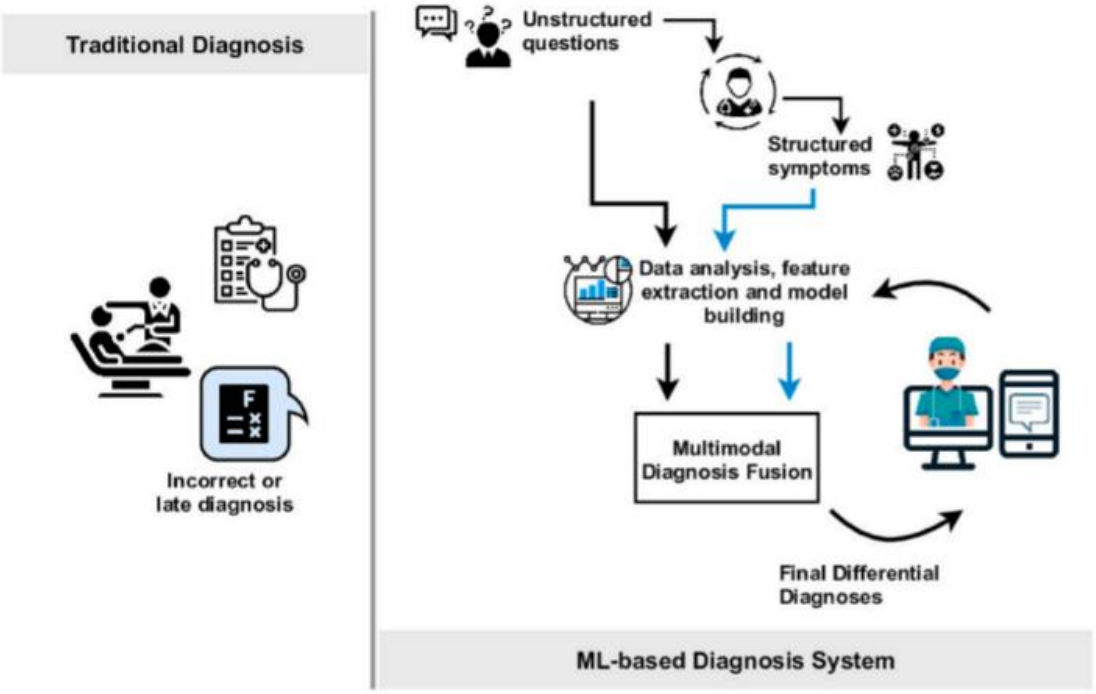

Traditional Diagnosis
Incorrect or late diagnosis
Unstructured questions
Structured symptoms
Data analysis, feature extraction and model building
Multimodal Diagnosis Fusion
Final Differential Diagnoses
ML-based Diagnosis System

Table 2.4.1: - Difference between Traditional Diagnosis and ML Based Diagnosis

2.5 Observations From Literature Survey

1. Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan and Ankush Mittal, in their paper, have appraised the functionality of pre-trained CNN (Convolutional Neural Network) models to utilize feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. They have analytically determined the optimal CNN model for the purpose. Statistical results obtained demonstrate that pre-trained CNN models employed along with supervised classifier algorithms which prove to be highly beneficial in order to detect pneumonia. But in this model, the accuracy was very less. Also, high level image processing was required.
2. M. Togaçar, B. Ergen, Z. Cömert and F. Özyurt in their paper have made use of the convolutional neural network as a feature extractor, alongside some of the existing convolutional neural network models that are AlexNet, VGG-16 and VGG-19 were utilized so as to realize this specific task. Then, the number of deep features was reduced from 1000 to 100 by using the minimum redundancy maximum relevance algorithm for each deep model. Accordingly, they were able to achieve 100 deep features from each deep model and hence combined these features to provide an efficient feature set consisting of a total of 300 deep features. In this step of the experiment, this feature set was given as an input to the decision tree, k-nearest neighbours, linear discriminant analysis, linear regression, and support vector machine (SVM) learning models.

3. Mohammad Farukh Hashmi, Satyarth Katiyar, Avinash G Keskar, Neeraj Dhanraj Bokde and Zong Woo Geem went for a novel approach based on a weighted classifier, which combines the weighted predictions from the state-of-the-art deep learning models such as ResNet18, Xception, InceptionV3, DenseNet121, and MobileNetV3 in an optimal way. This approach is a supervised learning approach in which the network predicts the result based on the quality of the dataset used. Finally, the model is evaluated. The final proposed weighted classifier model is able to achieve a test accuracy of 98.43% and an AUC score of 99.76 on the unseen data from the Guangzhou Women and Children's Medical Center pneumonia dataset. The false positives were greater than the false negatives, and hence, the classification error of pneumonia suffering patients as healthy was comparatively lesser, which is ideally required in medical diagnosis.
4. Hossam Faris, Maria Habib, Mohammad Faris, Haya Elayan, Alaa Alomari in "An intelligent multimodal medical diagnosis system based on patients' medical questions and structured symptoms for telemedicine,". In this research paper the authors have shown a methodology which consist of feature extraction and vectorisation using the scikit library. The authors have also compared different ways of classification and regression and have showed us the results.
5. Okeke Stephen, Mangal Sain, Uchenna, and Do-Un Jeong proposed a convolutional neural network model trained from scratch to classify and detect the presence of pneumonia from a collection of chest X-ray image samples. Unlike other strategies that depend entirely on transfer learning draws near or conventional handmade strategies to accomplish a wonderful arrangement performance, they built a convolutional neural network model from scratch to extract features from a given chest X-beam picture and group it to decide whether an individual is tainted with pneumonia. This model could help moderate the dependability and interpretability challenges regularly confronted when managing medical imagery.
6. Daniel S Kermany, Michael Goldbaum, Wenjia paper, here, it established a diagnostic tool based on a deep-learning framework for the screening of patients with common treatable blinding retinal diseases. The exhibition of model relies exceptionally upon the loads of the pre-trained model. Subsequently, the presentation of this model would probably be upgraded when tried on a bigger ImageNet dataset with further developed deep-learning techniques and architecture.

2.6 Summary

After going through a number of papers and analyzing different algorithms based on efficiency and accuracy, we along with the help of our internal guide Dr. Kanchan Lata Kashyap, have come up with a final architecture of the system. The proposed system and the overall architecture are described in detail in next section of the report. Using many learning resources and research papers by many eminent scholars we have designed our algorithms and models and thus the smart medical diagnosis system is efficient, accurate and easy to implement and affordable to use.

3. SYSTEM ANALYSIS

3.1 Existing Technologies and Its Disadvantages

Machine learning is a branch of artificial intelligence in which a machine learns and executes tasks by being trained. There are algorithms for supervised learning (under the control and "guidance" of a human expert), in which we are initially aware of both input and outcomes, and unsupervised learning (requiring very little human interaction or domain expert's assistance), in which we are unaware of the results. A machine can be taught to understand a notion by providing examples and constructing pattern models that can distinguish between two or more items. Machine learning assists experts in the medical area in handling massive and complex medical data as well as investigating the outcomes. This process' output can be used to enhance research. As a result, when machine learning is used in healthcare, patients' trust in medical research grows, making it easier to forecast disease using machine learning algorithms.

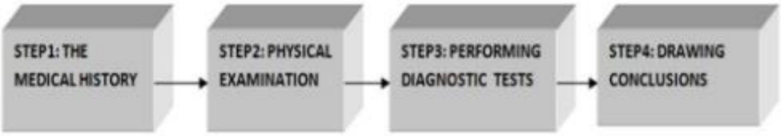

STEP1: THE MEDICAL HISTORY
STEP2: PHYSICAL EXAMINATION
STEP3: PERFORMING DIAGNOSTIC TESTS
STEP4: DRAWING CONCLUSIONS

Fig. 3.1.1: - Block Diagram of existing system

Sometimes, illness is not early detected by human experts, in such types of cases machine learning can be used to detect early stages of the disease before its occurrence or it becomes dangerous to someone. In this way, it can help to prevent future problems as “Prevention is better than cure”. The popularity of machine learning in different areas has tended towards machine learning algorithms that produce correct outcomes as compared to traditional models with little processing of raw data. Machine learning algorithms like Decision trees, Support Vector machine, Multilayer perceptron, Bayes classifiers, K-Nearest Neighbor, Ensemble classifier techniques, etc are used to determine various ailments.

Machine learning algorithms can help predict diseases quickly and accurately. Observations or information, such as examples, direct experience, or instruction, start the learning process. The algorithms, in particular, hunt for data trends and make smarter conclusions. The main goal is for the machines to learn on their own, without the need for human intervention, and change their responses accordingly.

The Intended contribution of AI in the field of medical sciences to develop programs that can help a medical expert in practicing expert and more accurate diagnosis. In machine learning, disease forecasting is extremely significant. Various types of diseases can be predicted using ML techniques. Here, we examine how machine learning techniques are used to predict various disease types. Machine learning has given computers new capabilities that we could never have imagined. Machine learning is a field of AI that gives machines the power to learn itself by examples in order to analyze how different models perform in ML without using human judgment. Machine learning, all the features extracted by a domain specialist to minimize the complications of data and to develop patterns in such a way that would be easily visible to ML algorithms. However, deep learning-based techniques can extract features manually without human intervention, the only condition is to make precise decisions in which the testing data could be accurate. This technique eliminates the requirement of a domain expert for feature extraction.

Disadvantages of existing systems: -

- One of the major problems is the imprecision and inaccuracy in many cases.
- According to some studies, between 20 and 50 percent of all advanced imaging results fail to offer information that improves patient outcome, albeit these studies do not take into consideration the importance of negative imaging results in influencing patient care decisions.
- Not applicable among patients with complex diseases. Lack of flexibility and individuality.
- Another limitation of diagnosis is that it can only be approved by research, discussion, and consensus. Even if the symptoms are overlapping, a diagnosis can differ from one to the next. Different diagnosis from different psychologists may cause the patient to become confused.
- Disadvantages of current existing systems is the lack of proper dataset in order to perform detection. These types of datasets tend to go towards overfitting and hence can be harmful to be used in daily lives.

3.2 Proposed System and its Advantages

We present a model for detecting Pneumonia with the use of X-Ray pictures. We begin by using Kaggle to download a dataset. There are 5216 training photos, 624 testing images, and 16 validation images in the dataset. Each piece of information is divided into two categories: normal and pneumonia. The images of a normal X-RAY of the lungs are called normal images. X-RAYs of infected lungs with pneumonia are used to diagnose pneumonia. First, we use the Image Data Generator Class to execute image processing. Image data augmentation is a technique for increasing the size of a training dataset artificially by producing changed versions of the images in the dataset. The ImageDataGenerator class is created, as well as the settings for the several sorts of data augmentation. For picture data, there are five basic types of data augmentation approaches. The width shift range and height shift range variables are used to shift the image. The horizontal flip and vertical flip arguments flip the image, the rotation range argument rotates it, the brightness range argument brightens it, and the zoom range parameter zooms it. To the Train, Test, and Validation sets, apply ImageDataGenerator functionality.

After that, we begin to train our model. Convolution Neural Networks, abbreviated as CNN, are used to train the models. We'll start by setting up the model. The Sequential class is used to build linear network layer initializations, which are subsequently combined to form a model. In the example below, we'll create a model with the Sequential and then add layers to it with the add () method. The convolution layer, the initial layer of the neural network model, will be introduced. The Convolution2D class is used to generate a convolution layer. It accepts as parameters the number of feature detectors, the size of the feature detectors, the image's expected input shape, and the activation function. This layer generates a feature map by applying feature detectors to the input image (features from the image). Max Pooling finds the most significant element from the feature map region covered by the filter. As a result, the output of the max-pooling layer would be a feature map with the most prominent features from the previous feature map. A pooling layer is put after the convolution layer. The MaxPooling2D class can be used to add a maximum pooling layer. It uses the size of the pool as a parameter. The size of the pooling matrix should be as small as possible (2,2). It returns the feature maps that have been pooled. To convert n-dimensional arrays to 1-dimensional arrays, utilize the flatten layer. This 1D array will be fed into the ANN layers as input. Three dense layers are added, each of which requires a certain number of units/neurons. It is not required to specify the activation function or the type of weight initialization.

After adding all the required layers, the model is to be compiled. Compilation is the last stage in the model creation process. We may go on to the training step after the compilation is complete. In the learning process, the loss function is utilised to discover mistake or deviation. During the model compilation phase, Keras requires a loss function. By comparing the prediction and the loss function, optimization is an important procedure that optimizes the input weights. We're using Adam Optimizer in this case. Fit the neural network model with the train and test set, number of epochs and validation steps. The weights are to be saved for future use. The weights are saved in as .h5 file using save(). Finally after the training is complete for 20 epochs, we test the model. We observe that at the end of the last epoch, we obtain an accuracy of 93.12% of training accuracy. We load a testing image and observe that the accuracy of the model is 99.99974%

Next, we move towards the flask application. After the model is built, we will be integrating it to a web application so that normal users can also use it. The users need to give the scan to know if the tumor is present or not. In this part, we'll create a web application that's linked to the model we created before. A UI is provided for the uses where he has uploads an image. The uploaded image is given to the saved model and prediction is showcased on the UI. We build an HTML page where the user can upload an image and clicks on the predict button to see the result. We also build main.js and main2.css file for the backend. We build the flask file 'app.py' which is a web framework written in python for server-side scripting. We initialize the flask file by importing various libraries and also load the model into the flask file. After running the flask file, we can observe that the program will run on localhost:5000

After building the flask file, we start training our model by the help of IBM Watson studio. For that first, we create an IBM academic initiative and an IBM cloud account. We register the services of Watson studio and machine learning and then we launch Watson studio. Next, we start training the model on IBM. We run the same procedure and run till 20 epochs have been completed. But before training, we have to upload the .zip file of the dataset and then we have to unzip the same dataset for using it. Watson learning -y2 and Machine learning -lf are registered by the location of Dallas in order to use all the services completely. After registering the services on IBM Cloud, we launch these services by click on the Watson studio service.

We can observe that a tab opens with Watson studio. Here we first of all set up the working directory. We load the images in Watson studio and start working on the notebook. We check that all the images are correctly classified and loaded or not. After data augmentation, we shift to model building. Then sequential layer, Convolution 2D layer, MaxPooling 2D layer and the flatten layer are all added. There are a total of 3 dense layers added to this model with the activation functions of relu and sigmoid. After training, we observe that on IBM we achieve an accuracy of 94.73%. But the testing accuracy that is achieved on IBM Watson Studio is 99.99974%. After the training of the model has been completed, we start testing the model. After the message of SUCCESS has been displayed, we generate a unique ID which will help us download the model on our own system. We download the model trained on IBM and then deploy the project on IBM. It is now shown in the deployments section of IBM. Now the model that we have downloaded from IBM is saved and then we again follow the same steps as above for application development using flask. Thus, this is how we propose our methodology in order to detect pneumonia using the X-RAYs.

Advantages of the proposed system: -

- The suggested approach successfully classifies the photos to determine whether or not the person has pneumonia. The maximum accuracy that has been observed in 99.99974% on testing dataset. This type of accuracy can be used in hospitals, diagnostic centres and other medical facilities.
- The dataset that has been used in the proposed system is not at all prone to overfitting. Live, real-time data has been used in order to correctly predict the image.
- It involves zero cost of maintenance and installation and can be used very easily be any human.
- Need of the doctor will be reduced for diagnosis purpose and thus doctors can concentrate on more serious cases.
- Pneumonia was one of the after-effects of COVID19 and thus pneumonia detector can help the society by reducing fraudulent detection which many doctors did in the greed of money.
- It is user friendly. Modelling is simple and requires little formal statistical understanding.
- The classical way of detecting pneumonia is by taking out an x ray of the lung and then wait for the doctor's appointment to detect whether the patient has pneumonia or not, but using our model a lot of

time, money, effort and hassle can be saved, because the patient has to just click a picture of the x ray and upload it on our website, which can detect pneumonia within seconds.

- The website can be is easily available on the internet.
- The accuracy of the model increases over time as we keep increasing the number of epochs.
- It helps the community by not only saving money but also giving accurate results which can save a lot of lives by providing correct medication at correct time to prevent pneumonia.
- It can also be an aid to doctors, as it saves a lot of time for them.
- It can prove to be extremely useful during pandemic situations like COVID 19, as it can cut the hassle of detecting between pneumonia and covid using this model.
- The accuracy of the IBM trained as well as the locally trained model is very high and thus it can be used in daily life. Prediction can be done by sitting at home and this allows the doctors to focus on more serious cases.
- The model is capable of capturing non linearities between predictors and outcomes.

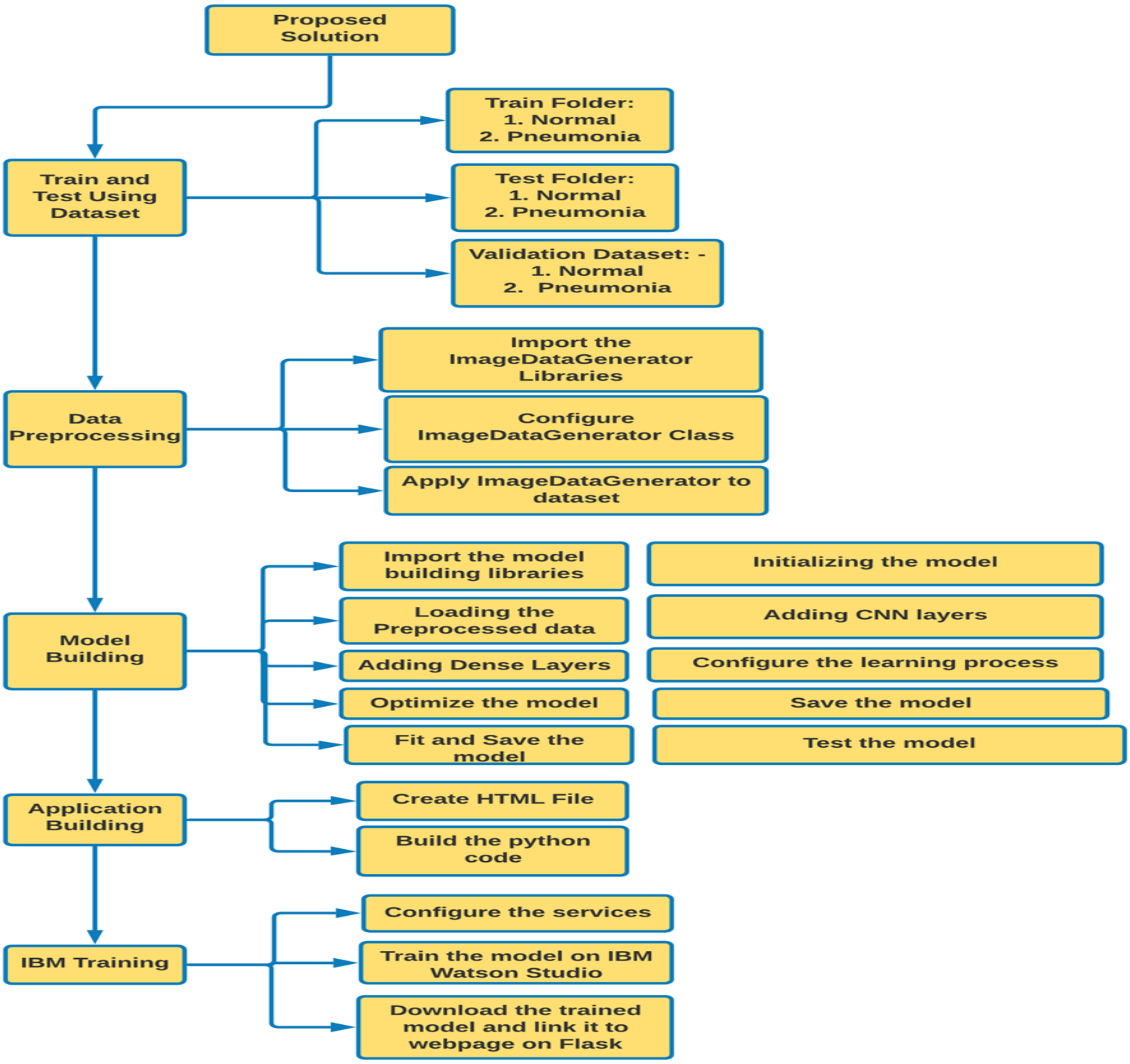

Proposed Solution
Train and Test Using Dataset
Train Folder: 1. Normal 2. Pneumonia
Test Folder: 1. Normal 2. Pneumonia
Validation Dataset: - 1. Normal 2. Pneumonia
Data Preprocessing
Import the ImageDataGenerator Libraries
Configure ImageDataGenerator Class
Apply ImageDataGenerator to dataset
Model Building
Import the model building libraries
Loading the Preprocessed data
Adding Dense Layers
Optimize the model
Fit and Save the model
Initializing the model
Adding CNN layers
Configure the learning process
Save the model
Test the model
Application Building
Create HTML File
Build the python code
IBM Training
Configure the services
Train the model on IBM Watson Studio
Download the trained model and link it to webpage on Flask

Fig 3.2.1: - Proposed System Methodology

4. SYSTEM IMPLEMENTATION

4.1 Introduction: -

We have proposed a total of 5 different modules in the complete implementation of the system. The system is complex to code but easy to use by the user. The backend part is completely hidden from the user. In this section, we will explain the functionality of each and every module in depth.

4.2 Data Preprocessing Module: -

This is the first and one of the most important modules in the implementation and proper working of the system. Medical data is usually highly fragmented and very tough to be processed. This is because the size of images in the dataset may vary from image to image. Not only the size, many other parameters also vary like brightness, shear, rotation etc. It is important to first of all analyze all this data properly and apply the right number of parameters in order to make the data ready and obtain the maximum accuracy on the model. The dataset for this model has been obtained from Kaggle, hand-picked images from Google and also the dataset has been fed with some real-time images of patients who suffered from pneumonia in the past. All these images are X-RAY images of chest in which lungs are visible clearly. The dataset consists of a total of 5216 training images, 624 testing images and 16 validation images. Each of the data is split into normal and pneumonia. The complete dataset has been firstly split into normal as well as pneumonia folder. The normal folder contains all the X-RAY of a person not infected with pneumonia whereas the pneumonia folder consists of all the X-RAYs of the person who is infected with pneumonia. Further, we segregate the images in this folder into training, testing and validation dataset. Image data augmentation was performed on the dataset to achieve the test, train and validation sets as to improve performance and outcomes of the model by forming new and different examples to datasets.

WORKING: -

Image Data Augmentation is a way in which we can enhance the quality of the image and perform operations on the image. ImageDataGenerator class is instantiated and the configuration for the types of data augmentation.There are five main types of data augmentation techniques for image data; specifically.

- Shifting of the image via the width_shift_range and height_shift_range arguments.
- Flipping of image via the horizontal_flip and vertical_flip arguments,
- Rotation of the image via the rotation_range argument,
- Enhancing the brightness of the image via the brightness_range argument,
- Increasing/Decreasing the sie of the image via the zoom_range argument.
- Apply ImageDataGenerator functionality to Train, Test, and Validation set.

The images are divided in the 90:10 ratio to maintain the bias-variance trade-off. In order to perform data preprocessing, we use many different functionalities so that the outliers from the dataset can be removed. Initially, we set the batch_size variable to 15. We initialize the image_size variable to (120,120). Next, we initialize two different functions. The first function is the train_datagen function. This function use ImageDataGenerator functions in which we can input the arguments according to which we want to process the data. We have set the width_shift_range to 0.2, length_shift_range to 0.2, horizontal_flip to “True”, rotation_range to 15 and rescale to 1./255. The next function is the test_datagen function in which we have only one argument in the ImageDataGenerator class. This is the rescale argument set to 1./255. After setting all the arguments, we have prepared to the data preprocessing part. One by one we pass the images in their respective functions. First, we pass the training images from the train_generator function. We initialize the target_size to image _size which is (120,120) and class mode to binary. The same is done with testing as well as the validation images in the dataset.

CATEGORY	TRAINING SET	TESTING SET	VALIDATION SET
Normal	1341	234	8
Pneumonia	3875	390	8
Total	5216	624	16
Percentage	89.07	10.65	0.27

TABLE 4.2.1: - SEGREGATION OF IMAGES IN THE DATASET

4.3 Local Model Development Module: -

In this section, we build a model to detect pneumonia from images of x-rays of lungs and thus classify them for the disease. This model is then trained and tested on the chosen dataset. After this we can use the model for prediction of pneumonia. This model can be run without the help of internet. The accuracy obtained on this model is 93.52% and the testing accuracy is 99.99974%.

WORKING: -

This module development includes the following steps: -

- Import the model building Libraries
- Initializing the model
- Adding CNN Layers
- Adding Hidden Layer
- Adding Output Layer
- Configuring the Learning Process
- Training and testing the model
- Model has to be saved

There are 2 ways in which Keras will define a neural network:

- Sequential
- Function API

The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. We will be using the Sequential constructor to create a model, which will then have layers added to it using the add () method. Now, will initialize our model. The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes a number of feature detectors, feature detector size, expected input shape of the image, activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image). The Activation Function are the functions which help us to decide if we need to activate the node or not. Such functions are used to introduce the concept on non-linearity in the model. The Max Pooling selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map. After the convolution layer, a pooling layer is added. Max pooling layer can be added using MaxPooling2D class.

It takes the pool size as a parameter. Efficient size of the pooling matrix is (2,2). It returns the pooled feature maps. (Note: Any number of convolution layers, pooling and dropout layers can be added). The flatten layer is used to convert n-dimensional arrays to 1-dimensional arrays. This 1D array will be given as input to ANN layers. Three dense layers are added which usually takes number of units/neurons. Specifying the activation function, kind of weight initialization is optional. After adding all the required layers, the model is to be compiled. For this step, loss function, optimizer and metrics for evaluation can be passed as arguments. After this, compilation in the last phase of the model. Also, a function known as loss function is defined in order to find the error and deviation in the learning process. Keras requires loss function during model compilation process. We then fit the neural network model with the train and test set, number of epochs and validation steps.

Loss value shows that how poorly or well a model is working after each iteration. An accuracy metric is used for calculation of the algorithm's performance so that it can be interpreted. The weights are to be saved for future use. The weights are saved in as **.h5 file using save ().** model.summary() can be used to see all parameters and shapes in each layer in our models. The model is to be tested with different images to know if it is working correctly. We then import the packages and load the saved model. After that we load the test image, pre-process it and predict the classification with the help of the model. Pre-processing the image includes converting the image to array and resizing according to the model. We then give the pre-processed image to the model to know to which class your model belongs to. The model predicts whether the given image of an X-Ray corresponds to pneumonia or not. The accuracy of this model comes out to be 93.52% as training accuracy and 99.99974% as testing accuracy.

4.4 Flask Deployment Module: -

This is the start of the process of creating a web application and deploying the python models that we have on our local system. Next, we move towards the flask application. After the model is built, we will be integrating it to a web application so that normal users can also use it. The users need to give the scan to know if the tumor is present or not. In this section, we will be building a web application that is integrated to the model we built. Spyder is used in order to perform flask deployment A UI is provided for the uses where he/she has to upload an image. The uploaded image is given to the saved model and prediction is showcased on the UI. We build an HTML page where the user can upload an image and clicks on the predict button to see the result. Web development part will be explained in the last module. We also build main.js and main2.css file for the backend. We build the flask file 'app.py' which is a web framework written in python for server-side scripting. We initialize the flask file by importing various libraries and also load the model into the flask file. After running the flask file, we can observe that the program will run on localhost:5000. The flask folder consists of 3 sub-folders. Static folder consists of main2.css and main.js. These are CSS files and Javascript files used for backend development. Base.html file is in the template folder for frontend web development. Uploads folder consists of al images uploaded in the model while running it. It also consists of app.py and app_IBM.py files. Flask Deployment has been performed by the help of a software known as Spyder. Spyder is readily available of Anaconda Prompt and is under the GPL which is General Public License. It is a free to use software and easily accessible and provides a platform to link our models with the python interface. The frontend development working i.e. working of CSS, JavaScript and HTML is shown in the module 5 of the system implementation

WORKING: -

We build the flask file ‘app.py’ which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application. App starts running when the “__name__” constructor is called in main. render_template is used to return HTML file. “GET” method is used to take input from the user. “POST” method is used to display the output to the user. We load the following modules:-

- sys, os, glob, numpy, tensorflow
- from tensorflow.keras.preprocessing import image
- from tensorflow.keras.applications.imagenet_utils import preprocess_input, decode_predictions
- from tensorflow.keras.models import load_model
- from tensorflow.keras import backend

Next we define a flask app using the Flask object and then load our desired model which is pneumonia.h5 using the load_model object. Flask then creates a default web page at the server id http://127.0.0.1:5000/. Using the @app.route function we map the url to the methods function that performs the GET logic. Next we define our base page for the website or the layout of the website by giving in a base.html file that we have already created. Now we define another two logics that is GET and POST using the @app.route function. Our next function is the upload function defined by us which enables the user to upload photos of chest x ray and check for presence of pneumonia. If the user requests for post method on the website then we generate another request to user for the required image using request.files method. The user can easily use the os module to browse for the x ray image in their PC and upload it. Next we resize the image according to the need of the model pneumonia.h5, in this case it has the dimensions (150, 150). Finally we use the model.predict() object to check for the presence of pneumonia using our pre-trained model pneumonia.h5 and return the user whether he has symptoms of pneumonia or not, on the website itself. Run the flask application using the run method. By default, the flask runs on 5000 port. If the port is to be changed, an argument can be passed and port can be modified.

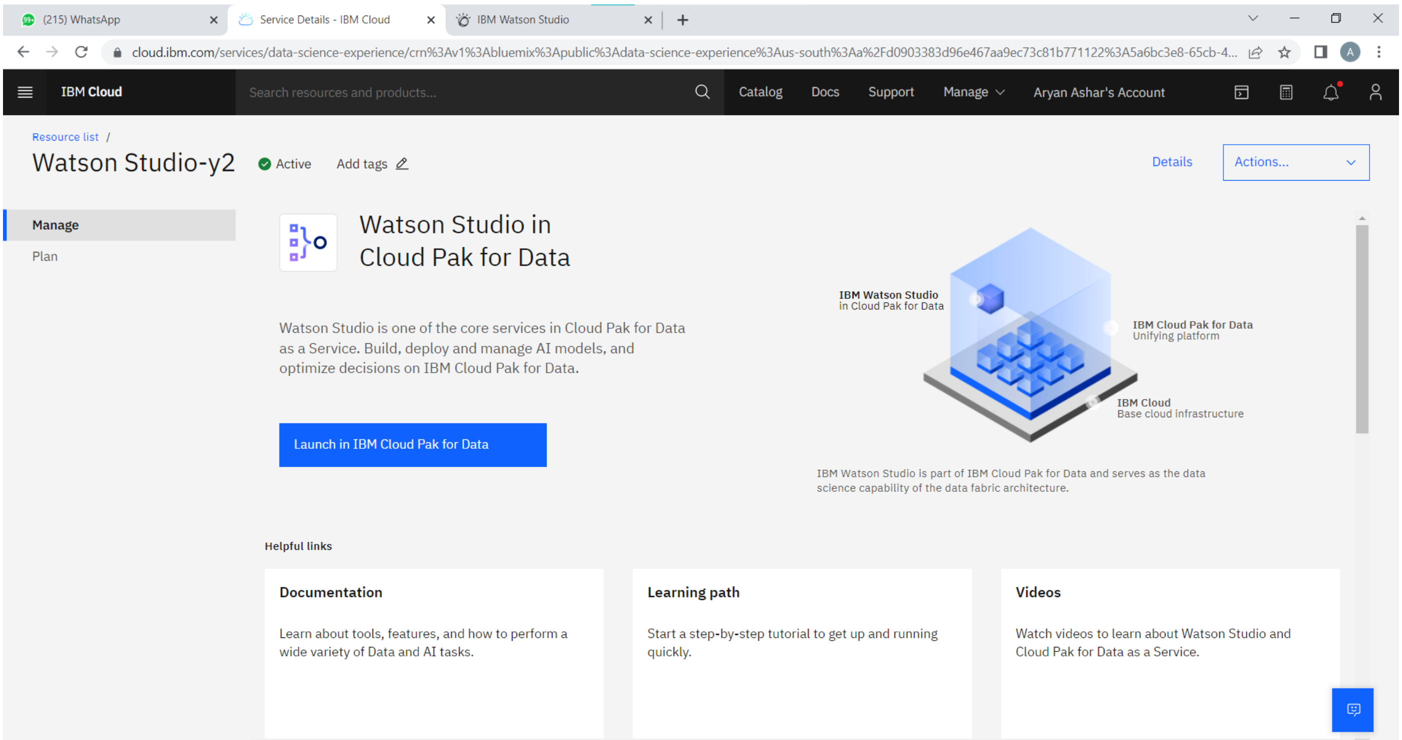
4.5 IBM Model Development Module: -

After we have trained and tested the model on the local machine, it is time to make a new model using IBM machine learning services. In this project, we have used IBM Watson Studio in order to train our model and then this model has been downloaded by the help of jupyter notebook. IBM Watson Studio is a service wherein we can build the model directly where the data exists. The service takes optimized decisions on dataset and choses the best images in order to train the model. This model will only work if internet connectivity is available. IBM allows users to scale services to fit their needs. In order to use these services, we first develop an academic account on IBM cloud. IBM Watson Studio is a service wherein we can build the model directly where the data exists. The service takes optimized decisions on dataset and choses the best images in order to train the model. IBM Watson Studio provides a service for data science engineers to work for.

In the Training_IBM folder, we have pneumo_IBM.h5 folder. This is the folder where we save the trained model which is downloaded from the IBM Watson Studio. We train a new model on IBM Watson Studio and then perform training and testing on the model. While training the model, we observe an accuracy of 94.75% and testing accuracy of   99.99974%. It is observed that the model which is trained on IBM performs better than the locally trained model. 94.75% of accuracy is seen on the IBM trained model. Also, we load the zipped dataset that we have on IBM cloud. Using this dataset, we train the model and download the new model and then integrate it to our system using flask deployment as shown.

WORKING: -

Firstly, we register an academic account on IBM cloud. An academic account is created because we can use the services for free for some amount of time. We register the services on IBM cloud. A total of 2 services are registered. IBM Watson studio and Machine learning is registered for Dallas location so that we can use all services easily and without any hinderance. Next, we click on the Watson studio service and launch the IBM Cloud Pak for Data. We start training the model on the notebook of IBM. We change the working directory to /home/wsuser/work'. We install numpy, os, pandas and tensorflow and keras which will be used in order to train and test our model.


(215) WhatsApp
Service Details - IBM Cloud
IBM Watson Studio
cloud.ibm.com/services/data-science-experience/crn%3Av1%3Abluemix%3Apublic%3Adata-science-experience%3Aus-south%3Aa%2Fd0903383d96e467aa9ec73c81b771122%3A5a6bc3e8-65cb-4...
IBM Cloud
Search resources and products...
Catalog
Docs
Support
Manage
Aryan Ashar's Account
Resource list /
Watson Studio-y2
Active
Add tags
Details
Actions...
Manage
Plan
Launch in IBM Cloud Pak for Data
IBM Watson Studio
in Cloud Pak for Data
IBM Cloud Pak for Data
Unifying platform
IBM Cloud
Base cloud infrastructure
Helpful links

Watson Studio in Cloud Pak for Data

Watson Studio is one of the core services in Cloud Pak for Data as a Service. Build, deploy and manage AI models, and optimize decisions on IBM Cloud Pak for Data.

IBM Watson Studio is part of IBM Cloud Pak for Data and serves as the data science capability of the data fabric architecture.

Documentation

Learn about tools, features, and how to perform a wide variety of Data and AI tasks.

Learning path

Start a step-by-step tutorial to get up and running quickly.

Videos

Watch videos to learn about Watson Studio and Cloud Pak for Data as a Service.

Fig. 4.5.1: - IBM Watson Studio Home Page

Before starting to train and test the model, we have to upload the dataset on IBM Cloud. We upload the dataset in compressed form as our dataset is too large to go normally. After uploading the dataset normally, we have to extract it. This is done by the help of python programming. The libraries installed are: -

- numpy, pandas,os
- from tensorflow.keras.preprocessing import image
- from tensorflow.keras.models import Sequential
- from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D,Flatten
- from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img

With the help of BytesIO library in the os module of python 3.8, we unzip the dataset. We store the IBM API key safely as this will be required in downloading our model. We set the working directory and start loading the images from the folder. After this is done, we start training our model. Model training is done by the help of Convolution Neural Networks also abbreviated as CNN. We will first initialize the model. The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add () method. The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes a number of feature detectors, feature detector size, expected input shape of the image, activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image). Max Pooling selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map. After the convolution layer, a pooling layer is added. After this step, a flatten layer is added and after this we add a total of 3 different dense layers with different activation functions. 2 layers with RELU activation function are added and one layer with sigmoid activation function is added. The flatten layer is used to convert n-dimensional arrays to 1-dimensional arrays. This 1D array will be given as input to ANN layers. Three dense layers are added which usually takes number of units/neurons. Specifying the activation function, kind of weight initialization is optional. Next, we use Adam optimizer in order to optimize our model. We calculate the loss through binary cross entropy function. We run a total of 20 epochs and observe an accuracy of 94.73%. Also, the model keeps training itself after the number of epochs are increased. We save this model in a .h5 file. Even this model is observed to have a testing accuracy of 99.99974% making it very accurate to use in medicinal centers and hospitals and also for local residents. By the summary() function, we will be able to see all the parameters of the model.

After that, we start testing the model on IBM Watson Studio. We set the credentials like URL and API key that we have which is registered with the service. We import the API_Client library from ibm_watson_machine_learning. We test the model and then confirm whether model is working perfectly. After this, we open our Jupyter notebook on the local machine and write a code in order to download the model that we have made on IBM Watson Studio. We install the ibm_watson_machine_learning library on the local machine and import API_client library. We enter the credentials and define a function guide_from_space_name. We download the tested model on our local model but this requires internet service. The accuracy comes out to be more than the locally developed model and thus is better to use if we require more accuracy.

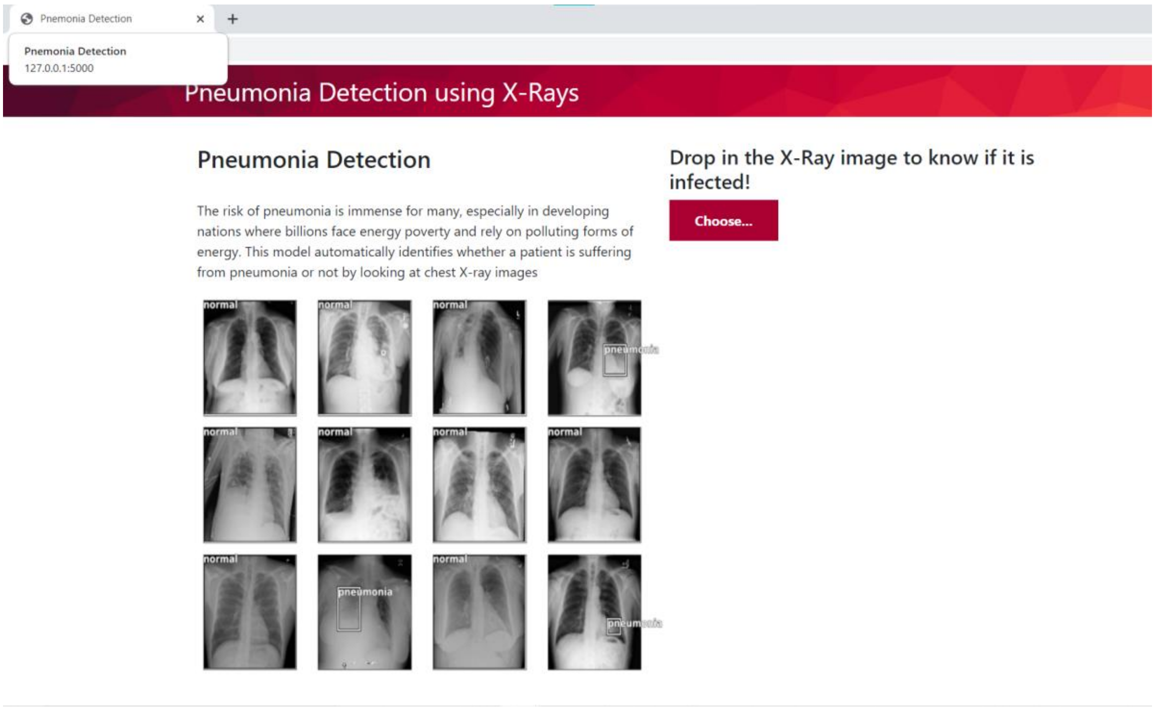
After the model has been downloaded and compiled, we need to link it with the webpage. This is done in the same way as explained in the flask deployment part with the help of Spyder and coding in python. We build the flask file ‘app_IBM.py’ which is a web framework written in python for server-side scripting. We develop the backed file in the same way we develop for app.py. Here we develop the file for the IBM trained model. The name of the model is app_IBM.py. We pre-process the captured frame and give it to the model for prediction. Based on the prediction the output text is generated and sent to the HTML to display. Next we define a flask app using the Flask object and then load our desired model which is pneumonia_IBM.h5 using the load_model object. Flask then creates a default web page at the server id http://127.0.0.1:5000/. Using the @app.route function we map the url to the methods function that performs the GET logic. It is the same way in which we build the app_IBM.py file.

4.6 Webpage Development: -

As for the Web Page development we focused on these 3 languages:

- HTML is used in order to define the content of web pages.
- CSS is used in order to specify the layout of web pages.
- JavaScript to program the behavior of web pages. It is also used in order to upload the images used in the model to the uploads folder.

The coding for all of the following is performed on the Spyder app on Anaconda Navigator. We have main2.css and main.js file which are the css and javascript files respectively. We build an HTML page where the user can upload an image and clicks on the predict button to see the result. Our page looks like: -



Pneumonia Detection using X-Rays

Pneumonia Detection

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. This model automatically identifies whether a patient is suffering from pneumonia or not by looking at chest X-ray images

Drop in the X-Ray image to know if it is infected!

Fig. 4.6.1: - Frontend webpage

CSS File

We applied CSS with HTML documents to change various styles of user interfaces in the web pages. It specified how the various HTML elements or tags are to be presented on the screen. Cascading style sheet is used with JSS(JavaScript) and HTML(Hyper Text Markup Language) in most of the websites to develop user interfaces for a lot of mobile applications and user interfaces for various web applications. CSS has controlling power on the documents of HTML, so it is easy to control various styles, such as the text color, the font style, the spacing among paragraphs, adjusting the size of columns and layout, changing the color of background and images, design of the layout and many other effects as well. The name of CSS file is main2.css

WORKING: -

CSS has been utilized to specify the webpage's physical properties and layout. The picture requirements, text fonts, positioning, space padding, and colour settings have all been added. Using the many features available in CSS, we've additionally specified the keyframes and different input types to be evaluated. To make it look more complex, basic animations such as spin have been implemented.

JavaScript File

JavaScript is a scripting or programming language that allows you to implement complex features on web pages. To enhance page’s functionality, we used JavaScript since the HTML we wrote only created a static page on the web. We have used Javascript so that the users can easily interact with our webpage. It is also used to upload the images on the webpages. JavaScript being a client-side script, speeds up the execution of the program as it saves the time required to connect to the server. JavaScript also provides various interfaces to developers for creating catchy webpages. This improves the interactivity of the user on our webpage.

WORKING: -

Starting with the ready method, which checks whether the loading of the document is complete or not. Next, we have used the readURL function to read the URL and upload the appropriate image preview. The click method is applied on the Predict Button appearing on the webpage, upon clicking the Prediction Button the loader animation is shown. The prediction results are collected by calling the API. Then the data is collected and the results are displayed, hiding the leader animation.

HTML (Hyper Text Markup Language) File

We used Markup languages to format the text to provide a visual structure to the document. We used unique tags to annotate different elements for specific purposes. In this project, HTML has been used for front end development of the project. HTML files are provided a framework by the help of tags and such tags make it convenient for us and browsers to read the document efficiently. It also allows a browser to apply CSS to the digital document, making it a powerful combination and appealing to look at. HTML aids in increasing and optimize the web pages for searchability. Small pieces of information like authentication tokens or usernames on the client using cookies can be stored.

WORKING: -

In the header section we've specified the title of the webpage using the <title> tag, then we have also specified the physical attributes like styling, positioning, fonts etc. We have also linked the main javascript file to the html file in the header section itself. We have also used various tags of HTML such as background image which can be seen in the head and body part of the HTML document. Text and image correlation has also been specified using the available features of HTML. The Information that has to be displayed on the webpage has been written under the paragraph (<p>) tag. We have also used the division (<div>) tag extensively to make various divisions in the webpage to make it look more structured and organized.

4.7 Overall Architecture Diagram: -

The below diagram gives a gist of the overall methodology that has been explained above. Note that this is just a summary and not the complete process that is depicted in the below diagram.

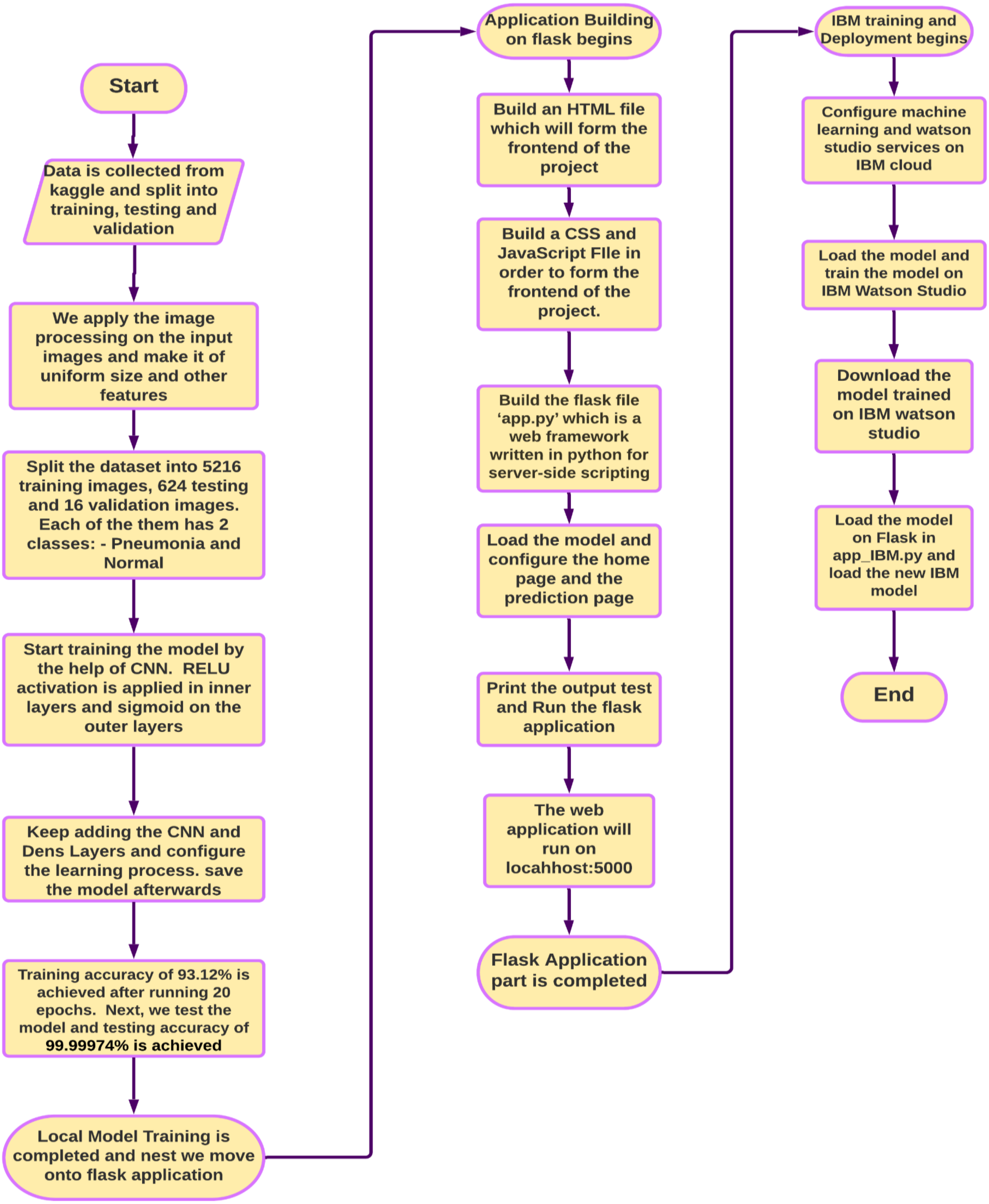

Start
Data is collected from kaggle and split into training, testing and validation
We apply the image processing on the input images and make it of uniform size and other features
Split the dataset into 5216 training images, 624 testing and 16 validation images. Each of the them has 2 classes: - Pneumonia and Normal
Start training the model by the help of CNN. RELU activation is applied in inner layers and sigmoid on the outer layers
Keep adding the CNN and Dens Layers and configure the learning process. save the model afterwards
Training accuracy of 93.12% is achieved after running 20 epochs. Next, we test the model and testing accuracy of 99.99974% is achieved
Local Model Training is completed and nest we move onto flask application
Application Building on flask begins
Build an HTML file which will form the frontend of the project
Build a CSS and JavaScript FIle in order to form the frontend of the project.
Build the flask file 'app.py' which is a web framework written in python for server-side scripting
Load the model and configure the home page and the prediction page
Print the output test and Run the flask application
The web application will run on locahhost:5000
Flask Application part is completed
IBM training and Deployment begins
Configure machine learning and watson studio services on IBM cloud
Load the model and train the model on IBM Watson Studio
Download the model trained on IBM watson studio
Load the model on Flask in app_IBM.py and load the new IBM model
End

Fig. 4.7.1: - Overall Architecture Diagram

4.8 Individual Contributions: -

ARYAN OMKAR ASHAR – 19BAI10094: -

Project Description: - Pneumonia is a disease that affects a large number of people around the world. In the recent few years of COVID-19, where the pandemic has brought all our lives to a standstill, pneumonia has been detected as one of the major side-effects of COVID-19. A large population of people who suffered from COVID, ended up being a prey to pneumonia as well. In order to combat this and make the lives of doctors easier, this project has been designed keeping in mind the safety of each and everyone. The project revolves around a smart medical diagnostic system that is successfully able to detect whether the person is suffering from pneumonia or not. The input image is an X-RAY image of the chest and this X-RAY should be a clear scan of the lungs. The project uses techniques of machine learning algorithms and risk analysis and gives an output whether the patient should rush to the doctor or he/she is safe. We have also created a webpage for this purpose. The dataset used was collected individually from google and Kaggle and some of the images are real-time images of patients. There are a total 5216 images on training set, 624 images on testing set and 16 images on validation dataset. We have used various programming languages like Python 3.0, HTML, JavaScript and CSS. Along with basic applications, we have also used IBM Watson Studio. A total of 2 models have been created: one which has been trained locally and has a training accuracy of 93.12%. Another model has been trained by the help of IBM Watson Studio and has achieved a training accuracy of 94.73%. We have also optimized the model by using optimizers and helped the model to have a testing accuracy of 99.99974%. Spyder has been used to perform flask deployment. Both these methodologies of training are unique and helps to increase the accuracy and properly link the python model to the webpage.

Contribution: -

This project has been developed as a team by assistance of all my team members without whom it would have been nearly impossible in order to reach such a level of precision and accuracy. My contribution for this project began with brainstorming of ideas and the topic that can help to bring a change in the community. I keenly observed what were the problems that the community was facing and how I can help to overcome those problems. During such a period only, the idea for this project was born.

Phase 1 Review: - During the review of phase 1, after keenly observing the problems that the community is facing, I laid down the problem statement in front of my team members as well as to my faculty guide. My problem statement was approved by everyone and we started working on the project for pneumonia detection. I contributed in introducing the project as well as the techniques to be used in the project to my team members and wrote about the same in the report. Also, shifting towards the literature review part, I studied many research papers and articles. I contributed in writing the introduction and core area of the project along with the abstract of the report of phase 1 (submitted in November). I also collected some X-RAY images by going to some patients who had suffered from pneumonia in the past and used those images in our dataset. I illustrated the methods to be used in the project and also designed the complete methodology and the block diagram which my teammates referred to in the course of the entire project. The Block Diagram, methodology and the overall architecture diagram designed by me served as the pillars of this project without which it would have been impossible to develop this project. I proposed the system methodology by jotting down its advantages in the report as well as to my teammates who approved it without any sort of hesitation. For the report and documentation part, I wrote down the abstract of the report and the problem statement in the introduction section. In the Literature Review section, I wrote about the introduction, core area of the project, existing algorithms and other methods used in the project. The flowchart for block diagram, methodology and overall architecture was designed by me. In the system analysis part of the report, I proposed the methodology of the system and its advantages as well as laid down the overall architecture to be followed in the project.

Phase 2 Review 1: - In the phase 2 review 1(presented in form of PPT in March), I wrote about the workflow of the model till date. Apart from this in the technical slide, I incorporated a new technology in the system by the help of IBM Watson Studio that helped to raise the accuracy of the system to 99.99974%. I worked on the model that can be used only with the help of internet and is trained on IBM Watson Studio. First of all, I registered all the services on IBM Cloud by creating an IBM Academic Initiative account. The location was set to Dallas in order to use all the services easily. After that, I wrote down the complete code for extraction of dataset, training of the model using CNN, optimization of the model and testing of the model. This model performed better than the locally developed model. After that, I enhanced the security of the model by inserting an API key. Only the user with the API key will be able to download the model. The model was downloaded by coding performed of Jupyter Notebook in Python 3.0. After this, I wrote the code for app_IBM.py part for flask deployment of my model so that it can be linked to the webpage. Also, I illustrated the results of the project and its advantages in the phase 2 review 1. I also filmed the video explaining and demonstrating our whole project.

Phase 2 Review 2: - In the phase 2, review 2 I worked on optimizing our model and helped it gain more accuracy. The accuracy went upto 99.99974 for the local model as well as for the IBM trained model. This task was performed in a team by me as well as my fellow mates: Shreya Shetye (19BAI10028) and Pranav Sharma(19BAI10154). Apart from the technical part, the designing of all the presentations and drafting of all reports was done by me. I served as a sole communicator between the faculty guide, supervisors and my team members.

5. PERFORMANCE ANALYSIS

5.1 INTRODUCTION

We will discuss about the efficiency and the results of our smart medical diagnosis system in this section. Also, the results will be discussed in this section in depth. Our model currently has a very high accuracy when comparing it with our competitive models. Such a level of accuracy can easily receive a green light from the government to be used in medical centers.

5.2 RESULTS

1. The final result of the project is to identify looking at the lung x ray whether the patient has pneumonia or not.
2. We have implemented several models in our project each providing a different set of outputs and combining their results yields the final output.
3. Our model is trained and tested on Jupyter notebook using python which results in training our model using the dataset and classifying them into two classes 0 and 1. 0 denotes pneumonia is not detected and 1 denotes pneumonia is detected.
4. We have created templates for our website using HTML, CSS and JavaScript which functions in Spyder.
5. We have created the website using flask which is a module in python for website creation using Spyder.
6. This gives us the final result which is our website where the user can upload a picture of the x ray and using our model it predicts whether the patient has pneumonia or not.
7. **The accuracy of the model trained locally is 93.12% of training accuracy at the end of 20th epoch. The testing accuracy comes out to be 99.99974%.**
8. **The accuracy of the model trained on IBM Watson Studio is 94.73% and a testing accuracy of 99.99974%.**
9. We observe that when the number of epochs increases, the accuracy of the model also increases. It is a self-learning model.

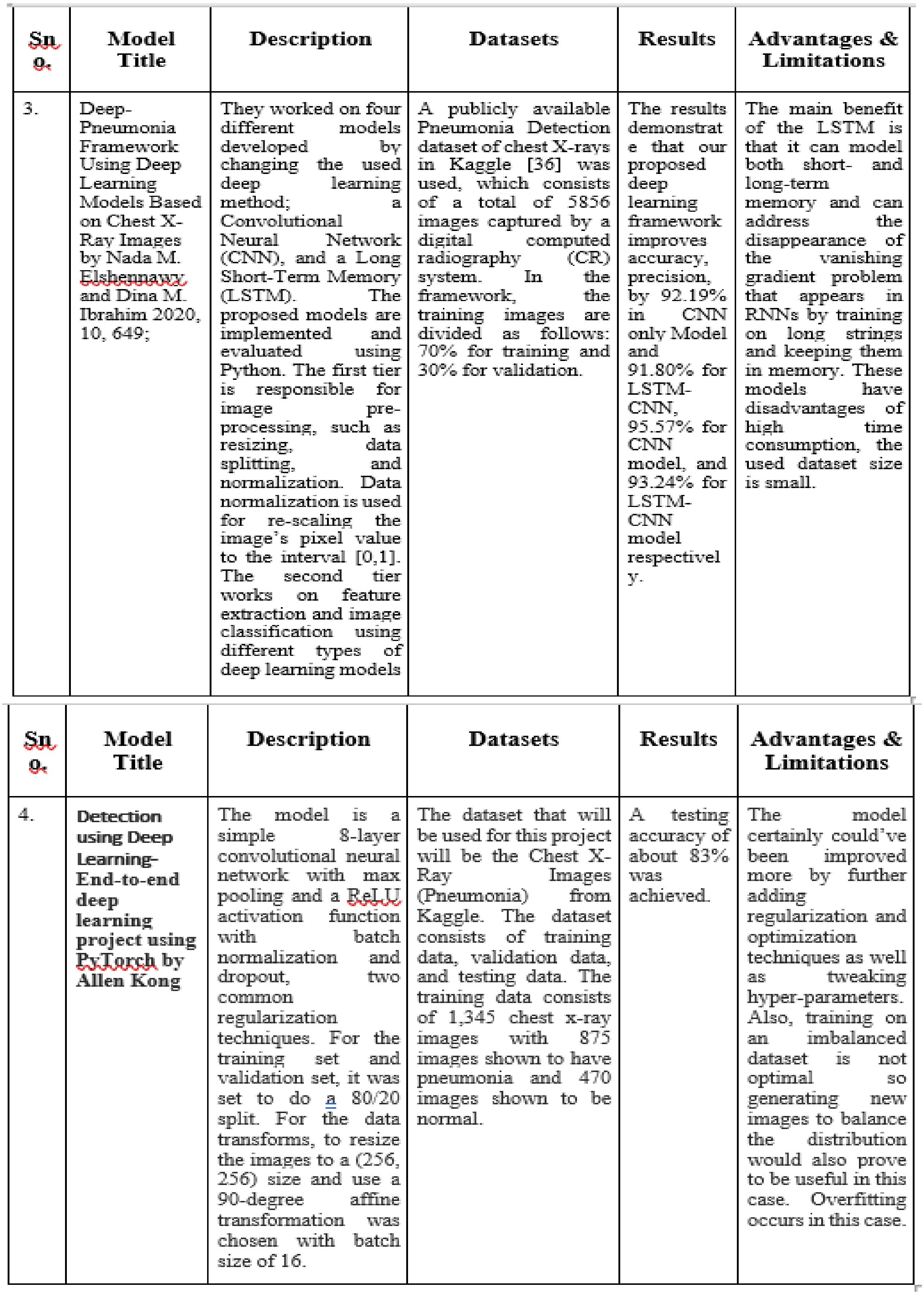
5.3 PERFORMANCE ANALYSIS

There are several measures that affect the working of our model. Firstly, the accuracy of our locally trained model during its training phase comes out to be 93.12%. The testing accuracy of the same locally trained model comes out to be 99.99974%. Apart from the locally trained model, the model trained by the help of IBM Watson Studio has an accuracy of 94.73% during the training phase and 99.99974% during the testing phase. It is easily observable that the IBM trained model performs better. But this model needs internet connectivity in order to run. Whereas, if there is an issue with the internet, the locally trained model can be used. The accuracy keeps increasing as we increase the number of epochs in the model.

5.4 COMPARITIVE STUDY OF THE MODELS

The table given below is used in order to compare our model with the already existing models with different algorithms used. It is clear that our model performs the best among all the methodologies.

Sn o.	Model Title	Description	Datasets	Results	Advantages & Limitations
1.	Our Model	Our proposed solution of using the Convolution Neural Networks (CNN) to analyse the images of the X-ray to predict whether the pneumonia is present or not, It is a sequential type of CNN model which contains Convolution 2D, Max Pooling 2D, Dropout, Flatten and Dense layers.Our model is trained and tested on Jupyter notebook using python which results in training our model using the dataset and classifying them into two classes 0 and 1. 0 denotes pneumonia is not detected and 1 denotes pneumonia is detected. Data Augmentation was achieved through ImageDataGenerator Class. Next we perform training of the model on IBM Watson studio. We observe that the accuracy we obtain there is greater than the accuracy of the locally trained model.	The dataset that we have chosen has a total of 5216 images in the training set with 2 categories, normal lung x-ray and pneumonia lung x-ray. The categories are divided into 234 images of normal lung x-ray and 390 images of pneumonia lung x rays in the test set, 1341 images of normal lung x-ray 3875 images of pneumonia lung x-ray in the train set, and 8 images of normal lung x rays and 8 images of pneumonia lung x rays in the validation set.	The training is complete for 20 epochs, we test the model. The final accuracy of the entire model after all the training is 99.99974% . The images are divided in the 90:10 ratio to maintain the bias-variance trade-off. The accuracy of the model trained on IBM Watson Studio is 94.73%.	The accuracy of the IBM trained as well as the locally trained model is very high and thus it can be used in daily life. Prediction can be done by sitting at home and this allows the doctors to focus on more serious cases. The model is capable of capturing non linearities between predictors and outcome. Yet, the model cannot be trusted completely as of now, that is, in the initial phase as we have limited dataset.
2.	Zhang, D.; Ren, F.; Li, Y.; Na, L.; Ma, Y. Pneumonia Detection from Chest X-ray Images Based on Convolutional Neural Network. Electronics 2021, 10, 1512.	The model's performance in pneumonia detection shows that the proposed VGG-based CNN model could effectively classify normal and abnormal X-rays in practice, hence reducing the burden of radiologists. They develop a straightforward VGG based model architecture with fewer layers. In addition, to tackle the inadequate contrast of chest X-ray images, which brings about ambiguous diagnosis, the Dynamic Histogram The model contains a total of six layers, where we employed 3 × 3 kernel convolution layers.	The proposed dataset used to evaluate model performance consists of 5786 X-ray images and is from the Kaggle competition. The original Pneumonia Classification Dataset in Kaggle is split into three sub-sets, training, validation and test sets randomly at ratios of 70%, 10% and 20%.	The training accuracy and validation accuracy all exceeded 95%, also the training loss and validation loss were below 0.025. The results of the obtained accuracy rate of 96.07% and precision rate of 94.41%.	The method aims to visualize the distribution of high dimensions and assists us in analyzing the connection among the sample. Yet, the enhanced model accuracy of VGG is below the original non-enhanced models. This is possibly caused by the fact that the number of model parameters affects the performance of different models.



Sn o.	Model Title	Description	Datasets	Results	Advantages & Limitations
3.	Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images by Nada M. Elshennawy and Dina M. Ibrahim 2020, 10, 649;	They worked on four different models developed by changing the used deep learning method; a Convolutional Neural Network (CNN), and a Long Short-Term Memory (LSTM). The proposed models are implemented and evaluated using Python. The first tier is responsible for image pre-processing, such as resizing, data splitting, and normalization. Data normalization is used for re-scaling the image's pixel value to the interval [0,1]. The second tier works on feature extraction and image classification using different types of deep learning models	A publicly available Pneumonia Detection dataset of chest X-rays in Kaggle [36] was used, which consists of a total of 5856 images captured by a digital computed radiography (CR) system. In the framework, the training images are divided as follows: 70% for training and 30% for validation.	The results demonstrate that our proposed deep learning framework improves accuracy, precision, by 92.19% in CNN only Model and 91.80% for LSTM-CNN, 95.57% for CNN model, and 93.24% for LSTM-CNN model respectively.	The main benefit of the LSTM is that it can model both short- and long-term memory and can address the disappearance of the vanishing gradient problem that appears in RNNs by training on long strings and keeping them in memory. These models have disadvantages of high time consumption, the used dataset size is small.
4.	**Detection using Deep Learning-End-to-end deep learning project using PyTorch by Allen Kong**	The model is a simple 8-layer convolutional neural network with max pooling and a ReLU activation function with batch normalization and dropout, two common regularization techniques. For the training set and validation set, it was set to do a 80/20 split. For the data transforms, to resize the images to a (256, 256) size and use a 90-degree affine transformation was chosen with batch size of 16.	The dataset that will be used for this project will be the Chest X-Ray Images (Pneumonia) from Kaggle. The dataset consists of training data, validation data, and testing data. The training data consists of 1,345 chest x-ray images with 875 images shown to have pneumonia and 470 images shown to be normal.	A testing accuracy of about 83% was achieved.	The model certainly could've been improved more by further adding regularization and optimization techniques as well as tweaking hyper-parameters. Also, training on an imbalanced dataset is not optimal so generating new images to balance the distribution would also prove to be useful in this case. Overfitting occurs in this case.

Table 5.4.1: - Comparison with existing models

5.5 OPTIMIZATION OF THE MODEL

Initially, the model was only able to achieve an accuracy of 93.12% and 94.73%. After this, we tried several optimization techniques in order to increase the accuracy of the model. But this was not an easy task as the greater number of optimizers that we were trying, the more issues the model was coming up with. Sometimes, the data was becoming prone to overfitting whereas on some times, the accuracy was indeed decreasing. But at the end, we tried to use ADAM optimizing techniques which we recently studied in the subject of convex optimization. As soon as we incorporated this algorithm, we observed that the accuracy increased to 99.99974%.

5.6 CONCLUSION

The overall performance analysis suggests that the above system is ready for implementation in medical centers, diagnostic center as well as in hospitals. This system can also be implemented in home desktops and computers. It can be proved as a feasible solution considering both performance and cost and this involves 0 cost of making and using. The model can and is capable of increasing its performance measure based on the current results. Thus, the model we have developed is a SELF LEARNING MODEL. The model is ready to come in use and make the diagnostic systems into a smart diagnostic system with less human intervention and zero cost factor.

6. FUTURE ENHANCEMENTS

6.1 INTRODUCTION

This smart medical diagnosis system can be of great help in the medical field. The future scope for this model is very wide and involves too many applications as well as improvements. Some of the applications and the further improvements of this model and listed in the below subsections.

6.2 CURRENT APPLICATIONS

- Pneumonia is a respiratory illness that affects a large number of people, particularly in poor and underdeveloped countries where pollution, unsanitary living conditions, and overcrowding are all too widespread, as well as a lack of medical infrastructure. It is critical to diagnose pneumonia early in order to receive curative treatment and boost chances for survival.
- The most common approach for detecting pneumonia is chest X-ray imaging. If your doctor suspects you have pneumonia, a chest X-ray will be taken to determine the extent of the infection in the patient's lungs. However, examining chest X-rays is a difficult process that is vulnerable to subjective variability. In this study, we created a computer-aided diagnosis system for automated pneumonia identification utilising chest X-ray pictures.
- In order to diagnose Pneumonia disease, it is necessary to be able to recognise infections in the lungs. The evaluation of the chest with X-ray imaging is normally done by experienced human examiners or doctors, which takes time and is difficult to standardise. We computed a Pneumonia detection model using the Deep Convolutional Neural Network with Pneumonia Chest X-ray dataset. The collected data was gathered from the various patients after clinically examination and categorization by clinicians and doctors.
- We devise a method for detecting pneumonia in frontal-view chest X-ray pictures that outperforms professional radiologists.
- In locations where there is a dearth of experienced personnel in procedures like radiography, disease detection with the help of computers using different Machine and Deep learning techniques is quite advantageous. Particularly in South Asian and African countries, where 60 to 70 percent of the population lives in rural areas.
- We anticipate that by automating at the expert level, this technology can enhance healthcare delivery and provide access to medical imaging knowledge in areas where qualified radiologists are scarce. Such tools are inexpensive, and instrumentation is widespread; as a result, they are simple to deploy in rural regions. Furthermore, these technologies will be extremely useful in automatically distinguishing between those who require immediate medical attention and those who can wait.
- The analysis of X-RAYS using CNN is right now being utilized at different clinical establishments including Singapore’s Changi General hospital (desktop application i.e., installed on radiology workstations) etc...
- The application of CNN in Pneumonia detection using X-RAYS assists us with using different libraries present to help identify the severity of pneumonia that correlates to the degree of Chest X-RAY (CXR) lung image abnormality.
- Many people have now started keeping portable X-RAY machines at their homes. This system can be installed else the website can be shared where they will be able to detect whether they are infected with pneumonia or not.

- The webpage that has been created can be shared anywhere easily and can be used at any corner of the world. It does not matter whether you are connected to the internet or not, you can still use our model.
- This model helps in quick diagnosis during the pandemic as well as after the pandemic as pneumonia is a disease that is going on from a long time.

6.3 FUTURE ENHANCEMENTS

Pneumonia is an important cause of morbidity and mortality; hence this is an impressive collection of support for adult emergency clinics, with countless people in these eventually dying. According to WHO, pneumonia can be prevented with simple procedures and early diagnosis and treatment. Nevertheless, the majority of the world's population requires radioactivity approval. With the availability of imaging hardware, there is a shortage of specialists who can analyze X-ray images anyway. There is no doubt that you can edit the predictive model much better by performing data expansion or by performing transfer learning concepts to manipulate and improve the model. With that in mind, this will be added to future research and projects as an additional upgrade. In the future, it will be exciting to see an approach that can better estimate weights compared to different models, that take patient experience and family history into account when making predictions. This is going to give the model an upper hand over the rest, as the accuracy of what the disease exactly might be, will eventually be greater.

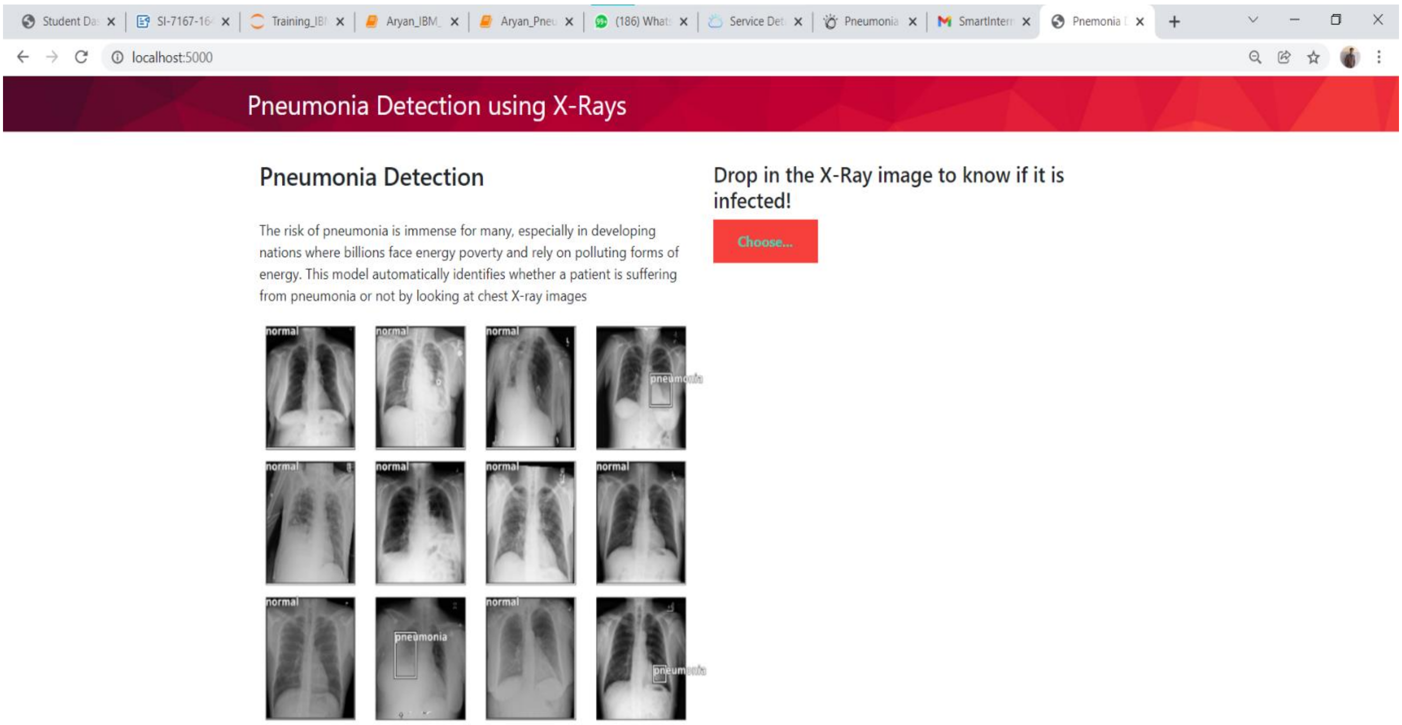
This model of pneumonia detection can be used in wide variety of applications. It can be installed in hospitals so that it helps the doctors save time. Ahead what can be inculcated in this model, is that it can be tried to install in an X-RAY machine. As soon as the results if the X-RAY are out, the person can come to know and thus in this way, he/she can start the medication as soon as possible. It is no doubt that the predictive model can be worked on far superior by performing data augmentation or carrying out transfer learning concept which works with the model an opportunity to get better. Along these lines, this will be added as additional upgrade in the forthcoming stories. In the future, it is intriguing to see approaches in which the weights relating to various models can be assessed all the more proficiently and a model that considers the patient's set of experiences while making predictions. In future, this complete system can also be brought on a mobile phone where we can detect by just clicking a picture of the X-RAY using a cellphone.

6.4 CONCLUSIONS

Pneumonia is responsible for a large percentage of patient comorbidity and fatality. Early diagnosis and treatments of pneumonia are critical to preventing complications such as death. X-rays are the most common imaging examination tool used in prognosis, diagnosis, and management of a variety of diseases, including pneumonia. We develop an algorithm that detects pneumonia from frontal-view chest X-ray images at a level exceeding practicing radiologist. We also show that a simple extension of our algorithm to detect multiple diseases outperforms the previous state of the art on ChestX-ray14, the largest publicly available chest X-ray dataset. With automation at the level of expertise, we hope that this technology can enhance healthcare delivery and unlock greater capabilities to medical imaging expertise in parts of the world where access to skilled radiologists is limited. The web app that is made by us, will be acting as a huge platform for all the users globally to be self-reliant to check out for pneumonia in the chest x-rays that they have received. It is programmed to only aid the radiologists in the decision-making process; the final decision has to be made by an expert himself/herself.

This work shows that systematic end-to-end study design and execution requires a multidisciplinary team effort involving healthcare professionals, machine learning researchers, medical imaging specialists, software developers, and statisticians. Unsolved problem that is as of now tormenting the doctors around the world. Our proposed model is designed and advancement to detect and classify pneumonia from chest X-ray pictures. It contains image processing technique as well as convolutional neural network. We developed a model; the algorithm starts by transforming chest X-ray images into sizes less than the original. The next step includes the identification and classification of pictures by the convectional neural network structure, in order to extracts features from the image and classify accordingly. This work has presented the X-Ray images for Pneumonia discovery based on convolutional neural networks and diverse machine learning. By training a bunch of solid CNNs for an enormous scope dataset, we built a model that can precisely predict Pneumonia. During each epoch data is trained again and again to learn the feature of data. The presentation assessment of the model is estimated by utilizing classification accuracy and cross-validation. We obtain an accuracy of 94.73% when trained on IBM and a testing accuracy of 99.99974%.

7. SOME OUTPUT IMAGES



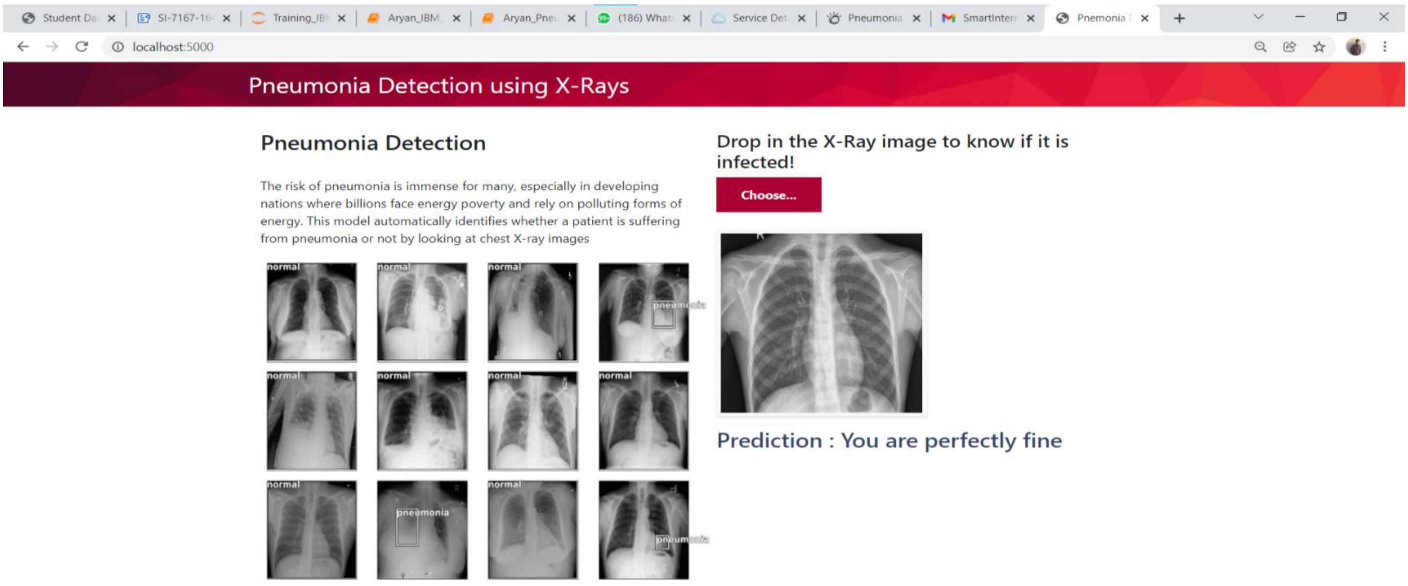
Pneumonia Detection using X-Rays

Pneumonia Detection

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. This model automatically identifies whether a patient is suffering from pneumonia or not by looking at chest X-ray images

Drop in the X-Ray image to know if it is infected!

Fig. 7.1: - Home Page



Pneumonia Detection using X-Rays

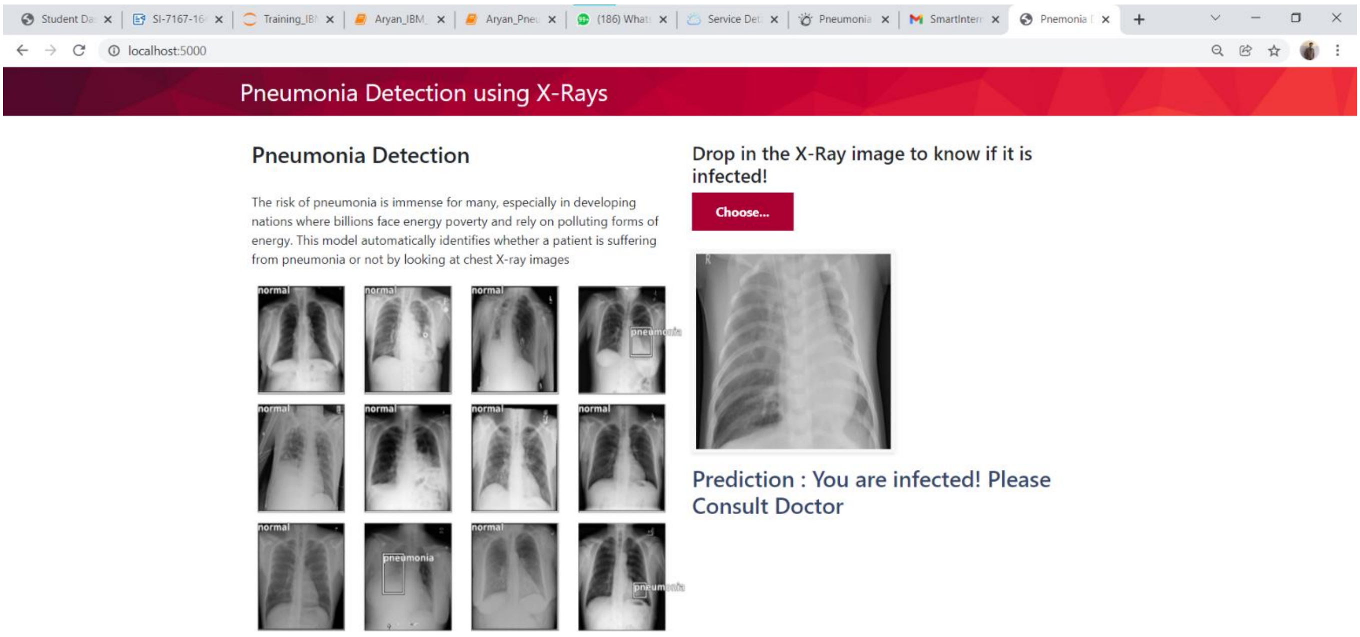
Pneumonia Detection

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. This model automatically identifies whether a patient is suffering from pneumonia or not by looking at chest X-ray images

Drop in the X-Ray image to know if it is infected!

Prediction : You are perfectly fine

Fig. 7.2: - Person is completely fine



Pneumonia Detection using X-Rays

Pneumonia Detection

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. This model automatically identifies whether a patient is suffering from pneumonia or not by looking at chest X-ray images

Drop in the X-Ray image to know if it is infected!

Choose...

Prediction : You are infected! Please Consult Doctor

Fig. 7.3: - Patient is infected with pneumonia

8. APPENDIX A: - ABOUT IBM WATSON STUDIO

IBM Watson Studio is a service that is available on IBM Cloud. It is a paid service but is available for students without any cost. The student needs to have an academic initiative account created in IBM and from this he/she will be able to access this service. IBM Watson Studio is used by data scientists and artificial intelligence engineers as well as many researchers for their tasks on AI/ML and deep learning. IBM Watson Studio helps to take optimized decisions on the dataset and helps to remove the outliers present in the dataset and thus it helps to improve the accuracy of the model. The model that is developed in IBM Watson Studio can be deployed as well as tested there. The IBM Watson Studio uses python interface in order to train, test and develop the model and involves advanced python coding in order to develop the model. IBM Watson Studio can be used to implement explainable AI. It is also used to develop and design model and thus can be a replacement for jupyter notebook. The models on IBM Watson Studio can be downloaded only by the help of a unique API key which aids in enhancing the security and safety of the model and helps the user control the passing of model from system to system.

9. REFERENCES

- Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R. and Mittal, A., 2019, February. Pneumonia detection using CNN based feature extraction. In 2019 IEEE international conference on electrical, computer and communication technologies (ICECCT) (pp. 1-7). IEEE.
- Toğaçar, M., Ergen, B., Cömert, Z. and Özyurt, F., 2020. A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models. Irbm, 41(4), pp.212-222.
- Hashmi, M.F., Katiyar, S., Keskar, A.G., Bokde, N.D. and Geem, Z.W., 2020. Efficient pneumonia detection in chest xray images using deep transfer learning. Diagnostics, 10(6), p.417.
- D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan and A. Mittal, "Pneumonia Detection Using CNN based Feature Extraction," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-7, doi: 10.1109/ICECCT.2019.8869364
- Zhang, D.; Ren, F.; Li, Y.; Na, L.; Ma, Y. Pneumonia Detection from Chest X-ray Images Based on Convolutional Neural Network. Electronics 2021, 10, 1512.
- Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images by Nada M. Elshennawy and Dina M. Ibrahim 2020, 10, 649;
- Faris, H., Habib, M., Faris, M., Elayan, H., & Alomari, A. (2021). An intelligent multimodal medical diagnosis system based on patients' medical questions and structured symptoms for telemedicine. *Informatics in Medicine Unlocked*, *23*, 100513.
- Gabruseva, Tatiana, Dmytro Poplavskiy, and Alexandr Kalinin. "Deep learning for automatic pneumonia detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.
- Pawshe, S., Prasad, N., Phondekar, V., & Shintre, R. (2020). Detecting Pneumonia from Chest X-Ray Images using Committee Machine. vol, 7, 6.
- Stephen, Okeke, et al. "An efficient deep learning approach to pneumonia classification in healthcare." Journal of healthcare engineering 2019 (2019).
- Chouhan, V., Singh, S.K., Khamparia, A., Gupta, D., Tiwari, P., Moreira, C., Damaševičius, R. and De Albuquerque, V.H.C., 2020. A novel transfer learning based approach for pneumonia detection in chest X-ray images. Applied Sciences, 10(2), p.559.
- Rahman, T., Chowdhury, M.E., Khandakar, A., Islam, K.R., Islam, K.F., Mahbub, Z.B., Kadir, M.A. and Kashem, S., 2020. Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. Applied Sciences, 10(9), p.3233.

Dr. Kanchan Lata Kashyap
Project Supervisor