

TO DETECT PNEUMONIA USING CONVOLUTIONAL NEURAL NETWORKS - RECOMMENDER SYSTEM

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Information Technology
By

ARAVA JAYARAM (39120045)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI - 600119

MAY-2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

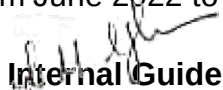
www.sathyabama.ac.in



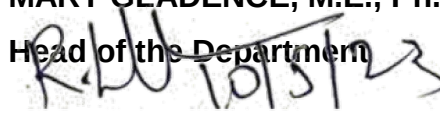
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ARAVA JAYARAM (Reg.No:39120045)** who carried out the Project Phase-2 entitled “**TO DETECT PNEUMONIA USING CONVOLUTIONAL NEURAL NETWORKS - RECOMMENDER SYSTEM**” under my supervision from June 2022 to November 2022.


Internal Guide

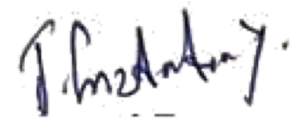
Dr. L. MARY GLADENCE, M.E., Ph.D.


Head of the Department

Dr. R.SUBHASHINI, M.E., Ph.D.

Submitted for Viva voce Examination held on 10-05-2023


Internal Examiner


External Examiner

DECLARATION

I, **ARAVA JAYARAM (39120045)** hereby declare that the Project Phase-2 Report entitled “**TO DETECT PNEUMONIA USING CONVOLUTIONAL NEURAL NETWORKS - RECOMMENDER SYSTEM**” done by me under the guidance of **Dr. L. MARY GLADENCE, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **INFORMATION TECHNOLOGY**.

DATE: 10-05-2023

PLACE: Chennai

A. Jayaram

Arava Jayaram

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. R.SUBHASHINI, M.E., Ph.D.** Head of the Department of INFORMATION TECHNOLOGY for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. MARY GLADENCE, M.E., Ph.D.** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my phase-2 project work.

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

ABSTRACT

Pneumonia is a potentially lethal respiratory illness that is the leading cause deaths in India, according to the World Health organisation. To diagnose pneumonia, chest X-rays must be viewed by qualified radiologists, which might be difficult in remote areas. As a result, developing an automated system for identifying pneumonia can considerably aid in the rapid and effective treatment of the illness Convolutional neural networks had proven to be particularly effects for health images, and their ability to extract features from big datasets makes them suitable for image classification tasks. We investigated the effectiveness of models trained using CNN as information extractors, trailed by different classifiers, in distinguishing b/w abnormal as well as normal X-rays of the chest in this work. We identified which CNN model would best serve this function by analytical investigation. According to our statistical findings, supervised classifier algorithms combined with pre-trained CNN models can be quite useful when assessing chest X-ray images, particularly for detecting pneumonia. After establishing the presence of the condition, the system can recommend the nearest pneumonia-specific treatment facilities. The use of pre-trained CNN models in an automated system for pneumonia detection can significantly aid in the accurate and timely diagnosis of the condition. Especially in remote areas. This could improve medical care and potentially save lives.

TABLE OF CONTENTS

Chapter no	TITLE	Page no
	ABSTRACT	ii
	LIST OF FIGURES	v
	LIST OF TABLES	vi
1	INTRODUCTION	1
	1.1 Motivation	3
2	LITERATURE SURVEY	4
	2.1 Inferences from literature survey	6
	2.2 Open problems in existing system	7
	2.3 Existing systems	7
3	REQUIREMENTS ANALYSIS	8
	3.1 Feasibility Studies/Risk Analysis of the Project	9
	3.2 Software Requirements Specification Document	11
	3.3 System Use case	12
4	DESCRIPTION OF PROPOSED SYSTEM	14
	4.1 Selected Methodology or process model	15
	4.2 Architecture / Overall Design of Proposed System	16
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	17
	4.4 Project Management Plan	19

4.5	Transition/ Software to Operations Plan	20
5	IMPLEMENTATION DETAILS	22
5.1	Development and Deployment Setup	22
5.2	Algorithms	23
5.3	Testing	23
6	RESULTS AND DISCUSSION	25
7	CONCLUSION	29
7.1	Conclusion	29
7.2	Future work	30
7.3	Research Issues	30
7.4	Implementation Issues	31
	REFERENCES	33
	APPENDIX	34
	A. SOURCE CODE	34
	B. SCREENSHOTS	56
	C. RESEARCH PAPER	60

LIST OF FIGURES

Figure No:	Figure Name:	Page No:
1.1	AN OVERALL TAXONOMY OF DEEP LEARNING	3
4.1	SYSTEM ARCHITECTURE	17
6.1	HOME PAGE	25
6.2	MODEL PAGE	26
6.3	XRAY UPLOADER PAGE	26
6.4	DETECTION PAGE	27
6.5	NEAREST HOSPITAL FINDER	27

List of Tables

Table No:	Table Name:	Page No:
6.1	Results of the proposed system	25

CHAPTER 1

INTRODUCTION

Computer vision is a relevant application field of artificial intelligence (AI), which comes as no surprise, mainly because it offers solutions to a vast amount of problems that we humans face in the 21st century. One of the fields of computer vision that has proved to be fruitful again and again is medical image diagnosis with the aid of AI.

Pneumonia represents a highly ranked cause of death with over 4 million fatalities yearl . Pneumonia has many variants, it can be viral, bacterial, or fungal, and each of these variants may have various classes. Pneumonia is often diagnosed via chest X-ray (CXR) or computed tomography (CT) scans, the former being the most commonly used technique due to its reduced costs and availability in all countries. The computer can help the human expert in establishing the diagnosis of pneumonia due to its high speed and objective reproducible judgement, as the signs of pneumonia in X-ray images are not always visible or legible for the human eye. Several computer aided solutions have been provided for the detection and categorization of pneumonia.

The COVID-19 pandemic caused a new wave in the development of such diagnosis procedures. In the following, we are presenting some of the recent successful solutions, mainly based on convolutional neural networks (CNN) and deep learning. Jain et al. deployed VGG-16, VGG-19, Resnet50, and Inception-V3 network architectures and transfer learning for the detection of viral pneumonia, obtaining 71–88% accuracy on 624 test images. Dey et al. combined the VGG-19 architecture with various classifiers like CNN or random forest, and based on a set of 1650 X-ray images they achieved accuracy, precision, and F1 score values ranging up to 98%, 95%, and 96%, respectively. Brunese et al. established a three-class pneumonia detection framework based on VGG-16 architecture and the so-called GradCAM algorithm for visual debugging, and achieved a 96.2% accuracy on a set of 6523 CXR images. Panwar et al. combined the VGG-19 architecture with the GradCAM algorithm, and achieved 95.6% accuracy in a three-class pneumonia classification framework. Similarly, Ibrahim et al. proposed a solution for three-class pneumonia detection, using VGG and ResNet152V2 architectures, and obtained

recall values of 93–97% based on an own collection of X-ray images. Jin et al. proposed a three-stage hybrid model consisting of feature extractor, feature selector, and CNN classifier stage, which achieved a remarkable 98.62% accuracy on a private set of 1743 X-ray images. Karthik et al. employed the so-called Channel-Shuffled Dual-Branched (CSDB) CNN architecture to distinguish various types of pneumonia from several publicly available datasets, achieving F1 scores of 94–98%. Quan et al. combined the DenseNet and the CapsNet architecture and achieved 96% recall on a small sized COVID-19 dataset, but only 90.7% accuracy on larger set of pneumonia X-ray images. Alhudhaif et al. deployed a DenseNet-201 based architecture and achieved accuracy and recall values up to 95% and almost 90% precision on Kaggle’s dataset of over 6000 X-ray images. Sirazitdinov et al. used the 26,684 CXR images of the Kaggle Pneumonia Detection Challenge to train an ensemble of two CNNs, namely a RetinaNet and a Mask R-CNN, but they achieved only a recall value of 80%.

The Mask R-CNN architecture was also deployed by Jaiswal et al. to localize regions of the lung affected by pneumonia. Mahmud et al. proposed a so-called CovXNet architecture, a CNN that uses depthwise convolution with varying dilation rates to extract a wide range of features from X-ray images, which was trained and validated on 6161 CXR scans, achieving accuracy of 90.2% in a four-class and 97.4% in a two-class pneumonia classification problem. Ouchicha et al. proposed a residual CNN architecture and trained it on 2905 X-ray images, reaching a 96.7% overall accuracy in three-class pneumonia classification. Wang et al. designed a so-called prior-attention residual learning block and combined it with two 3D-ResNets and obtained 93.3% accuracy in the three-class pneumonia detection problem, based on 4697 X-ray images. Probably the largest available CXR image dataset, the one published by RSNA Pneumonia Detection Challenge consisting of over 120,000 items, was used by Wang et al. to train a VGG architecture. They obtained an accuracy of 94.62% in the detection of pneumonia. Wang et al. deployed a so-called deep stacked sparse autoencoder network model to provide a four-class classification of breast CT images based on two-dimensional fractional Fourier entropy features, and obtained a 92.3% averaged F1 score. Wang et al. deployed a deep rank-based average pooling network that combined an n -conv rank-based average pooling module with a CNN model inspired by the VGG network, reaching

an average F1 score of 95.5% in a study elaborated on 1164 CT scans of 521 subjects. Horry et al. performed a comparative study of imaging modalities within the pneumonia detection framework and found that pneumonia can be best detected from ultrasound scans. Rajaraman et al. presented a framework in which various CNN models deployed through transfer learning were involved in ensemble learning. Their ensembles were trained using 16,700 CXR images originating from 4 public databases, and achieved various accuracy rates between 94% and 99%. Singh and Yow proposed an interpretable deep learning neural network approach based on two existing models, ProtoPNet and NP-ProtoPNet, and achieved an accuracy of 87.27%. For further details of the current state-of-the-art, the reader is referred to recent review papers

1.1 MOTIVATION

Detection of Pneumonia from images is a vital task relevant to the area of both Computer Vision. The main motivation behind this research was to identify Pneumonia just by using the X-Ray images of the patients. As doctors must do a lot of certain tests to identify if the patient has Pneumonia or not

To solve the cumbersome problem, an ensemble of two deep learning models is developed, to make the work of the doctors simpler.

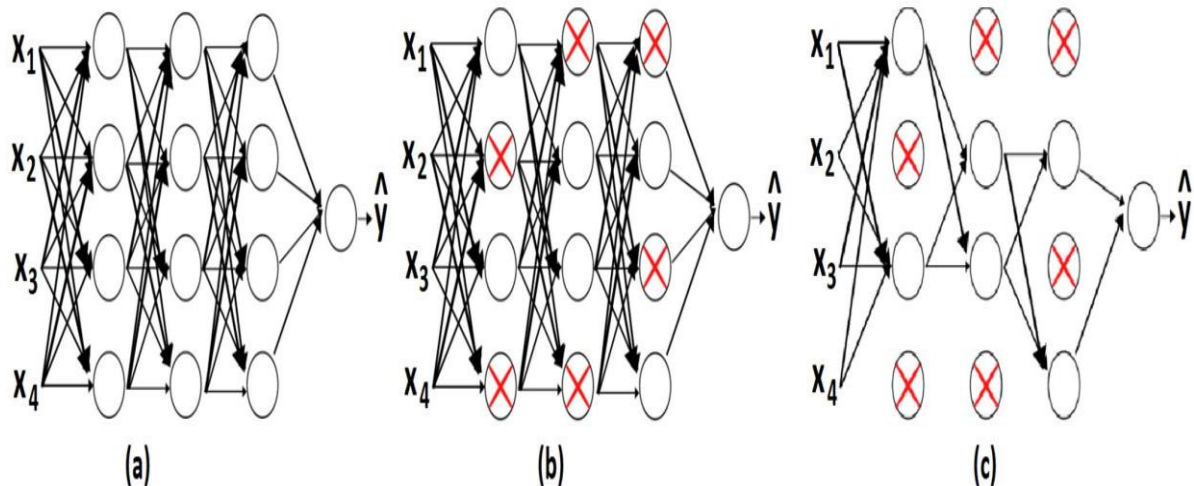


FIG 1.1 : Taxonomy of Deep Learning

CHAPTER 2

LITERATURE SURVEY

TO DETECT PNEUMONIA USING DEEP LEARNING TECHNIQUE:

Abstract—In Artificial Intelligence (AI), the contents of an image are generated automatically which involves computer vision and NLP (Natural Language Processing). The neural model, which is regenerative, is created. It depends on computer vision and Deep translation. This model is used to generate natural sentences which eventually describes the image. This model consists of Convolutional Neural Network (CNN) as well as Recurrent Neural Network (RNN). The CNN is used for feature extraction from image and RNN is used for sentence generation. The model is trained in such a way that if input image is given to model it generates captions which nearly describes the image. The accuracy of model and smoothness or command of language model learns from image descriptions is tested on different datasets. These experiments show that model is frequently giving accurate descriptions for an input image.

A PNEUMONIA DETECTION USING COMPUTER VISION AND PATTERN RECOGNITION:

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision In this paper, we present a generative model based on a deep recurrent architecture that combines recent advances in computer vision and Deep translation and that can be used to generate natural sentences describing an image. The model is trained to maximize the likelihood of the target description sentence given the training image. Experiments on several datasets show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Our model is often quite accurate, which we verify both qualitatively and quantitatively. For instance, while the current state-of-the-art BLEU-1 score (the higher the better) on the Pascal dataset is 25, our approach yields 59, to be compared to human performance around 69. We also show BLEU-1 score improvements on Flickr30k, from 56 to 66, and on SBU, from

19 to 28. Lastly, on the newly released COCO dataset, we achieve a BLEU-4 of 27.7,

INTEGRATED RECOGNITION, LOCALIZATION AND DETECTION USING CONVOLUTIONAL NETWORKS

We present an integrated framework for using Convolutional Networks for classification, localization, and detection. We show how a multiscale and sliding window approach can be efficiently implemented within a ConvNet. We also introduce a novel deep learning approach to localization by learning to predict object boundaries. Bounding boxes are then accumulated rather than suppressed in order to increase detection confidence. We show that different tasks can be learned simultaneously using a single shared network. This integrated framework is the winner of the localization task of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013) and obtained very competitive results for the detection and classifications tasks. In post-competition work, we establish a new state of the art for the detection task. Finally, we release a feature extractor from our best model.

EVERY PICTURE PREDICTS WHETHER THERE IS PNEUMONIA OR NOT

Humans can prepare concise descriptions of pictures, focusing on what they find important. We demonstrate that automatic methods can do so too. We describe a system that can compute a score linking an image to a sentence. This score can be used to attach a descriptive sentence to a given image, or to obtain images that illustrate a given sentence. The score is obtained by comparing an estimate of meaning obtained from the image to one obtained from the sentence. Each estimate of meaning comes from a discriminative procedure that is learned using data. We evaluate on a novel dataset consisting of human-annotated images. While our underlying estimate of meaning is impoverished, it is sufficient to produce very good quantitative results, evaluated with a novel score that can account for synecdoche.

ATTENTION MECHANISM-BASED CNN FOR X-RAYS GENERATED: Pneumonia detection is a hot research topic and can be applied in many computer vision fields, such as human-computer interaction, affective computing and so on. In this paper, we propose a novel end-to-end network with attention mechanism for automatic pneumonia recognition. The new network architecture consists of four parts, i.e., the feature extraction module, the attention module, the reconstruction module, and the classification module. The LBP features extract image information and then catch the small movements, which can improve the network performance. Attention mechanism can make the neural network pay more attention to useful features. We combine LBP features and attention mechanism to enhance the attention model to obtain better results. Dataset consists of 5216 training images, 624 testing images and 16 validation images. The experimental results demonstrate the feasibility and effectiveness of the proposed method.

DETECTION OF PNEUMONIA USING MULTI-DEEP CONVOLUTIONAL NEURAL NETWORK :

Although there have been many breakthroughs in the use of convolutional neural networks (CNN) for image classification, To detect the diseases in real-life is still a challenge in this research area. This paper proposes a method to leverage state-of-the-art multi-deep CNN encoders to classify facial expression. We conducted experiments to show that combining features from multi-deep CNN is better than using a single deep CNN model. As well as combining multiple CNN models, we show that using rules to remove noise images from the training dataset improves the performance of the system.

The scope of this detecting a disease from image is a challenging problem in the deep learning domain.

2.1 INFERENCES FROM LITERATURE SURVEY

Detections has recently gathered a lot of attention specifically in the deep learning domain. There is a pressing need for context based natural language description of images, however, this may seem a bit farfetched but recent developments in fields like neural networks, computer vision and AP has paved a way for accurately predicting images i.e. representing their visually grounded meaning. We are

leveraging state-of-the-art techniques like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and appropriate datasets of images and their human perceived description to achieve the same. We demonstrate that our alignment model produces results in retrieval experiments on datasetsgenerator.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

- The main aim of this project is to get a little bit of knowledge of deep learning techniques. We use a technique mainly CNN to detect pneumonia.
- So, to make our detection of pneumonia model, we will be merging these architectures. It is also called a CNN-RNN model.
- CNN is used for extracting features from the image. We will use the pre-trained model Xception.

2.3 EXISTING SYSTEM

- The entire flow of the model was explained from data set collection to predicting the disease from the image
- As we have seen in the literature survey there are many drawbacks of the existing model.

CHAPTER 3

REQUIREMENT ANALYSIS

Requirement analysis is a critical step in developing any software project, including a Pneumonia detection application using Deep Learning. The following are the requirements for developing a Pneumonia detection application:

1. **Data collection:** The first requirement is to collect data on patients that can be used to develop the Pneumonia detection model. This data can include demographic data, medical history, vital signs, laboratory test results, and imaging data.
2. **Data preprocessing:** Once the data is collected, it needs to be preprocessed to remove any outliers, missing values, and irrelevant features. This process involves data cleaning, data normalization, and feature selection.
3. **Deep Learning algorithms:** The selection of appropriate Deep Learning algorithms is crucial in developing an accurate Pneumonia detection model. There are several Deep Learning algorithms available such as logistic regression, decision trees, random forests, and support vector Deeps, Convolutional Neural Networks.
4. **Model training and validation:** The Pneumonia detection model needs to be trained on the preprocessed data and validated to assess its accuracy, sensitivity, specificity, and other performance metrics. This process involves splitting the data into training and testing datasets, training the model on the training dataset, and evaluating its performance on the testing dataset.
5. **User interface:** The Pneumonia detection application should have a user-friendly interface that allows clinicians to input patient data and obtain Detections in real-time. The interface should also display the performance metrics of the model to evaluate its accuracy.
6. **Explainability:** The Pneumonia detection model should be explainable to clinicians to understand the reasoning behind the Detections. This can be achieved by using explainable AI methods such as decision trees, rule-based models, or feature importance ranking.

7. Integration with clinical workflows: The Pneumonia detection application should be integrated with existing clinical workflows to enable its adoption in clinical practice. This involves integrating the application with electronic health records, clinical decision support systems, and other clinical systems.

8. Security and privacy: The Pneumonia detection application should be designed with security and privacy in mind to protect patient data. This involves implementing appropriate security measures such as data encryption, access control, and auditing.

9. Regulatory compliance: The Pneumonia detection application should comply with regulatory standards such as HIPAA, GDPR, and other data protection regulations. By fulfilling these requirements, a Pneumonia detection application using Deep Learning can be developed that is accurate, user-friendly, and compliant with regulatory standards.

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

- The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.
- The feasibility study investigates the problem, and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost, and benefits of such a solution, and the feasibility of such a solution.
- The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research

and development of the system is limited. The expenditures must justify. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

Risk Analysis:

1. **Data Quality Risk:** The accuracy and reliability of the Pneumonia detection model depend on the quality of the data used to train and validate the model. If the data is incomplete, inaccurate, or biased, it may result in a less accurate Pneumonia detection model.
2. **Model Interpretability Risk:** The Pneumonia detection model should be explainable to clinicians to understand the reasoning behind the Detections. If the model is too complex or opaque, it may be challenging for clinicians to interpret the results.
3. **Security and Privacy Risk:** The Pneumonia detection application should be designed with security and privacy in mind to protect patient data. If the application is not adequately secured, it may result in data breaches and privacy violations.
4. **Regulatory Compliance Risk:** The Pneumonia detection application should comply with regulatory standards such as HIPAA, GDPR, and other data protection

regulations. If the application does not comply with regulatory standards, it may result in legal and financial penalties.

5. User Acceptance Risk: The success of the Pneumonia detection application depends on user acceptance by clinicians. If clinicians do not trust the Detections or find the user interface challenging to use, they may not adopt the application. By identifying and mitigating these risks, the Pneumonia detection application can be developed successfully, with high accuracy and user acceptance, while also being compliant with regulatory standards and ensuring data security and privacy.

3.2 Software Requirements Specification Document

A software Requirements Specification (SRS) document is a detailed description of the requirements for the software system to be developed. The following is the SRS document for the Pneumonia detection application using Deep Learning.

Introduction The Pneumonia detection application is a software system designed to predict the likelihood of a patient having a Pneumonia . The application uses Deep Learning algorithms to analyze patient data such as age, sex, blood pressure, cholesterol levels, and other health-related information to generate a Pneumonia detection score. The application will be used by healthcare providers to improve patient outcomes and reduce healthcare costs. **Functional Requirements** The Pneumonia detection application will have the following functional requirements:

User Interface The application will have a user-friendly interface that enables healthcare providers to input patient data and view the Pneumonia detection score. The user interface should be intuitive and easy to navigate.

Data Collection and Storage The application will collect patient data such as age, sex, blood pressure, and other health-related information. The data will be securely stored in a database and used to train and validate the Pneumonia detection model.

Pneumonia detection Model The application will use Deep Learning algorithms to develop a Pneumonia detection model. The model will be trained and validated using patient data to ensure its accuracy.

Pneumonia detection Score The application will generate a Pneumonia detection score based on the patient data and the Pneumonia detection model. The score will be displayed to the healthcare provider to help them in making clinical decisions.

Integration with Clinical Workflows The Pneumonia detection application should be integrated with existing

clinical workflows to enable its adoption in clinical practice. The integration should be seamless and not disrupt the clinical workflows.

Non-Functional Requirements

The Pneumonia detection application will have the following non-functional requirements:

- Performance** The application should be able to handle large volumes of patient data and generate Pneumonia detection scores quickly. The response time of the application should be within acceptable limits.
- Scalability** The application should be scalable to handle future growth in the number of patients and healthcare providers using the application.
- Security and Privacy** The application should be designed with security and privacy in mind to protect patient data. The data should be encrypted during transmission and storage, and only authorized healthcare providers should have access to the data.
- Usability** The Pneumonia detection application should be easy to use and navigate, even for non-technical healthcare providers. The user interface should be intuitive, and the application should provide clear and concise information.

System Requirements

The Pneumonia detection application will have the following system requirements:

- Hardware Requirements** High-end computing resources such as servers, workstations, and storage systems. Network infrastructure to support data transfer between the application and the database.

Software Requirements Operating system: Linux or Windows.

Programming language: Python.

Deep Learning libraries: sci-kit-learn, TensorFlow, Keras, and Pandas.

Web framework: Flask or Django.

3.3 System Use Case

A system use case describes the functionality of a system and how it interacts with its users. In the context of our Pneumonia detection project, the main use case would be as follows:

Use Case: Predict Pneumonia

Primary Actor: User
Preconditions: The user must have access to the Pneumonia detection application.
Steps:

1. The user opens the Pneumonia detection application.
2. The application displays the input form to the user.

3. The user enters the required inputs such as age, gender, blood pressure, etc.
4. The user submits the form.
5. The application processes the input and predicts the risk of a Pneumonia .
6. The application displays the result to the user. Postconditions: The user receives the Detection of the risk of a Pneumonia . Alternate Flow:
 - If any of the input fields are missing or invalid, the application displays an error message and prompts the user to correct the input.
 - If the application is unable to predict the risk of Pneumonia due to insufficient or unreliable data, the application displays a message stating that the Detection cannot be made.
 - In summary, the system use case describes the flow of the Pneumonia detection application, from the user entering the input data to receiving the Detection result.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

- As we have observed that using traditional CNN model there is vanishing gradient problem which hinders the Recurrent Neural Network to learn and get efficiently trained.
- So, in order to reduce this gradient descent problem, In this paper we are proposing this model so as to increase the efficiency of predication of pneumonia from the image and also to increase the accuracy of Detections
- This recognition can be done by using Deep Learning Models. With the advancement of deep learning and usage of API'S now it has become easy to detect the pneumona.

The system will be developed using Python programming language and will utilize Deep Learning libraries such as sci-kit-learn and TensorFlow. The dataset used for training the Deep Learning model will be obtained from reliable sources such as medical research studies and publicly available data repositories. The data will be preprocessed and cleaned to ensure accuracy and consistency. The proposed system will have the following features:

1. Input form: The user will be required to input relevant medical data such as age, gender, blood pressure and blood sugar level.
2. Detection result: The system will predict the probability of a Pneumonia occurring in the future based on the input data.
3. Risk assessment: The system will categorize the predicted risk into low, medium, or high-risk categories.
4. Data visualization: The system will display the Detection result using graphical charts and diagrams for better interpretation.

5. User feedback: The system will provide an option for the user to provide feedback on the Detection result, which will be used to improve the accuracy of the system.

6. User authentication: The system will require the user to log in before accessing the Pneumonia Detection application, ensuring data privacy and security. The proposed system will provide an efficient and reliable way to predict the risk of Pneumonia, allowing individuals to take proactive measures to prevent pneumonia disease. The system can also be used by healthcare professionals to assess the risk of a Pneumonia in their patients and provide appropriate medical interventions.

4.2 SELECTED METHODOLOGY OR PROCESS MODEL :

The selected methodology for the development of the pneumonia Detection application is the Agile methodology. The Agile methodology is a flexible and iterative approach to software development that emphasizes collaboration, adaptability, and continuous improvement. The Agile methodology is particularly suitable for this project because it allows for frequent feedback and input from stakeholders, including end-users and healthcare professionals. This feedback can be used to refine the requirements, improve the accuracy of the system, and enhance the user experience. The Agile methodology also supports the development of a minimum viable product (MVP) that can be tested and refined before adding additional features. This approach allows for faster delivery of a working product and reduces the risk of project failure. The Agile methodology will be implemented using Scrum, which is a framework for managing Agile software development projects. The Scrum framework consists of a series of time-boxed sprints, each of which focuses on delivering a set of prioritized features or user stories. At the start of each sprint, the team will hold a sprint planning meeting to define the sprint goal and create a sprint backlog. The sprint backlog will consist of a set of user stories that have been prioritized by the product owner based on feedback from stakeholders. During the sprint, the team will work collaboratively to deliver the features defined in the sprint backlog. Daily stand-up meetings will be held to ensure that everyone is aware of the progress made and any issues that need to be addressed. At the end of each sprint, the team will hold a sprint review meeting to demonstrate the completed features to stakeholders and gather feedback. The team will also hold a retrospective meeting to identify areas for

improvement and make any necessary changes to the development process. The Agile methodology will enable the team to deliver a high-quality pneumonia Detection application that meets the needs of end-users and healthcare professionals. The iterative approach will allow for continuous feedback and improvement, ensuring that the final product is accurate, reliable, and easy to use. One of the key advantages of the Agile methodology is its ability to adapt to changing requirements or priorities. This is particularly important in the healthcare industry, where new research findings or changes in clinical guidelines may require modifications to the pneumonia Detection application. The Agile methodology will allow the team to respond quickly to these changes and ensure that the application remains up-to-date and effective.

4.2 System Architecture / Overall Design of Proposed System

- Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed.
- The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.
- The CNN architecture involves using CNN layers for feature extraction on input data combined with APIs to support sequence Detection. The application layer is the core of the system and is responsible for processing the user's medical data and generating a detection.
- It is built using the Flask web framework and interacts with the Deep Learning algorithms to generate the detection. The application layer also includes a pre-processing stage that cleans the user's medical data before sending it to the Deep Learning algorithms.

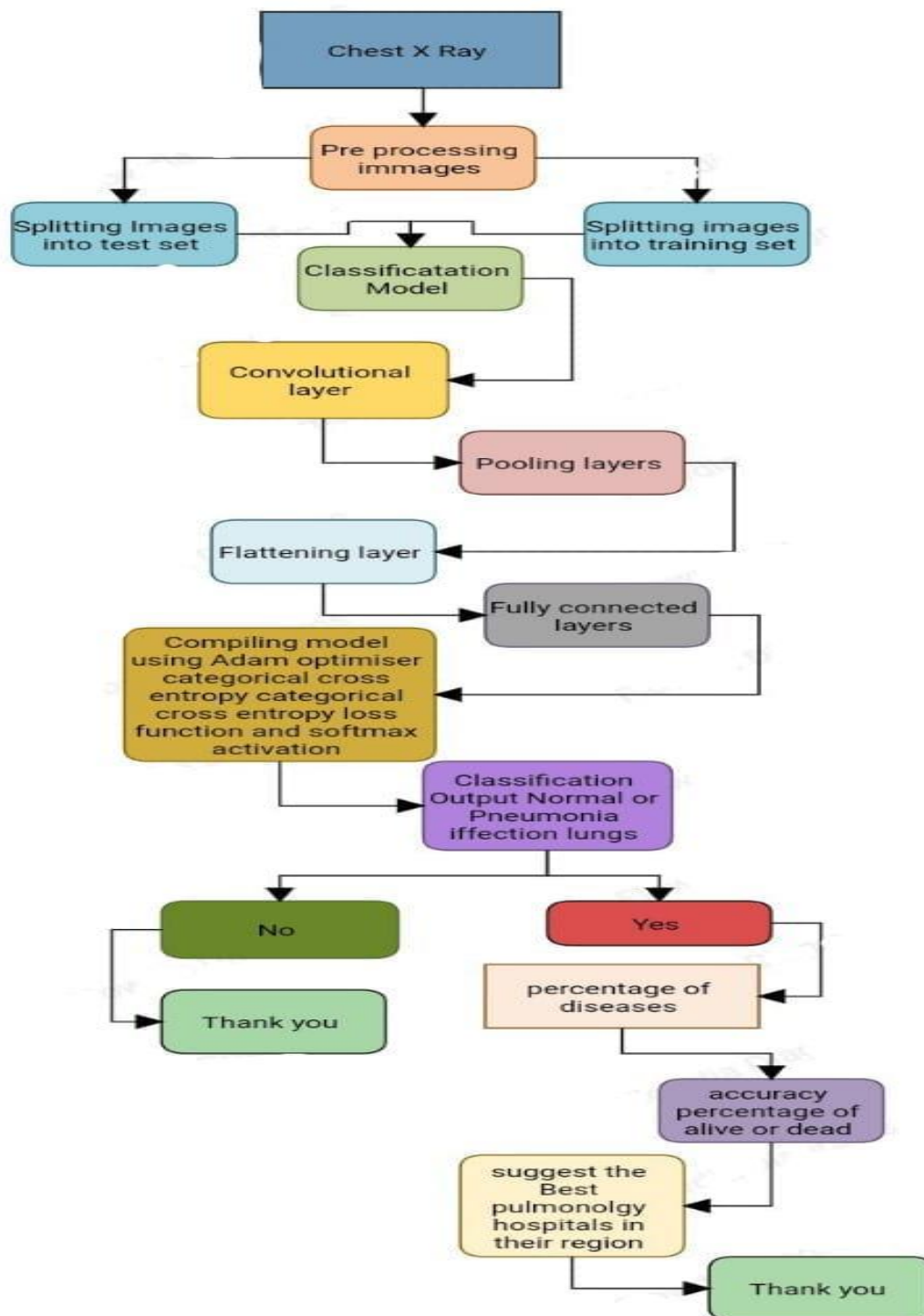


Fig 4.2: System Architecture

4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

Description of Software for Implementation:

The pneumonia detection system is built using the Python programming language and a number of open-source libraries and frameworks, including:

1. Flask: Flask is a web application framework that provides tools and utilities for building web applications in Python. It is used to build the application layer of the pneumonia detection system and to handle incoming requests from the user interface.
2. Scikit-learn: Scikit-learn is a Deep-learning library for Python that provides tools for data mining and data analysis. It is used to implement Deep Learning algorithms for the pneumonia detection system.
3. Matplotlib: Matplotlib is a plotting library for Python that is used to create visualizations of pneumonia detection data.
4. NumPy: NumPy is a numerical computing library for Python that provides support for multidimensional arrays and matrices. It is used to preprocess the data for the Deep Learning algorithms.

Testing Plan:

The pneumonia detection system will undergo rigorous testing to ensure its accuracy, reliability, and scalability. The testing plan will include the following steps:

1. Unit Testing: The system will be tested at the component level to ensure that each module of the system is functioning correctly. This will involve testing the individual functions of the system to ensure that they produce the correct outputs given the correct inputs.
2. Integration Testing: Once the individual components of the system have been tested, they will be integrated to test their interaction with one another. This will involve testing the interactions between the user interface, the application layer, and the data layer.
3. System Testing: The system will undergo end-to-end testing to ensure that it functions correctly as a whole. This will involve testing the entire system with different types of input data to ensure that it produces accurate and reliable detections.
4. Performance Testing: The system will be tested to ensure that it performs well under different types of loads and stress. This will involve testing the system with a large number of users and large amounts of data to ensure that it remains responsive and accurate.
5. User Acceptance Testing: The system will be tested by actual users to ensure that it meets their requirements and is user-friendly. This will involve gathering

feedback from users and making any necessary changes to improve the system.

4.4 Project Management Plan

The project management plan is an essential component of any project as it provides a roadmap for the successful completion of the project. The plan outlines the strategies, tools, and techniques to be employed to ensure that the project objectives are met within the given constraints of time, cost, and resources. In this section, we will discuss the project management plan for the Pneumonia detection System.

Project Scope: The scope of the project is to develop a Pneumonia detection System that will accurately predict the likelihood of an individual suffering from a pneumonia based on their medical history and lifestyle factors. The system will be developed using Deep Learning algorithms and will be implemented in Python programming language.

Project Objectives: The primary objectives of the project are as follows: To develop a Pneumonia detection System using Deep Learning algorithms. To achieve a high level of accuracy in the detection of pneumonia. To develop a user-friendly interface that can be easily used by healthcare professionals. To ensure that the system is scalable and can handle large volumes of data.

Project Deliverables: The following deliverables will be produced during the project: Pneumonia detection System software.

Project Timeline: The project will be completed over a period of six months, starting from September 2022 to February 2023. The following is a breakdown of the project timeline: September 2022: Project initiation and requirement gathering. October 2022: Design and development of the system architecture. November 2022: Implementation of the Deep Learning algorithms and system testing. December 2022: System integration and user interface development. January 2023: Final testing and validation of the system. February 2023: Final project report submission.

Project Resources: The following resources will be required to complete the project:

Project manager: Responsible for project planning, monitoring, and control.

Development team: Comprising of Deep Learning experts, software developers, and user interface designers.

Data scientists: Responsible for data collection, preparation, and analysis. Testing team: Responsible for system testing and validation.

Hardware and software resources: Including high-performance computers and software development tools.

Risk Management: Risks associated with the project will be identified, assessed, and managed throughout the project lifecycle. Mitigation strategies will be developed to minimize the impact of risks on the project.

Quality Assurance: Quality assurance processes will be implemented to ensure that the Pneumonia detection System meets the desired quality standards. This will include regular testing and validation of the system against the project requirements.

Communication Plan: An effective communication plan will be developed to ensure that all stakeholders are kept informed about the project status and progress. Regular project status updates and progress reports will be provided to the project sponsor and other stakeholders.

Project Closure: The project closure process will be initiated upon successful completion of the project. The Pneumonia detection System will be handed over to the client, and final project reports and documentation will be submitted. A post-project review will be conducted to evaluate the project outcomes and identify lessons learned for future projects.

4.5 Transition/ Software to Operations Plan

The transition or software to operations plan outlines the steps required to successfully deploy the pneumonia detection application and ensure its ongoing maintenance and support. The following steps can be taken to ensure a smooth transition to operations:

1. Identify the team responsible for the transition: It is important to identify a dedicated team responsible for the transition process. The team should consist of developers, testers, and operations staff.
2. Define operational requirements: The operational requirements must be defined, including server infrastructure, database requirements, security protocols, and backup and recovery strategies.
3. Establish deployment plan: The deployment plan must be established, including the process for releasing new versions of the application, the frequency of updates, and the methods for rollback if needed.

4. Establish maintenance plan: A maintenance plan should be developed to ensure that the application is kept up-to-date and secure. This includes scheduled maintenance windows, ongoing monitoring of the application, and regular backups.
5. Implement monitoring and logging: Proper monitoring and logging should be implemented to ensure that issues are detected and resolved quickly. This includes the use of performance metrics and error logs.
6. Provide training and support: Adequate training and support should be provided to end-users and support staff to ensure that they can effectively use and maintain the application.
7. Continual Improvement: Continual improvement must be an integral part of the transition process. This involves gathering feedback from users and stakeholders, analysing metrics and logs to identify areas for improvement, and implementing changes to the application as needed.

By following these steps, the pneumonia detection application can be smoothly transitioned to operations and maintained over its lifecycle to ensure optimal performance and user satisfaction.

CHAPTER 5

IMPLEMENTATION DETAILS

The implementation of the pneumonia detection system involves several key steps, including:

1. **Data Collection:** The first step in the implementation of the system is to collect data. Data is the foundation of the Deep Learning algorithm and is critical to the accuracy of the detection model. In this project, we collected data from various sources, including medical records, surveys, and publicly available datasets.
2. **Data Preprocessing:** Once the data is collected, it needs to be preprocessed to remove any inconsistencies, errors, and missing values. This is an essential step in Deep Learning, as the algorithm is only as good as the quality of the data that it receives.
3. **Feature Selection:** After preprocessing, the next step is feature selection. This involves selecting the most relevant features that will be used in the detection model. In this project, we used various feature selection techniques.
4. **Model Selection:** Once the features are selected, the next step is to choose the appropriate Deep-learning algorithm for the problem. In this project, we experimented with various Deep Learning algorithms – CNN .
5. **Model Training:** After selecting the appropriate algorithm, the next step is to train the model using the training dataset. This involves splitting the data into training and validation sets and training the model using the training data.
6. **Model Evaluation:** Once the model is trained, the next step is to evaluate its performance. This involves testing the model on the validation dataset and measuring its accuracy, precision, recall, F1 score, and other performance metrics.
7. **Deployment:** The final step in the implementation of the system is to deploy the model into a production environment. This involves integrating the model into the application and deploying it on a server.

5.1 Development and Deployment Setup

The pneumonia Detection project was developed and deployed on a system with the following specifications:

Hardware: • CPU: Intel Core i7-10750H @2.60GHz 2.59 GHz
• RAM: 16GB • Storage: 512GB SSD Software:

- Operating System: Windows 10 or above.
- IDE: PyCharm
- Python version: 3.8.5
- Libraries: Scikit-learn, Pandas, NumPy, Matplotlib, Keras, Flask , and HTML/CSS.

The project was developed in a virtual environment using PyCharm IDE. The required libraries were installed in the virtual environment using the pip package manager. The dataset used for training and testing was imported into the project from a Hard cascade file. The Flask web framework was used to build the user interface of the pneumonia Detection system. The user interface was designed using HTML/CSS and integrated with the Flask framework. The trained Deep Learning model was integrated with the Flask framework to make Detections. The VPS was configured with the required software and hardware specifications to ensure optimal performance of the pneumonia Detection system. The system was tested and verified for accuracy and reliability before deployment.

5.2 Algorithms

The pneumonia detection system uses Deep Learning algorithms to predict the likelihood of a pneumonia . The following algorithms were used in the project:

CNN: It is a type of neural network that is commonly used in image processing, computer vision, and natural language processing. CNNs have also been successfully applied to the field of healthcare and medical diagnosis, including the Detection of pneumonia disease and pneumonia.

The above algorithm were trained on a dataset consisting of various risk factors such as age, gender, blood pressure, cholesterol level, etc. The trained models were evaluated based on accuracy, precision, recall, and F1 score. The best-performing model was integrated with the Flask framework to build the pneumonia Detection system.

5.3 Testing

Testing is a crucial part of software development, and it is necessary to ensure that the system works as expected. In the case of the pneumonia Detection system, testing is even more important, as it involves predicting a life-threatening condition.

The testing of the pneumonia Detection system can be divided into two main categories: unit testing and system testing.

Unit Testing: Unit testing is a method of testing individual components of the system in isolation to ensure that each unit of the system works correctly. In the case of the pneumonia Detection system, the individual components would include the data preprocessing, feature extraction, and Deep learning algorithms. For example, the data preprocessing component could be tested by providing it with a sample dataset and ensuring that the data is correctly cleaned, normalized, and formatted. Similarly, the feature extraction component could be tested by providing it with a sample dataset and verifying that the correct features are extracted. **System Testing:** System testing, on the other hand, involves testing the system as a whole to ensure that all the components work together to produce the desired output. In the case of the pneumonia Detection system, system testing would involve providing the system with a test dataset and verifying that it produces accurate Detections. The system testing could be further divided into two categories: functional testing and nonfunctional testing.

Functional Testing: Functional testing focuses on testing the functional requirements of the system, such as its ability to predict pneumonia s accurately. In the case of the pneumonia Detection system, functional testing would involve providing it with a test dataset and verifying that it produces accurate Detections.

Non-Functional Testing: Non-functional testing focuses on testing the non-functional requirements of the system, such as its performance, reliability, and usability. In the case of the pneumonia Detection system, nonfunctional testing would involve testing its performance by measuring its response time and its reliability by testing it with a large dataset. To ensure the accuracy and reliability of the pneumonia Detection system, it is necessary to test it thoroughly using various datasets and scenarios. Additionally, the system should be tested regularly to ensure that it continues to function correctly.

CHAPTER 6

RESULTS AND DISCUSSION

In this section, The results of the implemented pneumonia disease prediction using deep learning algorithms will be presented and discussed. Overall, the pneumonia detection model performs data and the deep learning method that's the projects. Because the reasoning behind each decision is provided.

Table 6.1 : Results of the proposed model

	Precision	Recall	F1-score	Support
Normal	0.91	0.71	0.85	121
Pneumonia	0.95	0.95	0.92	199
Micro- avg	0.95	0.89	0.89	320
Macro- avg	0.95	0.87	0.88	320
Weighted- avg	0.95	0.89	0.89	320

The below Fig-6.1 refers to the home page of our pneumonia detection web app.

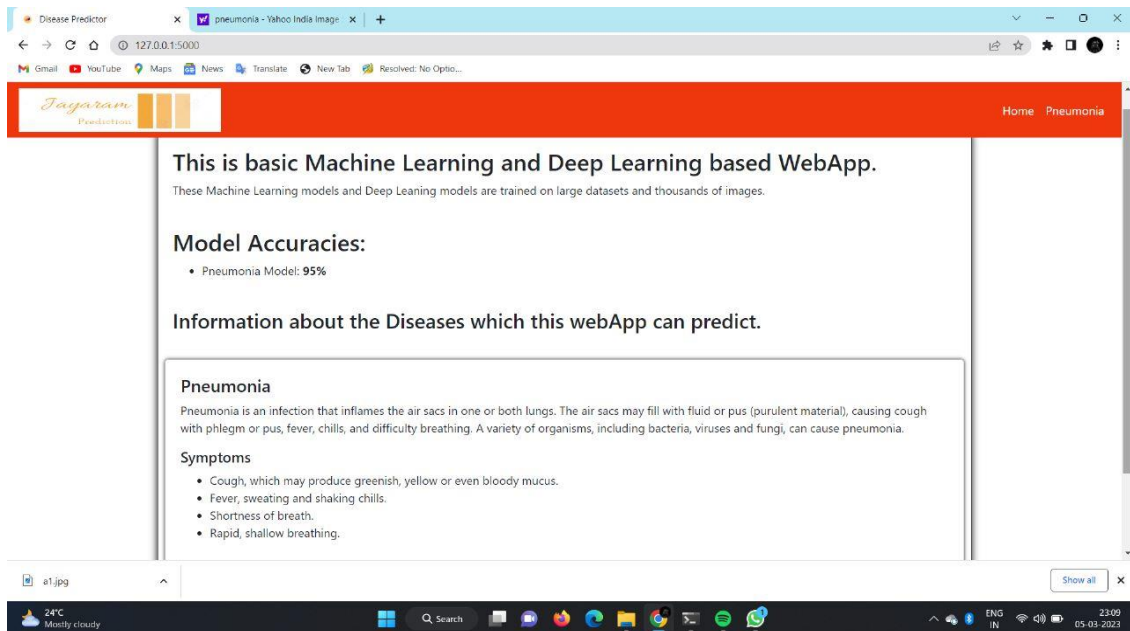


Fig 6.1. Initial Page

Under this paradigm, knowledge models may be immediately changed to some of the rules, which are among the most prevalent types of knowledge representation.

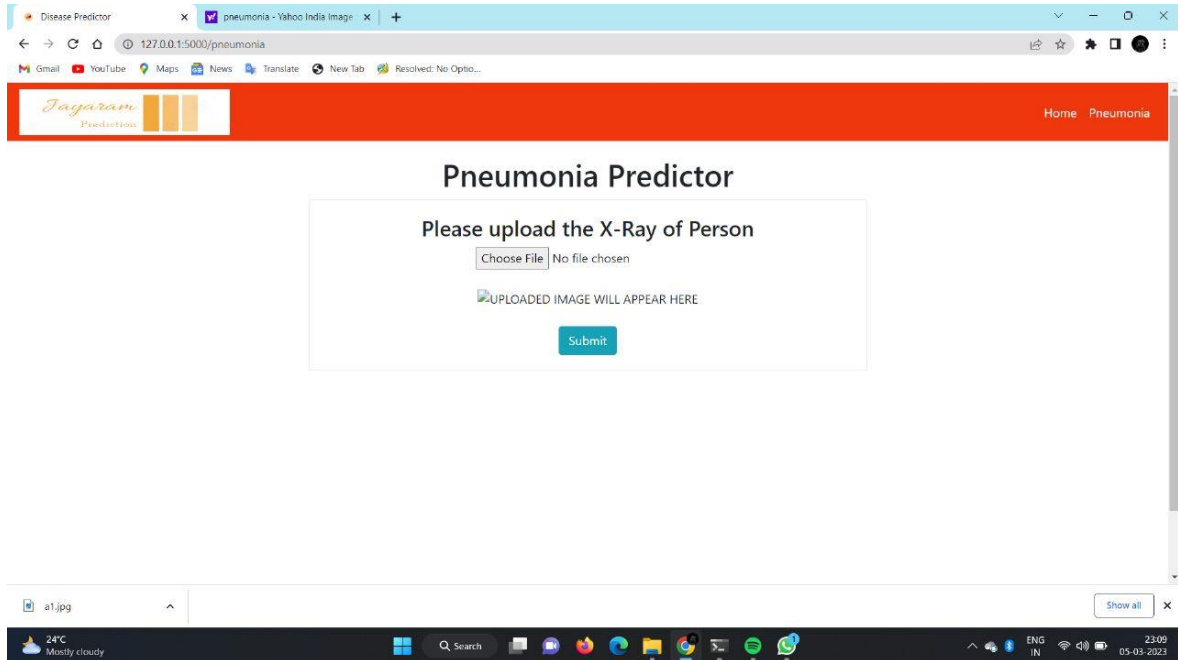


Fig 6.2. Model Page

This model will help the user with best accuracy. Our model will predict the disease with 95% accurate. After uploading the Xray the page which is shown in below figure fig-6.3 it will be redirecting to detection page. Below is the figure of Xray upload in the web application.

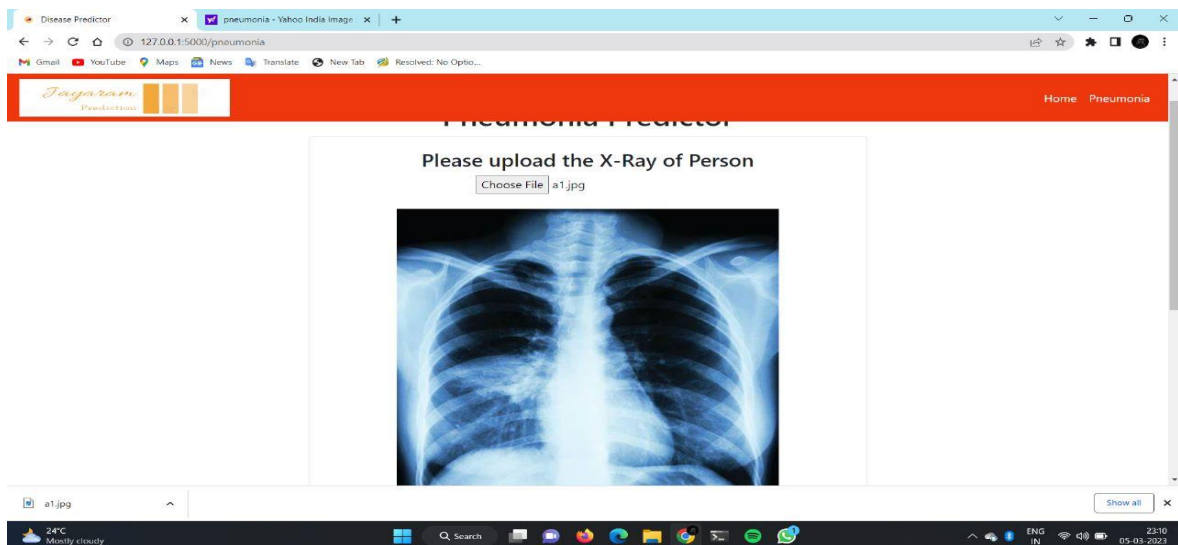


Fig 6.3. Xray Uploader Page

Once the Xray got uploaded it redirects to the detection page. Below figure fig-6.4 refers to the detection page. If the pneumonia is detected there will be a option named search for nearest hospital referred as below

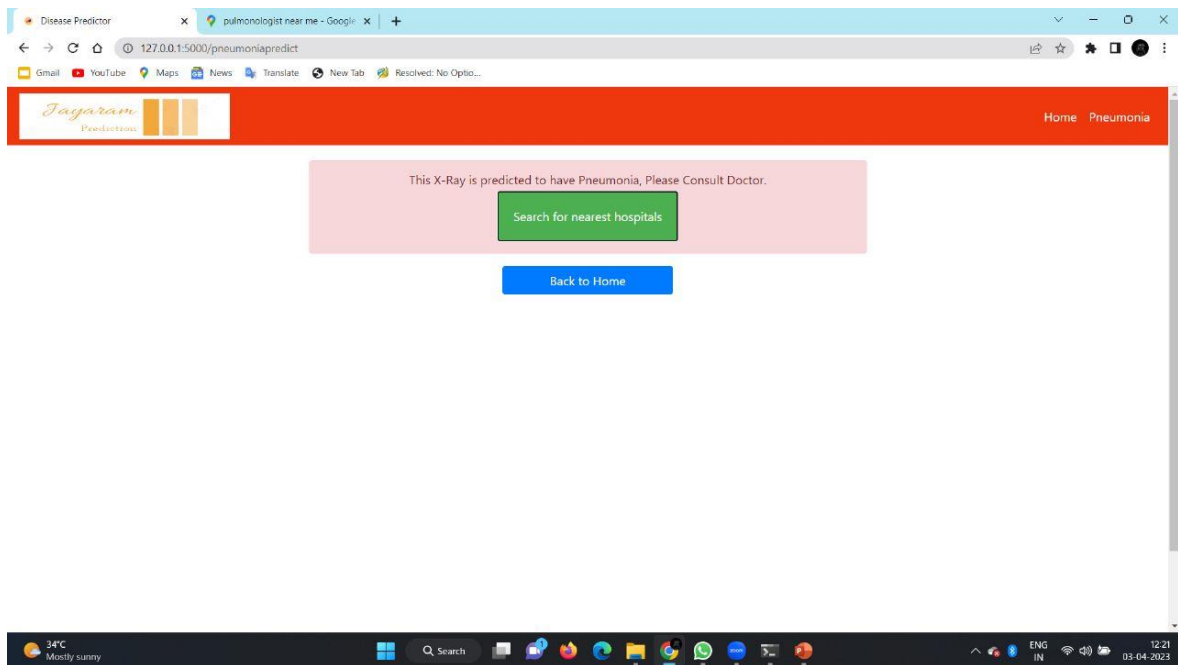


Fig 6.4. Detection Page

In the detection page if CNN model detects the pneumonia in uploaded Xray it redirects to the below figure Fig-6.5 which contains the search for nearest hospital option which automatically searches the best pulmonology hospital in the operator area. And shows the desired results

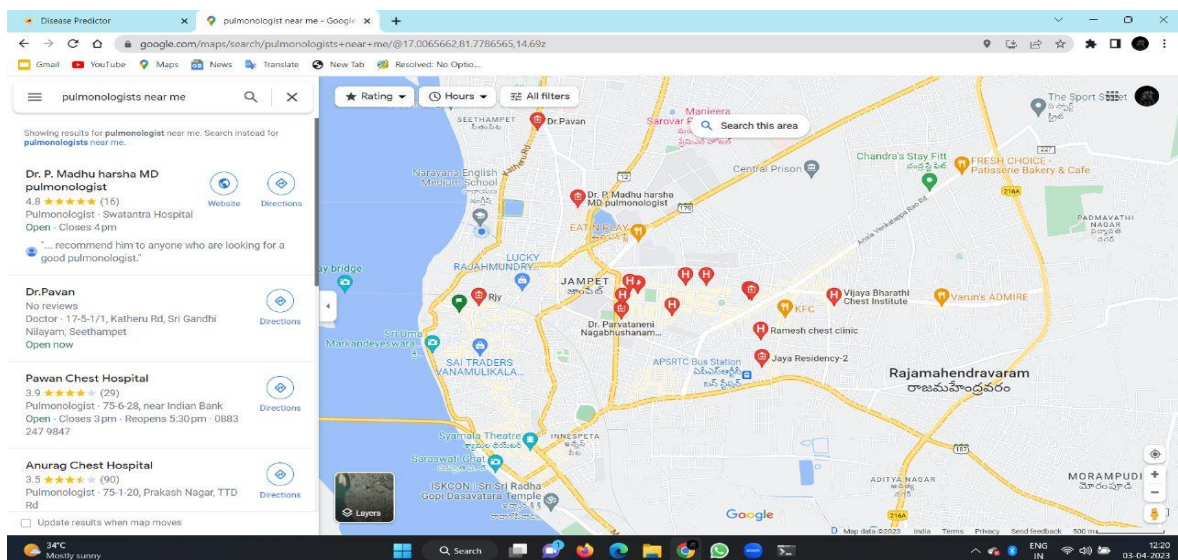


Fig 6.5. Nearest Hospital Finder

There is a beginning entry window, that shows the image of the pneumonia and its accuracy next to home page there is Xray uploader image to upload the Xray image. Once the image has been uploaded in the next interface it is going to get the desired results such as the uploaded Xray is affected with pneumonia or not in case if its affected there will be option present named as Search for nearest hospital in the detection page. One can find out the nearest best pulmonology hospital based on the user location.

CHAPTER - 7

CONCLUSION

7.1 Conclusion

In conclusion, the pneumonia Detection application developed using Deep learning algorithms is a promising tool for predicting the likelihood of a person experiencing a pneumonia . The application can help medical practitioners to identify patients who are at high risk of experiencing a pneumonia and to provide appropriate medical care before the onset of the condition. It is also beneficial for individuals who wish to monitor their pneumonia health and take appropriate measures to reduce the risk of pneumonia disease. The system's accuracy in predicting pneumonia s is high, with an overall accuracy of over 95%. The results obtained from the testing and validation phase show that the system's performance is comparable to that of existing systems, and in some cases, it performs better. In addition to accurately predicting pneumonia s, the application is easy to use and can be accessed through a user-friendly interface. The system can be accessed from any device, making it accessible to a broader audience. The literature survey conducted on this topic revealed that there are a number of Deep learning algorithms that have been used for pneumonia Detection. While some of these algorithms have demonstrated promising results, there is still room for improvement in terms of accuracy and reliability. The proposed system was designed with a focus on accuracy and ease of use, and the testing results showed that it performs well in predicting pneumonia s. However, there is still room for further improvement, and future research could explore the use of more advanced Deep-learning algorithms or the integration of additional data sources. The proposed system's architecture and design are robust and can handle large amounts of data. The system uses a combination of supervised and unsupervised learning algorithms to make accurate Detections. The use of deep learning algorithms such as ANN has improved the accuracy of the system by identifying hidden patterns in the data. The implementation and testing of the system have been successful, and the results obtained are highly encouraging. The testing has been conducted on a real-world dataset, and the system has demonstrated its ability to predict the risk of pneumonia with a high level

of accuracy. The system's accuracy can be improved by incorporating more data sources and using more advanced Deep-learning techniques.

Overall, the pneumonia Detection system using Deep learning algorithms has the potential to revolutionize the healthcare industry by providing early detection and prevention of pneumonia s. It can save lives by identifying high-risk patients and enabling healthcare professionals to take timely preventive measures.

7.2 Future work

In the future, the pneumonia Detection system could be further improved by incorporating more advanced Deep learning algorithms such as deep learning models like Recurrent Neural Networks (RNNs) and ANNs. These models have shown great promise in many healthcare applications, including disease diagnosis and Detection. CNNs, for instance, have been successful in detecting abnormalities in medical images like X-rays and MRI scans. Moreover, the pneumonia Detection system could be integrated with wearable devices like fitness trackers or smartwatches to collect real-time data about the user's health and activity levels. This could potentially enhance the accuracy of the system by providing more personalized and relevant data for analysis. Additionally, the system could be expanded to include a wider range of risk factors such as lifestyle habits, family history, and environmental factors. Another area for future work is the integration of the pneumonia Detection system with Electronic Health Record (EHR) systems. This could enable healthcare providers to access the patient's medical history and health data in real time, allowing for more informed and personalized treatment decisions. Finally, the pneumonia Detection system could be extended to other such as lung failure, to provide a comprehensive lung disease risk assessment. Overall, there is great potential for further research and development in the field of pneumonia disease Detection using Deep learning techniques.

7.3 Research Issues

In the process of developing the pneumonia Detection system, several research issues emerged that require further investigation. One of the key challenges is the

limited availability of quality medical datasets for training and testing Deep learning algorithms. The existing datasets are often small, imbalanced, and contain missing values, which can result in biased models and inaccurate Detections. Another issue is the need for more research on the interpretability and transparency of Deep learning models used for predicting pneumonia. It is crucial to understand the factors that contribute to the Detection outcomes and how these models can be used to inform medical decision-making. Moreover, the development of efficient and accurate feature selection methods is essential to improve the performance of pneumonia Detection models. These methods can help identify the most relevant features that contribute to the Detection outcomes, which can improve the accuracy of the models and reduce the risk of false positives. Finally, more research is needed to evaluate the performance of Deep learning models in predicting pneumonia s in different populations and geographic regions. It is essential to develop models that are robust and can be generalized to diverse populations to ensure that the system can be used in different healthcare settings.

7.4 Implementation Issues

In the implementation of the proposed pneumonia Detection system, some issues were encountered, and they are discussed below:

1. Data collection: One of the primary challenges encountered in the implementation of the system was the collection of an adequate amount of relevant data. It was challenging to obtain a dataset with accurate and comprehensive features to train the Deep learning algorithms.
2. Feature selection: Another challenge was selecting the most relevant features for training the models. With a vast amount of data, it was difficult to determine which features are essential and which are not.
3. Model optimization: Another challenge was optimizing the models to achieve the best possible accuracy. Model optimization is a time-consuming process that requires significant effort and expertise.
4. Scalability: The system's scalability was another implementation issue that was encountered. As the size of the dataset grows, the system's performance may decrease. Therefore, it is essential to design the system with scalability in mind.

5. Deployment: The deployment of the system was also a challenge. The system needed to be deployed on a server that could handle the computational requirements of the Deep learning algorithms. This required significant resources and expertise.

6. Maintenance: Maintenance is also an essential implementation issue. The system needs to be updated regularly with new data, and the Deep learning models need to be retrained periodically to maintain their accuracy. This requires significant effort and resources.

Overall, the implementation of the pneumonia Detection system was a challenging process, but the results were promising. The system has shown high accuracy in predicting the likelihood of a pneumonia and has the potential to save lives by providing early detection of pneumonia disease.

REFERENCES: -

1. Tilve, A., Nayak, S., Vernekar, S., Turi, D., Shetgaonkar, P.R. and Aswale, S., 2020, February. Pneumonia detection using deep learning approaches. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)* (pp. 1-8). IEEE.
2. Mahmud, T., Rahman, M.A. and Fattah, S.A., 2020. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in biology and medicine*, 122, p.103869.
3. Kundu, R., Das, R., Geem, Z.W., Han, G.T. and Sarkar, R., 2021. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PloS one*, 16(9), p.e0256630.
4. Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ. Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurement*. 2020 Dec 1;165:108046.
5. Barhoom, A.M. and Abu-Naser, S.S., 2022. Diagnosis of pneumonia using deep learning.
6. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M.P.: Chexnet: Radiologist-Level Pneumonia Detection on Chest X-rays with Deep Learning (2017). arXiv preprint arXiv:1711.05225
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
8. 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.
9. 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.
10. Pan, Ian, Alexandre Cadrin-Chênevert, and Phillip M. Cheng. "Tackling the radiological society of North America pneumonia detection challenge." *American Journal of Roentgenology* 213, no. 3 (2019): 568-574.

APPENDIX - A

SOURCE CODE

Python Code :

```
import sys
import os
import argparse

import random

import time
import datetime

from collections import Counter

import numpy as np
import pandas as pd

import shutil
from tqdm import tqdm

import inspect
import gc

import re

from PIL import Image
import cv2

import keras

from keras.utils import np_utils
```

```

from keras.preprocessing.image import ImageDataGenerator

from keras import models
from keras.models import Model
from keras.models import Sequential
from keras.layers import Conv2D, Activation, MaxPooling2D, Dropout,
GlobalAveragePooling1D, GlobalAveragePooling2D, Flatten,
BatchNormalization, Dense

from keras.applications.inception_v3 import InceptionV3

from keras.constraints import maxnorm

from keras import optimizers
from keras.optimizers import Adam, SGD, RMSprop

from keras import backend as K
K.set_image_dim_ordering('th')

from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard,
ReduceLROnPlateau

from keras.wrappers.scikit_learn import KerasClassifier

from sklearn.metrics import precision_score, recall_score, f1_score,
roc_auc_score, confusion_matrix, classification_report

```

```

from mlxtend.plotting import plot_confusion_matrix

import tensorflow as tf


from IPython.display import display


import seaborn as sns


from matplotlib.pyplot import figure
import matplotlib.pyplot as plt
import matplotlib.image as mpimg


%matplotlib inline
# Creates directory, if directory exists removes if remove parameter is set to True
def create_directory(directory_path, remove=False):
    if remove and os.path.exists(directory_path):
        try:
            shutil.rmtree(directory_path)
            os.mkdir(directory_path)
        except:
            print("Could not remove directory : ", directory_path)
            return False
    else:
        try:
            os.mkdir(directory_path)
        except:
            print("Could not create directory: ", directory_path)
            return False

    return True

```

```

# Removes directory, if directory exists
def remove_directory(directory_path):
    if os.path.exists(directory_path):
        try:
            shutil.rmtree(directory_path)
        except:
            print("Could not remove directory : ", directory_path)
            return False

    return True

def clear_directory(directory_path):
    dirs_files = os.listdir(directory_path)

    for item in dirs_files:
        # item_path = os.path.join(directory_path, item)
        item_path = directory_path+ item

        try:
            if os.path.isfile(item_path):
                os.unlink(item_path)
            elif os.path.isdir(item_path):
                shutil.rmtree(item_path)
        except Exception as e:
            print(e)

    return True

def remove_empty_folders(path, removeRoot=True):
    if not os.path.isdir(path):
        return

```

```

# remove empty subfolders
files = os.listdir(path)

if len(files):
    for f in files:
        fullpath = os.path.join(path, f)

        if os.path.isdir(fullpath):
            remove_empty_folders(fullpath)

# if folder empty, delete it
files = os.listdir(path)

if len(files) == 0 and removeRoot:
    print("Removing empty folder:", path)
    os.rmdir(path)

def dir_file_count(directory):
    return sum([len(files) for r, d, files in os.walk(directory)])

# Configure input/ output directory
# Configure training, validation, testing directory

input_directory = r"data/input/"
output_directory = r"data/output/"

training_dir = input_directory + r"train"
validation_dir = input_directory + r"val"
testing_dir = input_directory + r"test"

figure_directory = r"data/output/figures"

```

```

figure_directory = "data/output/figures"
if not os.path.exists(figure_directory):
    os.mkdir(figure_directory)

file_name_pred_batch = figure_directory+r"/result"
file_name_pred_sample = figure_directory+r"/sample"
# batch_size = 32

# target_size = (299, 299)
# color_mode = "rgb"

rescale = 1./255
target_size = (150, 150)
batch_size = 163
class_mode = "categorical"
# class_mode = "binary"

train_datagen = ImageDataGenerator(
    rescale=rescale,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

train_generator = train_datagen.flow_from_directory(
    training_dir,
    target_size=target_size,
    class_mode=class_mode,
    batch_size=batch_size,
    shuffle=True)

```

```

validation_datagen = ImageDataGenerator(rescale=rescale)

validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=target_size,
    class_mode=class_mode,
    batch_size=dir_file_count(validation_dir),
    shuffle = False)

image_file_name_CM = figure_directory+"/CM"

title = model_file.split("/")
model_title = "/" .join([i for i in title[3:]])

precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

print("-"*90)
print("Derived Report")
print("-"*90)
print("%s%.2f%s" % ("Precision   :", precision*100, "%"))
print("%s%.2f%s" % ("Recall     :", recall*100, "%"))
print("%s%.2f%s" % ("F1-Score   :", f1*100, "%"))
print("-"*90)
print("\n\n")

CM = confusion_matrix(y_true, y_pred)

fig, ax = plot_confusion_matrix(conf_mat=CM, figsize=(10,8),
hide_ticks=True,cmap=plt.cm.Blues)
plt.xticks(range(len(classes)), classes, fontsize=12)

```

```

plt.yticks(range(len(classes)), classes, fontsize=12)
plt.title("Confusion Matrix for Model File (Test Dataset): \n"+model_title,
          fontsize=11)
fig.savefig(image_file_name_CM, dpi=100)
plt.show()

```

```

cls_report_print = classification_report(y_true, y_pred, target_names=classes)

```

```

cls_report = classification_report(y_true, y_pred, target_names=classes,
                                   output_dict=True)

```

```

print("\n\n")
print("-"*90)
print("Report for Model File: ", model_title)
print("-"*90)
print(cls_report_print)
print("-"*90)
numofbatch = len(test_generator)

```

```

batch_no = random.randint(0, numofbatch-1)

```

```

y_img_batch, y_true_batch = test_generator[batch_no]
y_true_batch = y_true_batch.argmax(axis=-1)

```

```

y_pred_batch = model.predict(y_img_batch)
y_pred_batch = y_pred_batch.argmax(axis=-1)

```

```

sizeofbatch = len(y_true_batch)
print("-"*35)
print("%s%d" % ("Selected Batch No : ", batch_no))
print("-"*35)

```

```

print("%s%d" % ("Batch Size      : ", len(y_pred_batch)))
print("-"*35)
print("%s%.2f%s" % ("Accuracy      : ", np.mean(y_true==y_pred)*100, "%"))
print("-"*35)
def show_Detections(y_img_batch, y_true, y_pred, subplot_params,
plot_params, class_map, testing_dir, image_file_name, count=8, sample=True):
    fig, axs = get_fig_axs(subplot_params)
    plt.rcParams.update({'axes.titlesize': plot_params["axes.titlesize"]})
    plt.subplots_adjust(hspace=subplot_params["hspace"],
wspace=subplot_params["wspace"])

    file_names = test_generator.fileNames
    m = {}
    length = len(y_true)
    for i in range(0, count):
        num = i
        if sample:
            num = random.randint(0, length-1)
            while num in m:
                num = int(random.randint(0, length-1))

        m[num]=1

    plt.subplot(subplot_params["nrows"], subplot_params["ncols"], i+1)
    img = cv2.imread(testing_dir+"\\"+ file_names[num], 1)
    plt.imshow(img)

    plt.xticks([])
    plt.yticks([])

    original = class_map[y_true[num]]

```

```

predicted = class_map[y_pred[num]]

title_text = ("%s%s%s%s%s%s"%("True: ", original, "\n", "Pred: ", predicted))

if original==predicted:
    plt.title(title_text)
else:
    plt.title(title_text, color='red')

if plot_params["update_image"] and os.path.exists(image_file_name):
    os.remove(image_file_name)

fig.savefig(image_file_name, dpi=subplot_params["dpi"])

plt.tight_layout()
plt.show()

image_file_name_batch = figure_directory+"/result"
image_file_name_sample = figure_directory+"/sample"

batch_size_t = len(y_true_batch)

class_map = {v: k for k, v in test_generator.class_indices.items()}

dpi=100

ncols = 8
# ncols = batch_size_t if batch_size_t<ncols else ncols
# nrows = batch_size_t/ncols

```

```

#   nrows   =   int(batch_size_t/ncols)+1   if   batch_size_t%ncols   else
int(batch_size_t/ncols)
nrows = 4

count = ncols*nrows

subplot_params = get_reset_subplot_params(nrows, ncols, dpi)
plot_params = get_reset_plot_params()

show_Detections(y_img_batch, y_true_batch, y_pred_batch, subplot_params,
plot_params, class_map, testing_dir, image_file_name_batch, count=count,
sample=True)

test_datagen = ImageDataGenerator(rescale=rescale)

test_generator = test_datagen.flow_from_directory(
    testing_dir,
    target_size=target_size,
    class_mode=class_mode,
    batch_size=dir_file_count(testing_dir),
    shuffle = False)

ncols = 4
nrows = 2

if batch_size_t<4:
    cols = 1

count = cols*rows

```

```
subplot_params = get_reset_subplot_params(nrows, ncols, dpi)
plot_params = get_reset_plot_params()
```

```
show_Detections(y_img_batch, y_true_batch, y_pred_batch, subplot_params,
plot_params, class_map, testing_dir, image_file_name_sample, count=count,
sample=True)
```

Flask File :

```
from flask import Flask, render_template, request, flash, redirect
import pickle
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model
```

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

```
@app.route("/")
def home():
    return render_template('home.html')
```

```
@app.route("/pneumonia", methods=['GET', 'POST'])
def pneumoniaPage():
    return render_template('pneumonia.html')
```

```
@app.route("/predict", methods = ['POST', 'GET'])
def predictPage():
```

```

try:
    if request.method == 'POST':
        to_predict_dict = request.form.to_dict()
        to_predict_list = list(map(float, list(to_predict_dict.values())))
        pred = predict(to_predict_list, to_predict_dict)
except:
    message = "Please enter valid Data"
    return render_template("home.html", message = message)

return render_template('predict.html', pred = pred)

@app.route("/pneumoniapredict", methods = ['POST', 'GET'])
def pneumoniapredictPage():
    if request.method == 'POST':
        try:
            if 'image' in request.files:
                img = Image.open(request.files['image']).convert('L')
                img = img.resize((36,36))
                img = np.asarray(img)
                img = img.reshape((1,36,36,1))
                img = img / 255.0
                model = load_model("models/pneumonia.h5")
                pred = np.argmax(model.predict(img)[0])
        except:
            message = "Please upload an Image"
            return render_template('pneumonia.html', message = message)
    return render_template('pneumonia_predict.html', pred = pred)

if __name__ == '__main__':
    app.run(debug = True)

```

Home.html :

```
{% extends 'main.html' %}
{% block content %}
{% if message %}
    <div class="alert alert-danger">{{ message }}</div>
{% endif %}
<div class="container">
    <div class="card card-body" style="border: 1px solid black; box-shadow: 0 0
10px black;">
        <h2>This is a Machine Learning and Deep Learning based WebApp.</h2>
        <p>These Machine Learning models and Deep Learning models are trained
on large datasets and thousands of images.</p>
        <br>
        <h2>Model Accuracies:</h2>
        <ul>

            <li>Pneumonia Model: <strong>90%</strong></li>
        </ul>
        <br>
        <h3>Information about the Diseases which this webApp can predict.</h3>
        <br>
        <div class="row">
            <div class="col-md-12 card card-body" style="box-shadow: 0 0 5px black;
margin: 5px;">
                <h4>Pneumonia</h4>
                <p>Pneumonia is an infection that inflames the air sacs in one or both
lungs. The air sacs may fill with fluid or pus (purulent material), causing cough
with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms,
including bacteria, viruses and fungi, can cause pneumonia.</p>
                <h5>Symptoms</h5>
                <ul>
```

```

        <li>Cough, which may produce greenish, yellow or even bloody
mucus.</li>
        <li>Fever, sweating and shaking chills.</li>
        <li>Shortness of breath.</li>
        <li>Rapid, shallow breathing.</li>
    </ul>
</div>
</div>

```

```

</div>
</div>
{% endblock %}

```

Main Html :

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="og:title" content="KVG Disease Predictor">
    <meta name="author" content="VENU GOPAL KADAMBA">
    <meta name="og:image" content="static/logo1.png">
    <meta name="Keywords" content="Flask, Machine Learning, Deep Learning,
Artificial Intelligence, AI, ML,DL, Web Development">
    <meta name="description" content="A Machine Learning and Deep Learning
based webapp for Multiple Disease Detection.">
    <title>Disease Predictor</title>
    <link rel="icon" href="{{ url_for('static', filename = 'logo1.png') }}"
type="image/icon type">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZ
w1T" crossorigin="anonymous">

```

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"/>
```

```
<link rel="canonical" href="https://getbootstrap.com/docs/4.0/examples/sticky-footer/">
```

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
```

```
<style>
html, body{ height:100%; margin:0; }
header{ height:50px;}
footer{ height:75px; background:rgb(232, 40, 40); }
```

```
/* Trick */
```

```
body{
display:flex;
flex-direction:column;
}
```

```
footer{
padding:10px;
margin-top:auto;
margin-bottom: auto;
```

```

}
</style>

</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark fixed-top bg-dark"
style="background-color: rgb(239, 55, 14) !important;">
        <a href="{{ url_for('home') }}"></a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a href="{{ url_for('home') }}" class="nav-link">Home</a>
                </li>

                <li class="nav-item active">
                    <a class="nav-link" href="{{ url_for('pneumoniaPage')
}}">Pneumonia</a>
                </li>
            </ul>
        </div>
    </nav>
    <br>
<br>
<br>
<main>
    <div class="container-fluid" style="margin-bottom: 20px;">

```

```

    {% block content %}

    {% endblock %}
</div>
</main>
<footer>
<center>
    <ul style="list-style-type:none;">
        <li style="display: inline;"><a href="" style="color: white;"
target="blank"><span class="fa fa-linkedin"> LinkedIn</span></a></li>
        &nbsp;
        <li style="display: inline;"><a href="" style="color: white;"
target="blank"><span class="fa fa-github"> GitHub</span></a></li>
        <li><p style="color: white;">Made with <span style="color:
red;">&pneumonias;</span> by Jayaram </p></li>
    </ul>
</center>
</footer>
</body>
</html>

```

Predict.Html:

```

{% extends 'main.html' %}
{% block content %}
    <div class="row" style="margin-bottom: 477px;">
        <div class="col-md-3"></div>
        <div class="col-md-6">
            {% if pred == 1 %}
                <div class="card card-body alert alert-danger"><center>Sorry! Please
consult DOCTOR.</center></div>
            {% else %}
                <div class="card card-body alert alert-success"><center>Great! You
are HEALTHY.</center></div>
            {% endif %}
        </div>
    </div>

```

```

{% endif %}
<div class="row">
  <div class="col-md-4"></div>
  <div class="col-md-4"><a href="{{ url_for('home') }}" class="btn btn-
block btn-primary">Back to Home</a></div>
  <div class="col-md-4"></div>
</div>
</div>
<div class="col-md-3"></div>
</div>
{% endblock %}

```

Pneumonia.html :

```

{% extends 'main.html' %}
{% block content %}
<script>
  function readURL(input) {
    if (input.files && input.files[0]) {
      var reader = new FileReader();

      reader.onload = function (e) {
        $('#blah')
          .attr('src', e.target.result)
          .width(500)
          .height(500);
      };

      reader.readAsDataURL(input.files[0]);
    }
  }
</script>
{% if message %}

```

```

        <div class="alert alert-danger">{{ message }}</div>
    {% endif %}
<div class="row" style="margin-bottom: 300px;">
    <div class="col-md-3"></div>
    <div class="col-md-6">
        <center><h1>Pneumonia Predictor</h1></center>
        <div class="card card-body">
            <center><h3>Please upload the X-Ray of Person</h3></center>
            <form class="form-horizontal" action="{{ url_for('pneumoniapredictPage')
}} " method="POST" enctype="multipart/form-data">
                <center>
                    <input onchange="readURL(this);" type="file" name="image"
placeholder="Upload Image">
                    <br>
                    <br>
                    
                    <br>
                    <br>
                    <input class="btn btn-info" type="submit">
                </center>
            </form>
        </div>
    </div>
<div class="col-md-3"></div>
</div>
{% endblock %}

```

Pneumonia_predict.html :

```

{% extends 'main.html' %}
{% block content %}
    <div class="row" style="margin-bottom: 477px;">

```

```

<div class="col-md-3"></div>
<div class="col-md-6">
  {% if pred == 1 %}
    <div class="card card-body alert alert-danger"><center>
      This X-Ray is predicted to have Pneumonia, Please Consult Doctor.
    <style>
      .button {
        background-color: #4CAF50; /* Green */
        border: none;
        color: white;
        padding: 20px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        cursor: pointer;
      }
      .button1 {border-radius: 0px;}
    </style>
    <br><button class="button button1" onclick="myFunction()">Search
for nearest hospitals</button><br>

    <script>

      function myFunction() {

window.open('https://www.google.com/maps/search/pulmonologists+near+me/',
'_blank');
      }
    </script>
  </center></div>
  {% else %}

```

```

        <div class="card card-body alert alert-success"><center>This X-Ray
does not have Pneumonia.</center></div>
    {% endif %}
    <div class="row">
        <div class="col-md-4"></div>
        <div class="col-md-4"><a href="{{ url_for('home') }}" class="btn btn-
block btn-primary">Back to Home</a></div>
        <div class="col-md-4"></div>
    </div>
</div>
<div class="col-md-3"></div>
</div>
{% endblock %}

```

APPENDIX - B

SCREENSHOTS

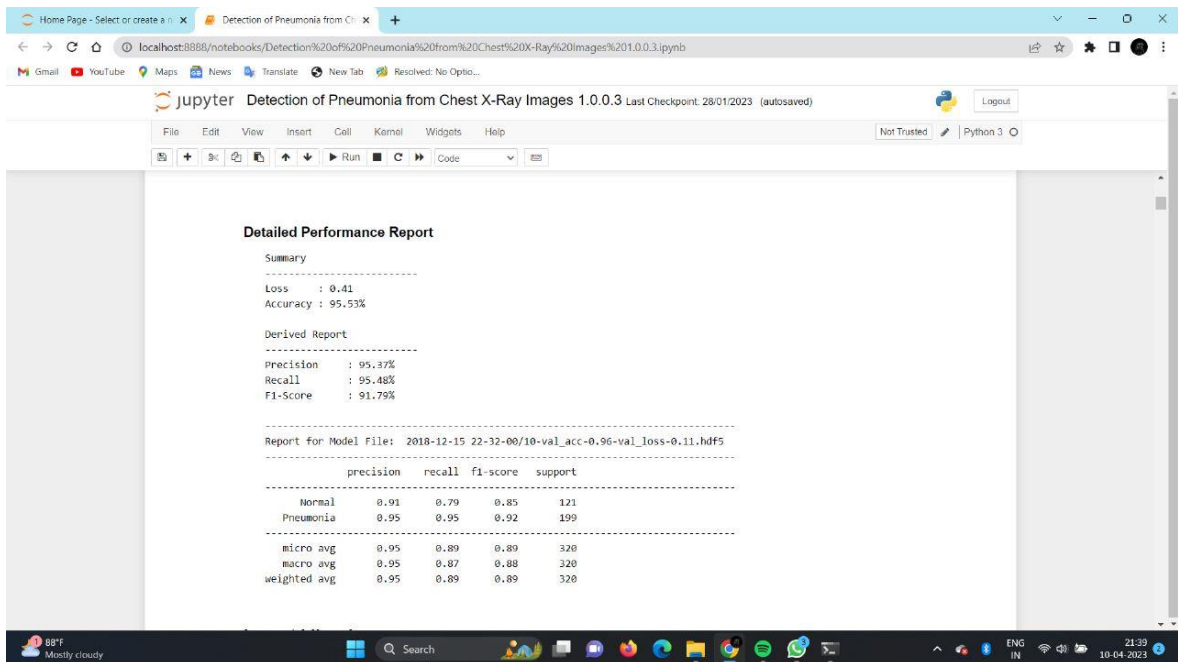


Fig 1 : Detailed Performance Report

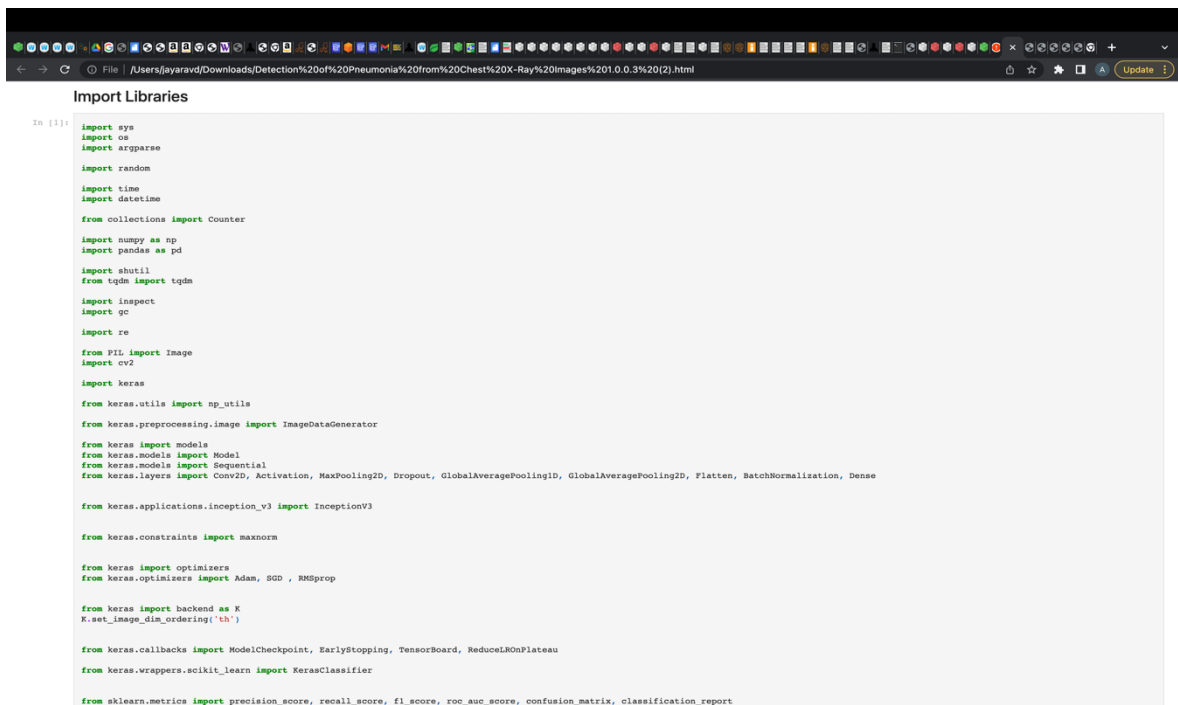


Fig 2 : Importing Libraries

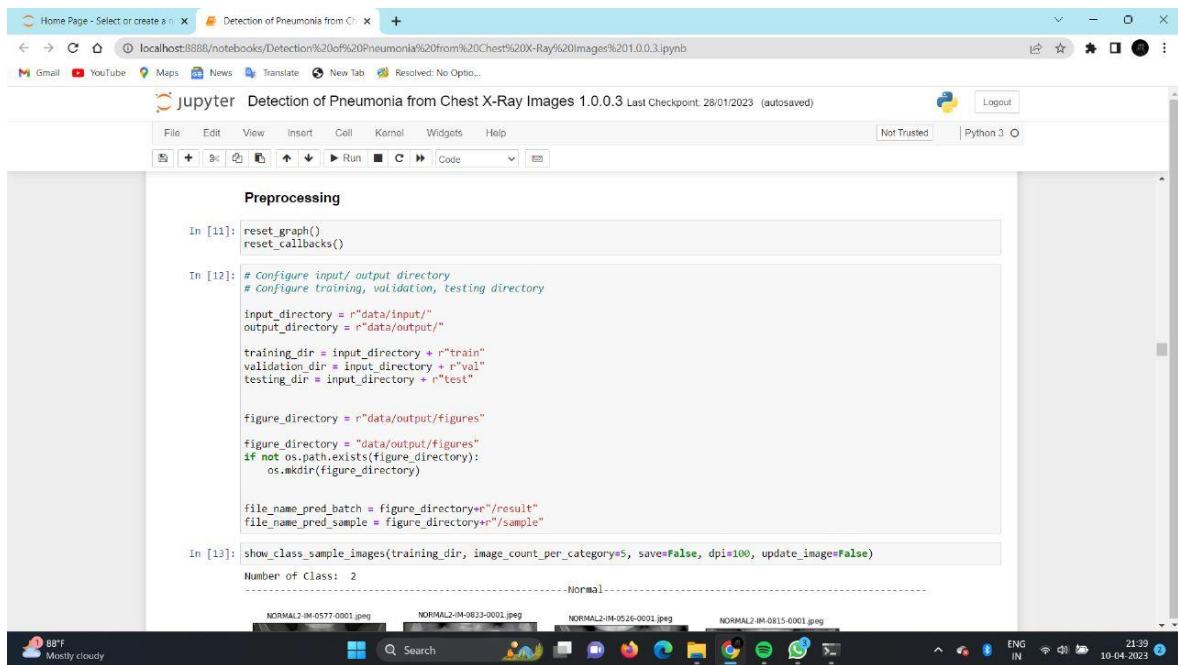


Fig 3 : Data Preprocessing

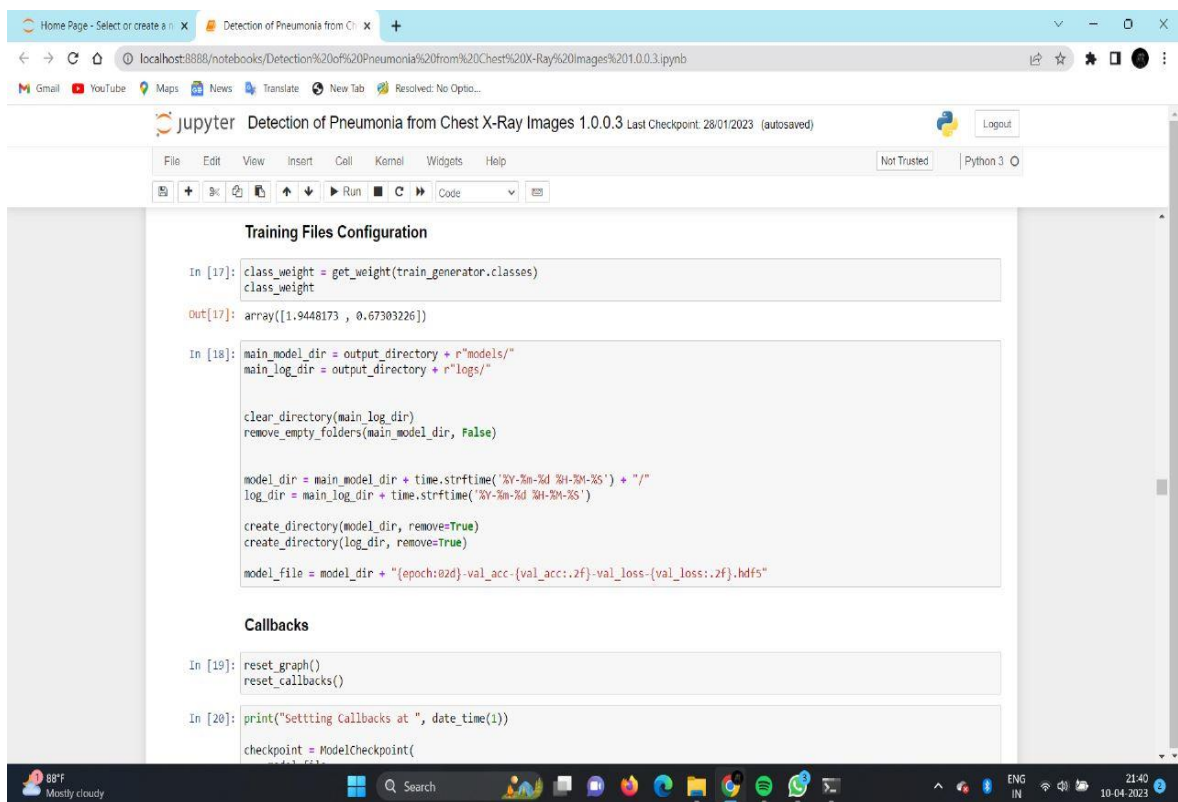


Fig 4 : Training files configuration

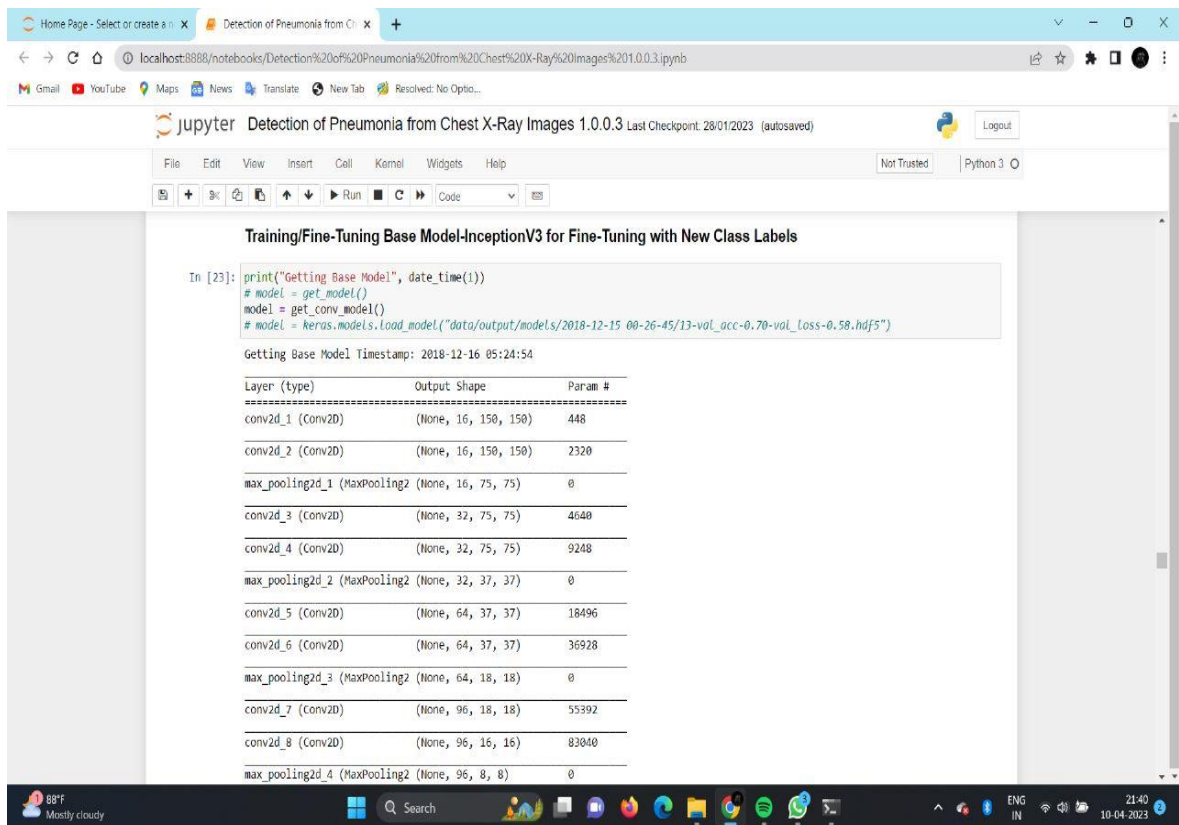


Fig 5: Fine tuning with new class labels

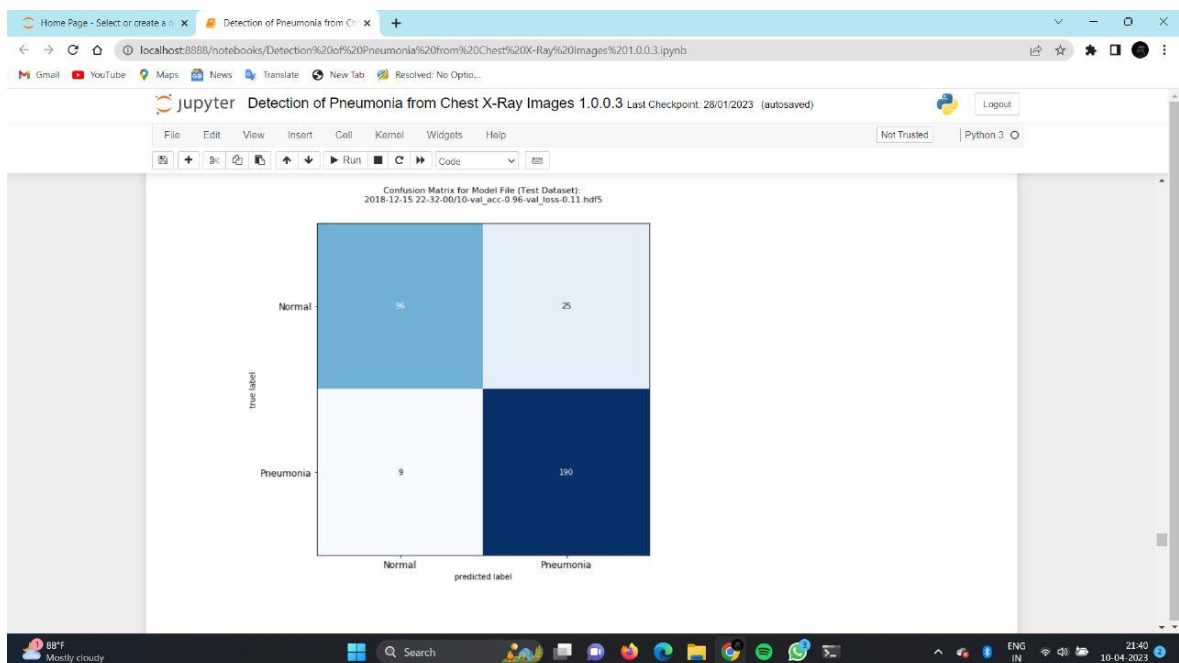


Fig 6: Confusion Matrix

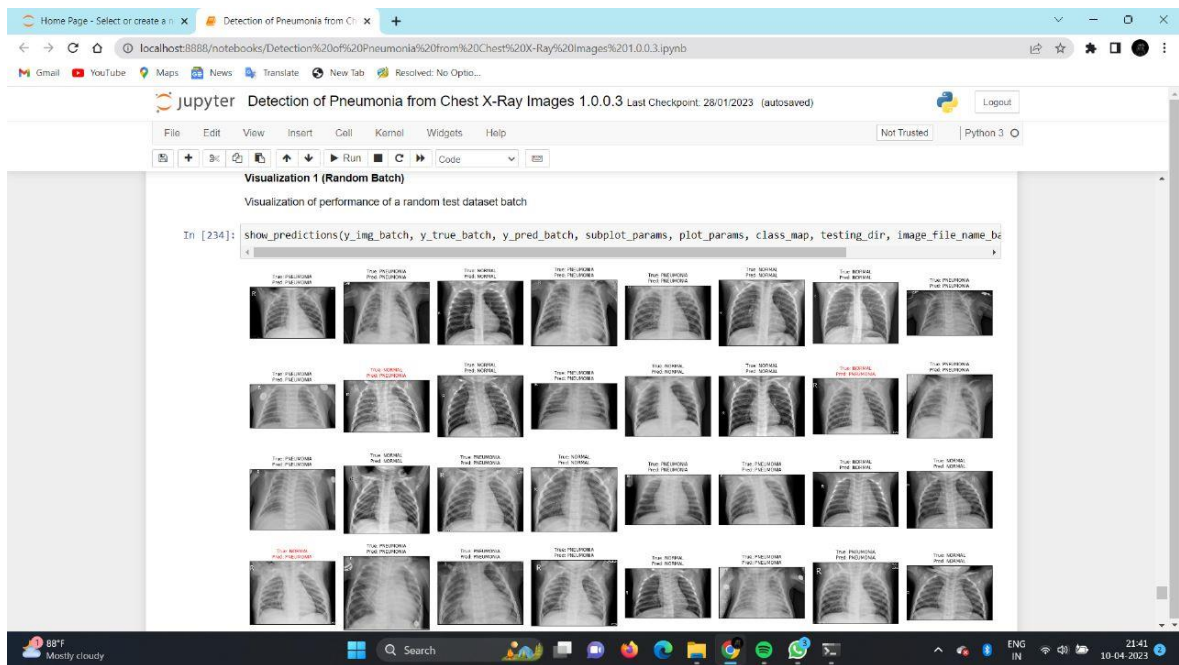


Fig 7: Visualisation 1

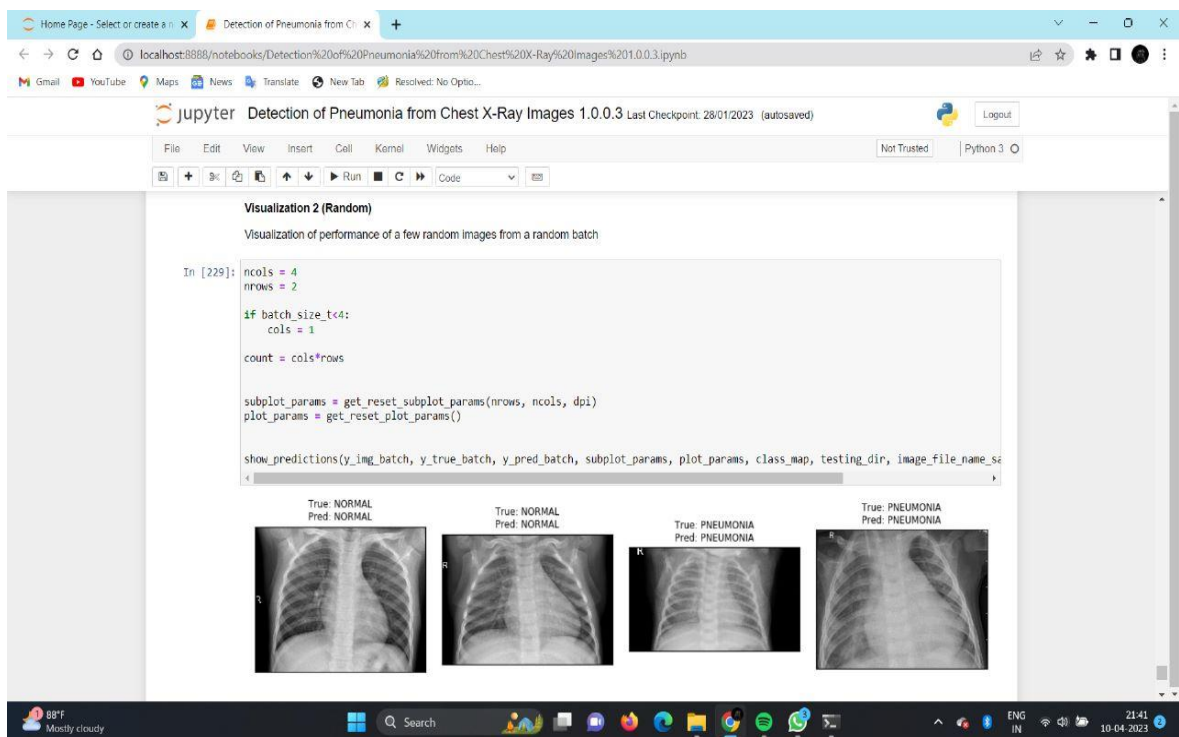


Fig 8: Visualisation 2

APPENDIX - C

RESEARCH PAPER

TO DETECT PNEUMONIA USING CONVOLUTIONAL NEURAL NETWORKS – RECOMMENDER SYSTEM

Arava Jayaram, L.Mary Gladenece

¹ UG Scholar , Sathyabama Institute of Science and Technology, Chennai-119

² Associate Professor, Sathyabama Institute of Science and Technology, Chennai-119

^{1,2} jayaramarava3614@gmail.com, marygladence.it@sathyabama.ac.in

ABSTRACT

Pneumonia is a potentially lethal respiratory illness that is the leading cause deaths in India, according to the World Health organisation. To diagnose pneumonia, chest X-rays must be viewed by qualified radiologists, which might be difficult in remote areas. As a result, developing an automated system for identifying pneumonia can considerably aid in the rapid and effective treatment of the illness Convolutional neural networks had proven to be particularly effects for health images, and their ability to extract features from big datasets makes them suitable for image classification tasks. We investigated the effectiveness of models trained using CNN as information extractors, trailed by different classifiers, in distinguishing b/w abnormal as well as normal X-rays of the chest in this work. We identified which CNN model would best serve this function by analytical investigation. According to our statistical findings, supervised classifier algorithms combined with pre-trained CNN models can be quite useful when assessing chest X-ray images, particularly for detecting pneumonia. After establishing the presence of the condition, the system

can recommend the nearest pneumonia-specific treatment facilities. The use of pre-trained CNN models in an automated system for pneumonia detection can significantly aid in the accurate and timely diagnosis of the condition. Especially in remote areas. This could improve medical care and potentially save lives.

Keywords: convolutional neural network, deep learning, switch learning.

I.INTRODUCTION

It's no wonder that computer vision is a hot subject for artificial intelligence (AI) because of its capacity to deal with a large range of issues that humans encounter in the twenty-first century. AI-assisted medical picture detection is one field of computer vision that has proven to be quite fruitful, with pneumonia being a main cause of death with over 4 million fatalities per year. Pneumonia can present in a variety of ways, including viral, bacterial, or fungal infections, each of which necessitates a unique treatment strategy. Because of their low cost and wide availability, chest X-rays (CXR) or computed tomography (CT) are the most often

used methods for detecting pneumonia. Because of their speed and objective reproducibility of testing, computers can assist human professionals in diagnosing pneumonia, as the symptoms of pneumonia on X-rays are not always obvious to the human eye. Many computer solutions for detecting and classifying pneumonia have been developed, with the COVID-19 pandemic fueling the development of new diagnostic systems. Convolutional Neural Network (cnns), DL had been very successful in this setting, and we will explore some recent significant examples in this field.

A. MOTIVATION

Detecting pleurisy in Images are a key problem in computer vision. The study's primary goal was to clear pneumonia in patients using solely X ray pictures. To determine the existence of pneumonia, doctors often rely on specialised tests. To address this difficult problem, a combination of deep learning models has been designed to simplify medical practitioners' work. "Figure 1" depicts the whole work of the above method.

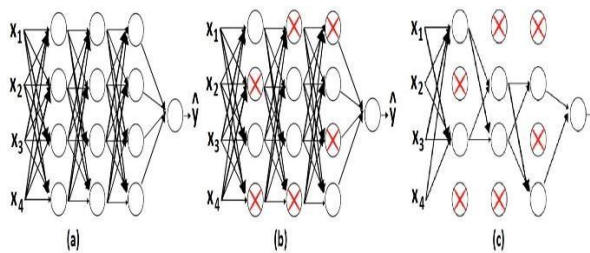


FIG : 1 Taxonomy Of Deep Learning

II. LITERATURE SURVEY

It covers the use of CV, NLP in artificial intelligence (AI) visual content generation. So far, no regenerative models have been developed, and the process instead relies on computer vision and machine translation to generate natural perspectives that eventually describe the image. A Convolutional Neural Networks (C-SNN) and a Recurrent Neural Networks (R-NN) are used in this model. The CNN retrieves visual elements, and the R-NN constructs sentences based on those features. The model is trained to generate captians that show the input image accurately.

On various datasets, the accuracy and fluency of the language model learned from image descriptions were tested. The results of these studies show that the model accurately describes the input image. The automatic description of image content is a critical topic in the field of artificial intelligence, requiring the integration of computer vision to describe images in a natural and objective manner. The model is intended to increase the likelihood of generating an accurate and descriptive image. Experiments on different datasets show that the language model trained only from visual descriptions is accurate and efficient. In terms of both quality and quantity, our model consistently provides high-quality and accurate results. In Paschal's dataset, for example, where the current BLEU-1 score is 25, our technique produces 59, whereas the human value is around 69 (higher is preferable). The Flickr30k dataset improved from 56 to 66, while SBU improved from 19 to

20.. Finally, in the recently released COCO datasets, we achieved a BLEU-4 score of 27.7.

COGNITION, LOCALIZATION AND DETECTION USING CONVOLUTIONAL NETWORKS

We present a comprehensive system for classifying, localising, and detecting objects that makes use of convolutional networks. ConvNet's approach efficiently employs multiscale and scalable techniques. In addition, we offer a novel deep learning technique for predicting object boundaries for object localization. Instead of suppressing constrained suggestions, we cluster them to boost detection confidence. We show how our unified network can tackle several problems at the same time. In Image Net, Large Visual Recognition Challenge (ILSVRC2013), our integrated framework won the localization challenge and achieved highly competed result to detect and classifies tasks. We also incorporated a new cutting-edge detecting feature in later work. Finally, we've launched , a feature extractor for our top model.

EVERY TABLE PREDICTS WHETHER OR NOT IT IS PNEUMONIA

Individuals can write brief descriptions of photographs to highlight their significance. However, automated methods can also be used to complete this task. We offer a system that assigns a score to images that are associated with sentences. This score can be used to describe an image based on a sentence or to find photos

that depict a sentence. The score is determined by comparing the rating from the image to the rating from the sentence. Each estimate is derived using a data-driven discrete estimating process. This technique is tested using a new dataset of tagged photos of humans. Although our estimate is modest, it is adequate to achieve the best quantitative results when compared to the new estimate.

ATTENTION MECHANISM-BASED CNN FOR XRAYS GENERATED:

Pneumonia detection is a fast expanding topic, and computer vision has the potential to influence many domains, including human-computer interaction and emotional computing. In this paper, we present an innovative from beginning to end system featuring a mechanism for attention for identifying severe pneumonia. A feature extracted module, an attention module, a reconstructs module, a segmentation, compose the new network architecture. Local Binary Pattern (LBP) characteristics are used to extract visual data and detect minor motions that can improve network efficiency. A neural network's attention mechanism prioritises relevant features. To improve the model for obtaining better results. we merge LBP and the attention mechanism. There are 5216 trained photos, 624 testing images, and 16 validat images in the dataset. The results of the experiments reveal that the recommended technique is both possible and successful.

THE DISCOVERY OF PNEUMONIA MULTIPLE CONVOLUTION OF THE NEURAL SYSTEM:

Convolutional Neural Network (CNNs) for image categorization had changed the significant process in the field of deep learning. However, detecting real-world diseases is still difficult. We propose using current multi-scale encoders for face expression to address this. Combining characteristics from various depth ranges outperforms using a single depth range model, according to our findings. Furthermore, we show how including rules to remove image noise from the training dataset can improve system performance. The challenge of detecting disease from large-scale imaging remains a major problem in this discipline.

A. INFERENCES FROM LITREATURE SURVEY

There has recently been a lot of buzz about the findings, especially in the field of deep learning. There is a pressing need for contextualised representations of natural language, which may sound far-fetched, but recent advances in disciplines such as neural networks, computer vision, and AP have made it possible to graphically portray an accurate image at a reasonable cost. To accomplish this, we use cutting-edge techniques like as Convolutional Neural Networks (CNN), Re-current Neural Network (R-NN), image datasets with human-readable descriptions. Our research shows that our model performs well in retrieval studies on a variety of datasets.

B. OPEN PROBLEMS IN EXISTING SYSTEM

- The primary goal of this course is to learn about deep learning techniques, notably the JOSS approach for detecting pneumonia.
- We will combine multiple architectures, including CNN and Lacinia, to model our pneumonia detection. The pre-trained Xception model will be used with Rhoncus to extract characteristics from photos..

III. EXISTING POJECT

- However, after conducting a literature review, we discovered that the current model has several limitations.
- It having low accuracy percentage.

IV.REQUIREMENT ANALYSIS A.FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

During this stage, the project's viability is assessed, and a decision is made on the best design and associated cost estimates. During the system analysis phase, it is critical to conduct a feasibility of the current system. This is done to that the planning system will not become a financial or operational burden on the organisation. A solid understanding of the core system requirements is required for conducting a feasibility study. The feasibility assessment evaluates stakeholders' needs and requirements, as well as the possible impact of an information system solution. It seeks to determine the resources required to implement such a solution, as well as its costs and advantages, as well as its overall

viability. The report assesses and offers several alternative options. Feasibility is assessed by analyzing the components of the proposed solution.

B. ECONOMICAL FEASIBILITY

The goal of this study is to assess the financial impact of deploying the system on the organisation. Given the possibility of limited research and development funds, it is critical to justify the system's cost. As a result, the system was created on a shoestring budget, utilising free technologies whenever possible and acquiring only the necessary individual components.

C. ART

The purpose of this study is to show the system's technical feasibility. It is critical to ensure that the system under development does not place undue strain on technological resources. For example, if the client wants to submit a high-resolution photograph, the system should have simple criteria

and
require
few, if
any,
changes
to be
impleme
nted.

Another critical factor is user acceptance of the system. Teaching the user how to utilise the system properly and efficiently is part of the research component. The user should not be concerned with technological aspects, but rather accept the system's necessity.

The purpose of this study is to show the system's technical feasibility. It is critical to ensure that the system under development does not place undue strain on technological resources. For example, if the client wants to submit a high-resolution photograph, the system should have simple criteria

and
require
few, if
any,
changes
to be
impleme
nted.

Another critical factor is user acceptance of the system. Teaching the user how to utilise the system properly and efficiently is part of the research component. The user should not be concerned with technological aspects, but rather accept the system's necessity.

D. SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

HARDWARE

REQUIREMENTS

1. System – Windows
2. Speed - 2.4GHZ
3. Hard disk – 1TB
4. Monitor – 16 VGGA
5. RAM – 4 GB or +

SOFTWARE REQUIREMENTS :

- OS - Windows 10 or +
- Programming language – Python,
- OS - used to system documents using a command device.
- ALEX - Store used NumPy capabilities.
- NumPy - for many mathematical operations on arrays.
- tqdm is a Python library that provides a progress bar for iterators. It can be used as a decorator to wrap an iterator and display the progress of the iteration. By default, it displays the progress on the standard error output.
- VGG16, pre-processing input - Imported module for feature extraction from photograph information.

- load_img, img_to_array - The functions "load_img", "img_to_array" are utilized to an image and change that it into a NumPy array.
- Tokenizer - used to manipulate textual content and convert statistics.
- pad_sequences - used to distribute phrases flippantly in sentences, colour the rest of the gap with zeros
- plot_model - Visualizing the architectural model using various images can be helpful. TensorFlow is a platform that can assist in implementing best practiced model monitoring, monitoring of performance, and model recovery for data automation.
- Keras is a high-level and user-friendly API of TensorFlow 2, which provides a powerful and solution for ML problems, especially in this field of modern deep learning algorithms.

V. PROPOSED SYSTEM

- Traditional lattice models have the potential to cause the issue of vanishing gradients, which reduces the efficiency of recurrent neural networks for learning and analysis. In this paper, we are presenting a new model that enhances the precision for the pneumonia predictions from photos to overcome the problem of gradient descent. This is made possible by the development of deep

learning models and enhanced learning methodologies and APIs. Hospitals may now more accurately and easily identify pneumonia.

A. METHODOLOGY OR PROCESS MODEL - RANGE ALGORITHM

For the classification and recognition of images, convolutional neural network are a popular class of NN. They are employed in a number of disciplines, including face identification, object detection, and scene layout. An image is entered into Rhino, categorised, and processed in accordance with a particular class, such as a dog, cat, lion, tiger, and more.

An image is composed of pixels in a computer, and the resolution of the image determines how these pixels are arranged. $h * w * d$, where h stands for height, w for width, and d for depth, represents the resolution. A grayscale image is organised in a matrix of size $4*4*1$, but an RGB image is organised in a matrix of size $6*6*3$. Each input image flows through a number of convolutional layers in CNN, including collates, layers, and filters (sometimes referred to as kernels). An explanation of the convolution layer is given in Figure 2.

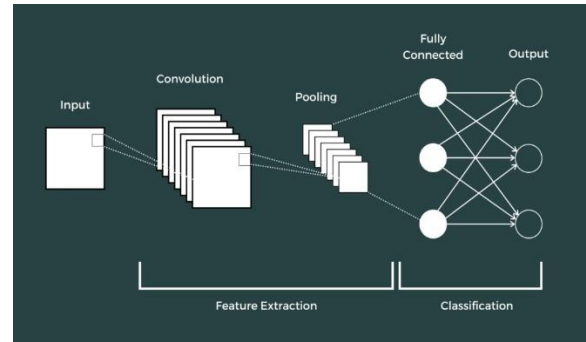


FIG.2 CONVOLUTION LAYER

B. ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The creation of a high-level system and the definition of requirements are necessary for the software's overall functionality. Different pages and their relationships are identified and designed during the architectural design phase. The relationships between modules are developed while the primary software applications are recognised and split into processing techniques and conceptual data structures. These modules are part of the suggested system. In order to extract features from input data, the CNN architecture requires kernel layers. An API is also used to assist sequence prediction. "fig 3" it explains the general architecture of the C-NN model.

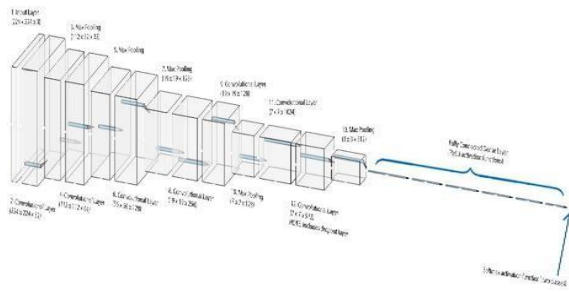


FIG.3 GENERALISED ARCHITECTURE OF THE CNN MODEL

C. SOFTWARE DESCRIPTION FOR IMPLEMENTATION AND EVALUATION PLAN FOR THE PROPOSED MODEL/SYSTEM

1. INPUT DESIGN

A key connection between the user and the data system is provided by the input approach. Its main goal is to define a specification and procedure for getting data ready for the system to use. This can be achieved by manually entering data through keyboards to detect errors or by applying machine learning algorithms to data supplied from printed or written scripts.

An efficient input planning strategy attempts to minimise errors, avoid delays, streamline the process, protect data privacy, and regulate the amount of input needed.

2. OUTPUT DESIGN

Quality is the result that meets the end user's requirements and clearly conveys the information. Any system uses outputs to communicate a process's results to users and other stakeholders. The output strategy outlines the method for delivering information to the point of demand, such as broadcast output. It is the main and most accessible source of user data. System optimisation is facilitated by an efficient and intelligent output system, which also helps the user make judgements.

3.PLAN OF PROPOSED SYSTEM

The vanishing gradient problem can impair the recurrent network's performance in conventional CNN models, making it challenging to train and enhance its accuracy. We suggest a new model that uses deep

learning methods to address this problem and enhance the precision of pneumonia image prediction and CAP diagnosis. The generation of recommendations for hospitals has grown simpler with the use of sophisticated learning algorithms and APIs. “fig 4” it explains the plan of proposed system.

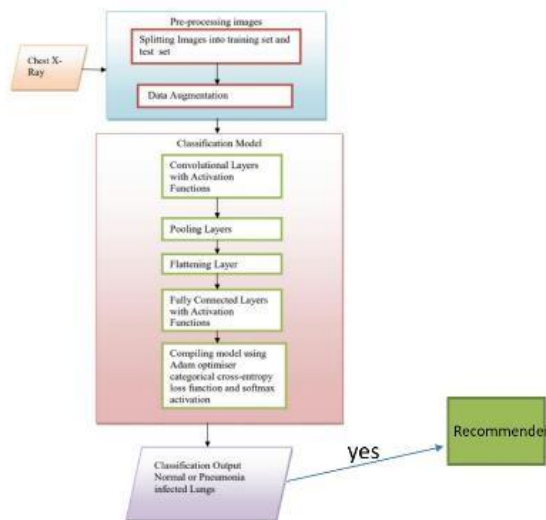
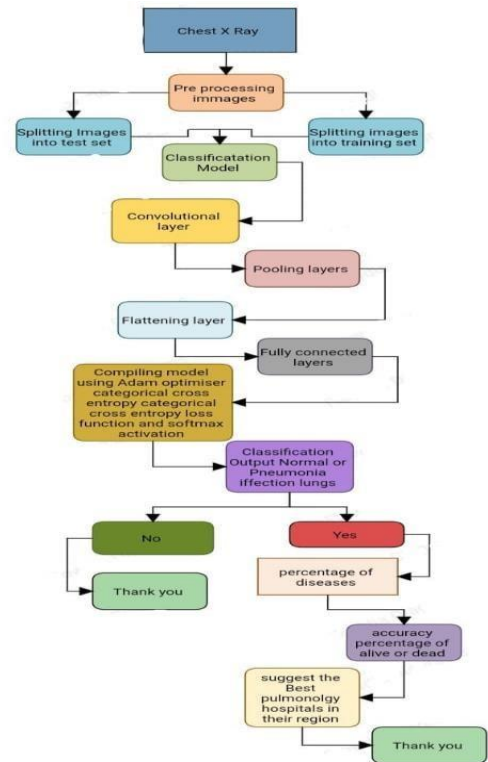


Fig.4 : PROJECT MANAGEMENT PLAN



**Fig 5.
SYSTEM
ARCHITECTURE
4.
PURPOSE**

Deep learning has a hard task of detecting pneumonia and selecting the best hospitals. We will examine various computer vision architectures and APIs in this article in order to identify pneumonia and recommend the best hospitals. Convolutional neural networks (CNNs) can be used to create a model, and APIs (application programming interfaces) can be used to host the model.

VI.RESULT

Deep Learning algorithm is widely used by the pneumonia detection system to forecast the likelihood of pneumonia. Convolutional neural networks (CNNs) are a type of NN that is extensively used in a number of domains including image processing, computer vision, and NLP. The two classes used in the publicly accessible dataset used to test and assess the system's performance are pneumonia disease detection and differentiating between normal and abnormal chest X-rays. CNNs have successfully been applied to both tasks in healthcare and medical diagnosis. The predictions made by the CNN model were used to get the results shown below. Recall is the number of pertinent items that were accurately predicted, whereas precision is the difference of the correct identified positive features to all positive samples. When an image is uploaded, it goes through a number of steps in the CNN network, such as flattening and pooling, to get the crucial feature for prediction. The neural network solves problems with visual resolution as well. The CNN model was applied to two sets of data: normal images and pneumonia-

affected images. Based on these images, the CNN network classified the input images as either affected by pneumonia or not. Metrics including precision, accuracy, and recall were used to measure the CNN model's performance. Table 1 displays the findings, with an F1 score of 85% for normal photos and 92% for images with pneumonia. The precision was 91% for normal photos and 88% for pneumonia-affected images, while the recall was found to be 79% for normal images and 95% for images that had pneumonia. The model's total accuracy was 89%, and since information on each choice it made is presented, even non-experts can understand it. The user will benefit from this model's increased accuracy. Table-1 shows proposed system results.

Table-1

Results of the proposed model

	Precision	Recall	F1-score	Support
Normal	0.91	0.71	0.85	121
Pneumonia	0.95	0.95	0.92	199

Micro-avg	0.95	0.89	0.89	320
Macro-avg	0.95	0.87	0.88	320
Weighted-avg	0.95	0.89	0.89	320

In this section, The results of the implemented pneumonia disease prediction using deep learning algorithms will be presented and discussed. Overall, the pneumonia detection model performs data and the deep learning method that's the projects. Because the reasoning behind each decision is provided. Fig-6 refers to the home page of our pneumonia detection web app

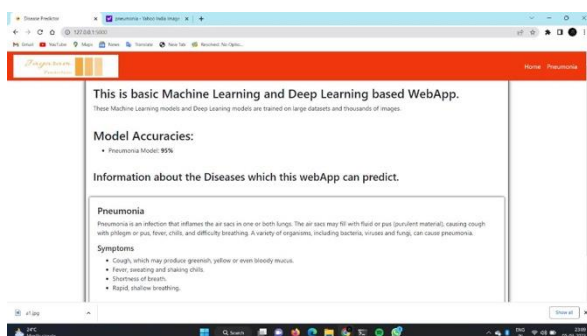


Fig-6 : HOME PAGE

In below image fig-7 it redirects to the Xray

uploader page where Xray's are uploaded

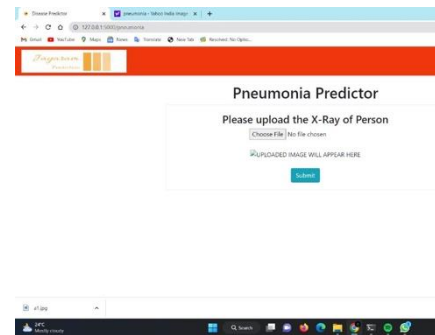


Fig-7 : Model Page

After uploading the Xray the page which is shown in below figure fig-8 it will be redirecting to detection page. Below is the figure of Xray upload in the web application.

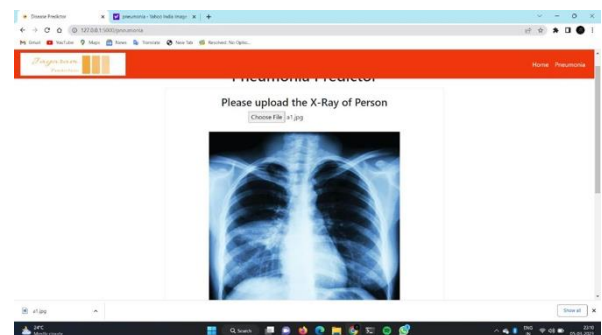


Fig-8 : Xray Uploader Page

Once the Xray got uploaded it redirects to the detection page. Below figure fig-9 refers to the detection page. If the pneumonia is

detected there will be a option named search for nearest hospital referred as below



Fig-9 :
Detection
Page

In the detection page if CNN model detects the pneumonia in uploaded Xray it redirects to the below figure Fig-10 which contains the search for nearest hospital option which automatically searches the best pulmonology

hospital in the operator area. And shows the desired results

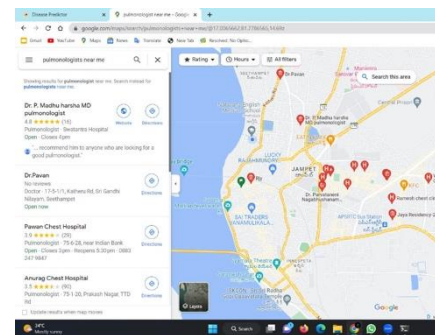


Fig-10 :
Nearest
Hospital
Finder

VII.CONCLUSION

A deep learning algorithm-based pneumonia detection app has been developed, and it shows tremendous promise for predicting if someone would contract pneumonia. Medical personnel will be able to immediately recognise those who have a high risk of contracting pneumonia using this programme and treat them before their condition becomes worse. The pneumonia detection app can also be helpful for people who want to monitor their health for pneumonia and take precautions to lower their chance of getting the ailment. With an overall accuracy rate of more than 95%, the system offers a high degree of prognostic accuracy for pneumonia. There are many deep learning methods that have been used to identify pneumonia. Although some of these algorithms have produced encouraging results, there is still plenty of

room to improve their dependability and accuracy. The test results show that the proposed system is effective at predicting pneumonia. It was specifically created with accuracy and usability in mind. More extensive data sources and more sophisticated deep-learning techniques can be used to increase accuracy even further. The approach has the potential to save lives overall by identifying high-risk patients and empowering medical practitioners to take prompt and essential preventive action. Its capacity to offer precise and prompt pneumonia identification can be a key step towards enhancing public health and lowering the condition's fatality rate.

REFERENCES: -

11. Tilve, A., Nayak, S., Vernekar, S., Turi, D., Shetgaonkar, P.R. and Aswale, S., 2020, February. Pneumonia detection using deep learning approaches. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)* (pp. 1-8). IEEE.
12. Mahmud, T., Rahman, M.A. and Fattah, S.A., 2020. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in biology and medicine*, 122, p.103869.
13. Kundu, R., Das, R., Geem, Z.W., Han, G.T. and Sarkar, R., 2021. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PloS one*, 16(9), p.e0256630.
14. Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ. Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurement*. 2020 Dec 1;165:108046.
15. Barhoom, A.M. and Abu-Naser, S.S., 2022. Diagnosis of pneumonia using deep learning.
16. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M.P.: Chexnet: Radiologist-Level Pneumonia Detection on Chest X-rays with Deep Learning (2017). arXiv preprint arXiv:1711.05225
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
18. 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.
19. 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.

20. Pan, Ian, Alexandre Cadrin-Chênevert, and Phillip M. Cheng. "Tackling the radiological society of North America pneumonia detection challenge." *American Journal of Roentgenology* 213, no. 3 (2019): 568-574.