GENERATING CHEST X-RAY PROGRESSION OF PNEUMONIA USING CONDITIONAL CYCLE GENERATIVE ADVERSARIAL NETWORKS

A Mini-Project Report submitted to
Jawaharlal Nehru Technological University
in partial fulfillment of the requirements for the award of Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

Submitted by

GANGAPURAM NANDINI **22WJ8A05Z3**

Under the Guidance of
Mrs. A. Roja Ramani
Assistant Professor

Department of Computer Science & Engineering
School of Engineering & Technology
Guru Nanak Institutions Technical Campus
Ibrahimpatnam, Hyderabad, R.R. District – 501506
November, 2025

CERTIFICATE

This is to certify that the Mini-project entitled **"GENERATING CHEST X-RAY PROGRESSION OF PNEUMONIA USING CONDITIONAL CYCLE GENERATIVE ADVERSARIAL NETWORKS"** is being submitted by **Ms. GANGAPURAM NANDINI**, bearing Roll No. **22WJ8A05Z3**, in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Jawaharlal Nehru Technological University is a record of Bonafide work carried out by him under my guidance and supervision.

The results embodied in this Mini-project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

Internal Guide	**Project Coordinator**
Mrs.A.Roja Ramani	**Dr. N. Naveen Reddy**
Assistant Professor	**Assistant Professor**

Head of the Department

Dr. M V Narayana

RAM Innovative Infotech
M : +91 9581 012 012
E : raminnovativeinfotech@gmail.com
Flat No.#309, Amrutha Ville,
Opp: Yashoda Hospital, Somajiguda,
Hyderabad-82, Telangana, India
Www.raminnovativeinfotech.webs.com

PROJECT COMPLETION CERTIFICATE

This is to certify that the following students of 3rd year B. Tech, Department of **Computer Science and Engineering** – Guru Nanak Institutions Technical Campus have completed their training and Mini-Project at GNITC successfully.

STUDENT NAME	ROLL NO
GANGAPURAM NANDINI	22WJ8A05Z3

The training was conducted on **Python** Technology for the completion of the Mini-Project titled **"GENERATING CHEST X-RAY PROGRESSION OF PNEUMONIA USING CONDITIONAL CYCLE GENERATIVE ADVERSARIAL NETWORKS"** in **October 2025.** The Mini-Project has been completed in all aspects.

INNOVATIVE INFO HYD

Signature

Declaration of Student

I, GANGAPURAM NANDINI (22WJ8A05Z3), hereby declare that the Mini- Project titled "GENERATING CHEST X-RAY PROGRESSION OF PNEUMONIA USING CONDITIONAL CYCLE GENERATIVE ADVERSARIAL NETWORKS " has been carried out by me as part of the requirements for the award of the Degree of Bachelor of Technology in the Department of Computer Science & Engineering at Guru Nanak Institutions Technical Campus.

We confirm the following:

1. The Mini-Project was undertaken by us under the supervision of our guide, Mrs. Roja Ramani , from the selection of the topic to the completion of the final report.
2. I have ensured that the results presented in the report are accurate and based on our original work.
3. To the best of our knowledge, the content of this report is free from plagiarism and adheres to ethical standards.
4. The entire Mini-Project work out has been conceptualized ,implemented and documented by me.
5. The Mini-Project report has been prepared with diligence, ensuring clarity, accuracy, and adherence to academic standards.

I further declare that this report has not been submitted, in part or full, to any other institution or university for the award of any degree or diploma.

GANGAPURAM NANDINI
22WJ8A05Z3 Signature

Date: 01/11/2025
Place: Ibrahimpatnam

Declaration of Guide

I, Mrs. A. Roja Ramani, hereby declare that I have guided the Mini-Project titled "GENERATING CHEST X-RAY PROGRESSION OF PNEUMONIA USING CONDITIONAL CYCLE GENERATIVE ADVERSARIAL NETWORKS" undertaken by GANGAPURAM NANDINI (22WJ8A05Z3).This Mini-Project was carried out towards the fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering at Guru Nanak Institutions Technical Campus.

As the guide, I confirm the following:

1. I have overseen the entire project process, from the selection of the project title to the submission of the final report.
2. I have reviewed and certified the accuracy and relevance of the results presented in the report.
3. The contributions of the student have been appropriately recognized and assessed.
4. The Mini-Project report has been prepared under my supervision, ensuring adherence to high standards of quality, clarity, and structure.

I further certify that this Mini-Project report has not been previously submitted in part or full for the award of any degree or diploma by any institution or university.

Name of Guide**:Mrs.A.RojaRamani** Signature of the Guide

Date: 01/11/2025

Place: Ibrahimpatnam

Name of HOD: **Dr. M V Narayana** Signature of the HOD

Department: Computer Science & Engineering

Date: 01/11/2025

Place: Ibrahimpatnam

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to our internal guide, **Mrs.A.Roja Ramani**, Assistant Professor, Department of Computer Science & Engineering, for her valuable guidance, encouragement, and continuous support throughout the duration of this Mini-Project.

I also thankful to **Dr. M V Narayana**, Professor and HoD, Computer Science & Engineering Department, for his expert supervision and helpful suggestions, which contributed significantly to the successful completion of this project.

I would also like to thank the faculty members of the **Computer Science & Engineering Department** and the Lab Technicians for their assistance and cooperation during the practical work of our project.

I grateful thank to my friends and well-wishers for their encouragement, collaboration, and useful feedback throughout the project journey.

Lastly, I sincerely thank my parents for their constant support, patience, and motivation, which helped me to complete this Mini-Project successfully.

GANGAPURAM NANDINI 22WJ8A05Z3

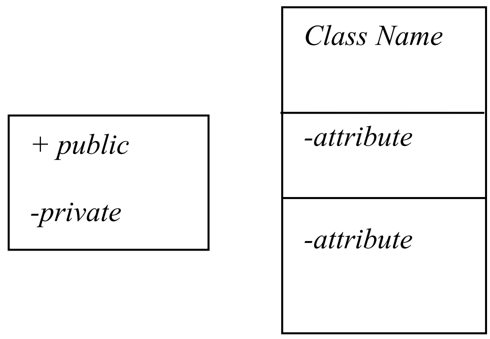
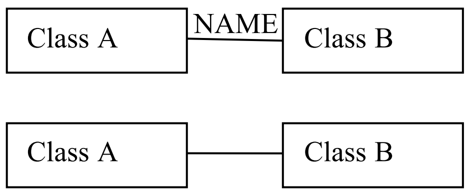
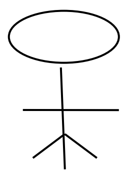
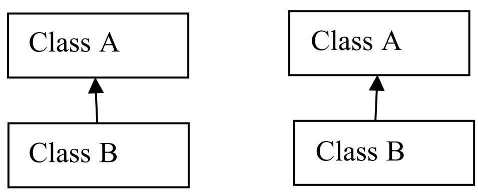
ABSTRACT

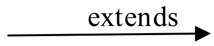
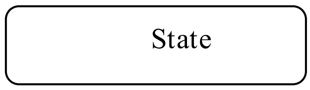
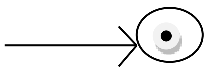
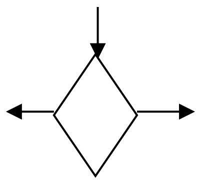
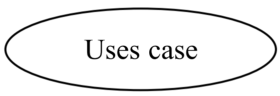
Pneumonia is a severe inflammation of the lungs caused by pathogens or autoimmune diseases, affecting approximately 450 million individuals worldwide annually. Chest X-ray analysis is the primary method for diagnosing pneumonia, and with the growth of deep learning technologies, there is now a surge of high-dimensional image, audio, and video data being utilized in the medical field. This paper explores the use of Generative Adversarial Networks (GANs) and its extension, Conditional GANs (cGANs), to aid in the diagnosis and progression monitoring of pneumonia. We leverage GANs to perform domain adaptation, transforming images between pneumonia and normal X-ray images. Our framework applies a ResNet152-based classifier to evaluate the change between these domains and generates intermediate images that depict the progression of pneumonia. By feeding conditional vectors to the GANs generator, we simulate various stages of pneumonia progression. We evaluate the performance of the model by comparing the generated images to the original dataset, showing the GAN's potential to generate realistic and plausible progression images for medical use.

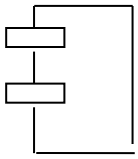
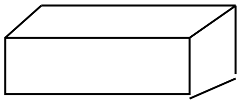
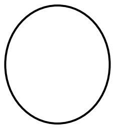
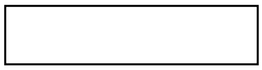
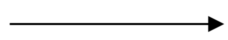
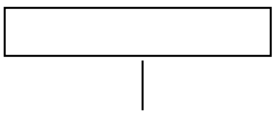
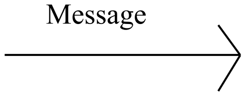
TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.

14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

CHAPTER-1

INTRODUCTION

1.1 GENERAL

Pneumonia is a life-threatening lung infection that affects millions of people worldwide each year. Traditional diagnostic methods, such as chest X-ray analysis, are widely used in medical practice to detect pneumonia; however, interpreting these images requires expertise and can sometimes lead to misdiagnosis. With the advancement of deep learning, artificial intelligence (AI) has become a powerful tool in medical imaging, enabling automated diagnosis and enhancing decision-making processes. One of the most promising AI techniques in this field is Generative Adversarial Networks (GANs), which have demonstrated their ability to generate realistic medical images for training, augmentation, and simulation purposes. This research explores the application of GANs and their variant, Conditional GANs (cGANs), to improve pneumonia detection and progression monitoring. By transforming chest X-ray images between normal and pneumonia-affected states, GANs facilitate domain adaptation, allowing for a better understanding of disease progression. Our model incorporates a ResNet152-based classifier to evaluate generated images and assess the effectiveness of transformation techniques. By conditioning the GAN with specific vectors, we simulate various stages of pneumonia, which can help doctors visualize how the infection develops over time. The ability to generate intermediate disease progression images provides a novel approach to training radiologists and improving diagnostic accuracy. This technique could also assist in predicting patient outcomes by analyzing disease severity based on generated images. Furthermore, by leveraging deep learning, this study contributes to advancing AI-driven medical diagnostics and bridging the gap between AI and clinical applications. The integration of GANs in pneumonia diagnosis has the potential to revolutionize medical imaging, reduce dependency on large annotated datasets, and enhance patient care. With further improvements, this approach could extend beyond pneumonia to other respiratory conditions, fostering innovation in AI-assisted healthcare solutions.

1.2 SCOPE OF THE PROJECT

The scope of this project revolves around utilizing deep learning, particularly Generative Adversarial Networks (GANs) and Conditional GANs (cGANs), to enhance pneumonia diagnosis and progression monitoring through chest X-ray images. The project aims to generate realistic medical images to assist healthcare professionals in disease detection, training, and patient monitoring. By leveraging domain adaptation, the model can transform normal chest X- rays into pneumonia-affected ones, enabling better visualization of disease development. Additionally, the project focuses on integrating a ResNet152-based classifier to evaluate image transformations and improve diagnostic accuracy. The generated progression images can serve as a valuable tool for medical practitioners, offering insights into different stages of pneumonia. Furthermore, this approach has the potential to be extended to other respiratory diseases and medical imaging applications, ultimately improving patient care, medical training, and clinical decision-making.

1.3 OBJECTIVE

The objective of this project is to develop an AI-driven framework that leverages GANs and cGANs to enhance pneumonia diagnosis and progression monitoring. By utilizing deep learning techniques, the project aims to generate realistic medical images that can simulate different stages of pneumonia, thereby aiding doctors and researchers in understanding disease development. The framework incorporates a ResNet152-based classifier to analyze and validate the generated images, ensuring reliability in medical applications. This project also seeks to improve medical training by providing AI-generated datasets that can support learning and research. Furthermore, the generated images can be used to augment existing medical datasets, reducing the need for large-scale annotated data. The proposed model is designed to improve the accuracy of pneumonia detection, enhance clinical decision-making, and ultimately contribute to better patient care. By extending this methodology to other respiratory diseases, the project aims to establish AI-driven solutions as an integral part of modern healthcare diagnostics.

1.4 EXISTING SYSTEM

In the existing systems for pneumonia detection and progression monitoring, traditional image classification techniques are employed, often relying on manual labeling and human expertise for diagnosis. Convolutional Neural Networks (CNNs) are frequently used to analyze chest X-ray images to classify normal and pneumonia-infected images. These systems typically focus on binary classification tasks and have demonstrated success in detecting pneumonia in X- ray images. However, challenges remain, particularly in generating intermediate progression stages or simulating the dynamic changes in the disease over time. Moreover, these models are often limited to the analysis of a single snapshot of the disease, lacking the ability to predict the future state or progression. Therefore, there is a need for more advanced methods that can generate realistic, intermediate images that simulate the disease's progression in a dynamic and continuous way.

1.4.1 EXISTING SYSTEM DISADVANTAGES

- Limited to binary classification tasks
- No generation of intermediate progression images
- Relies heavily on human expertise for diagnosis
- Lack of prediction for disease progression
- Requires a large number of labeled data for training

1.5 PROPOSED SYSTEM

The proposed system introduces a novel approach using GANs and Conditional GANs to generate intermediate domain images between pneumonia and normal chest X-ray images. This method involves training GANs to map between two domains (normal and pneumonia) and then employing conditional vectors to generate progression images at different stages of pneumonia. The framework applies a ResNet152-based classifier to evaluate the performance of these generated images, ensuring their quality and realism. By utilizing the flexibility of GANs, the system not only detects pneumonia but also simulates its progression, enabling a more comprehensive understanding of the disease. This method offers significant improvements over traditional diagnostic approaches by providing a way to visualize the dynamic changes in pneumonia, facilitating more informed clinical decisions and better monitoring of the disease.

1.5.1 PROPOSED SYSTEM ADVANTAGES

- Generates realistic and plausible images for pneumonia progression
- Enables domain adaptation between pneumonia and normal images
- Enhances diagnostic accuracy by visualizing disease progression
- Reduces the need for labeled data through image generation
- Improves the overall understanding of pneumonia evolution in patients

CHAPTER-2

LITERATURE SURVEY

2.1 GENERAL

1.Title: Lung Disease Detection from Chest X-Ray Using GANs
Author: Richa Sharma, Monika Mangla, Sharvari Patil, Priyanca Gonsalves, Neha Agarwal
Year: 2024

Description: Lung diseases such as pneumonia, tuberculosis (TB), chronic obstructive pulmonary disease (COPD), and COVID-19 cause serious lung damage. According to the World Health Organization, long-term illnesses account for upto 10.38% of the total mortality rate in India. The early and accurate diagnosis of these diseases can have a significant impact on the lives of patients, as a delayed diagnosis prevents counselling. As a result, early discovery is critical for human survival, necessitating the use of innovative techniques and cutting-edge technologies to accelerate recovery and enhance long-term survival rates. To solve this issue, Generative Adversarial Networks (GAN) and Convolutional Neural Networks (CNN) are used to create an automated method for detecting and classifying various lung diseases. This study intends to construct a comprehensive framework to effectively recognize and differentiate various lung diseases by leveraging the capabilities of GANs and CNNs. The proposed system will deliver categorization results that will be supplemented by explanations based on Explainable AI techniques. The proposed strategy ensures transparency and provides insights into the system's decision-making process. The proposed system enables the early detection of lung diseases with minimal effort and time. The power of GANs, CNNs, and XAI can be used to improve the efficiency and accuracy of identifying lung diseases, thereby improving outcomes and long-term survival rates.

2.Title: Diagnosis of Pneumonia from Chest X-Ray Images using Transfer Learning and Generative Adversarial Network

Author: Shekofeh Yaraghi, Farhad Khosravi

Year: 2024.

Description: Pneumonia is a life threatening disease, which occurs in the lungs caused by either bacterial or viral infection. A person suffering from pneumonia has some symptoms including cough, fever and chills, dyspnea, and low energy and appetite. The symptoms will worsen and it can be life endangering if not acted upon in the right time. Pneumonia can be diagnosed using various methods and devices, such as blood tests, sputum culture , and various types of imaging, but the most common diagnostic method is chest X-ray imaging. According to the progress achieved in the diagnosis of pneumonia, there are some problems such as the low accuracy of the diagnosis. Hence the purpose of this article is to diagnose pneumonia from chest x-ray images using transfer learning and Generative Adversarial Network (GAN) with high accuracy in two groups of normal and Pneumonia and then diagnose the type of disease in three groups: normal, viral pneumonia and bacterial pneumonia. The dataset of the article contains 5856 chest X-ray images, including normal images, viral pneumonia and bacterial pneumonia. Adversarial generator network was used in order to increase the data volume and accuracy of diagnosis. Two different pre-trained deep Convolutional Neural Network (CNN) including DenseNet121 and MobileNet, were used for deep transfer learning. The result obtained in dividing into two classes, normal and pneumonia, using DenseNet121 and MobileNet, reached an accuracy of 0.99, which is improved compared to the previous method. Therefore, the results of proposed study can be useful in faster diagnosing pneumonia by the radiologist and can help in the fast screening of the pneumonia patients.

3. Title: Diagnosis of Pneumonia from Chest X-ray Images with Vision Transformer Approach

Author: Emrah Aslan

Year: 2024.

Description: People can get pneumonia, a dangerous infectious disease, at any time in their lives. Severe cases of pneumonia can be fatal. A doctor would usually examine chest x-rays to diagnose pneumonia. In this work, a pneumonia diagnosis system was developed using publicly available chest x-ray images. Vision Transformer (ViT) and other deep learning models were used to extract features from these images. Vision Transformer (ViT) is an attention-based model used for image processing and understanding as an alternative to the convolutional neural networks traditionally used for this purpose. ViT consists of a series of attention layers, where each attention layer models the relationships between input pixels to represent an image. These relationships are determined by a set of attention heads and then fed into a classifier. ViT performs effectively in a variety of visual tasks, especially when trained on large datasets. The study shows that the ViT model's classification procedure has a high success rate of 95.67%. These results highlight how deep learning models can be used to quickly and accurately diagnose dangerous diseases such as pneumonia in its early stages. The study also shows that the ViT model outperforms current approaches in the biomedical field.

4.Title: Enhancing Pneumonia Detection in Chest X-Rays: A Combined GAN and CNN Approach

Author: P. AnnanNaidu, A. Abhilasha, N. Manikanta, T. Mukesh Tilak

Year: 2024

Description : Recent advances in Deep Learning have led to the development of supervised models to detect anomalies in medical images such as pneumonia in chest X-rays. Automatic detection of such anomalies can help clinicians with faster decision making and treatment planning for patients. Nonetheless, supervised models require complete labeled training data with all possible labels (i.e., positive and negative), which are cumbersome and expensive to obtain. We propose an adversarial learning-based semi-supervised algorithm for anomaly detection, which requires training data only with a single class (positive or negative). We applied our proposed Generative Adversarial Network architecture to detect anomalies and score pneumonia in chest X-rays and achieved statistically significant improvements compared to previous state-of-the-art generative network and one-class classifiers for anomaly detection.

5.Title: Pneumonia Detection in Chest X-Ray Images: Handling Class Imbalance

Author: Wardah Ali, Eesha Qureshi, Omama Ahmed Farooqi, Rizwan Ahmed Khan

Year: 2023**.**

Description: Pneumonia affects individuals globally, with the highest mortality rates in Sub-Saharan Asia and South Asia. Despite effective vaccines and antibiotics, the incidence and mortality rates of pneumonia have escalated in recent years, necessitating swift prevention and treatment strategies. A significant challenge in the artificial intelligence research community is the scarcity of publicly available datasets for chest diseases, including pneumonia. Additionally, existing datasets often exhibit severe class imbalance, with normal cases oversampled and pneumonia cases in the minority, complicating the problem further. This study introduces a novel framework for pneumonia detection that addresses the class imbalance issue. The proposed methodology employs a combination of Deep Convolutional Generative Adversarial Networks (DCGAN) and Wasserstein GAN with gradient penalty (WGAN-GP) for augmenting the minority class ("Pneumonia"), while applying Random Under-Sampling (RUS) to the majority class ("No Findings") to balance the dataset. The ChestX-Ray8 dataset, one of the largest available, is utilized to validate the performance of the proposed framework. Transfer learning is applied using state-of-the-art deep learning models, including ResNet-50, Xception, and VGG-16. The results obtained surpass current state-of-the-art performance.

CHAPTER 3

DESIGN AND DEVELOPMENT

3.1 PROJECT DESCRIPTION

3.1.1 GENERAL

Pneumonia is a critical respiratory disease that affects millions of individuals worldwide, requiring accurate diagnosis and continuous monitoring for effective treatment. Traditional methods rely on chest X-ray analysis, which demands expertise and time. This project introduces an AI-driven approach utilizing Generative Adversarial Networks (GANs) and Conditional GANs (cGANs) to generate realistic medical images that simulate the progression of pneumonia. The framework employs domain adaptation techniques to transform normal X-ray images into pneumonia-affected ones, helping in disease visualization and diagnosis. To ensure high accuracy, the project integrates a ResNet152-based classifier that evaluates the quality and reliability of the generated images. By feeding conditional vectors into the GAN model, the system can produce different stages of pneumonia, allowing healthcare professionals to study the disease's development in a controlled manner. The generated images serve as a valuable tool for medical training, dataset augmentation, and AI-assisted diagnosis. This project not only enhances pneumonia detection but also contributes to the broader field of AI in healthcare, offering a scalable solution that can be adapted for other respiratory diseases. Through deep learning techniques, the proposed framework aims to improve diagnostic accuracy, reduce the need for large annotated datasets, and support clinical decision-making.

3.2 METHODOLOGIES

3.2.1 MODULES NAME

- Collecting Data
- Analyzing the information
- Preprocessing Data
- Running the model
- Fine Tuning the model
- Model Efficiency
- Forecasting Results

3.2.2 MODULES EXPLANATION

Collecting Data:

This module involves gathering a comprehensive dataset of chest X-ray images, including both normal and pneumonia-affected cases. The data is sourced from publicly available medical databases, hospitals, and research institutions. Ensuring a diverse dataset is crucial for training a robust deep learning model. Proper data labeling and annotation are also carried out to enhancemodel accuracy.

Analyzing the Information:

In this stage, the collected data is examined to identify patterns, correlations, and potential biases. Statistical techniques and visualization tools are used to gain insights into data distribution. Understanding the dataset helps in refining the model architecture and determining relevant features. Data analysis ensures that the model learns from high-quality, well-balanced information.

Preprocessing Data:

This module focuses on cleaning and transforming the raw data to make it suitable for model training. Techniques like image resizing, normalization, noise reduction, and data augmentation are applied to improve the quality of X-ray images. Removing inconsistencies and ensuring uniformity in the dataset helps in achieving better model performance.

Running the Model:

The deep learning model, incorporating GANs and ResNet152, is executed using the preprocessed data. The GANs generate synthetic pneumonia images, while the classifier evaluates their realism. The model learns to differentiate between normal and pneumonia-affected X-rays, enabling automated diagnosis. This phase involves multiple training iterations to optimize performance.

Fine-Tuning the Model:

Once the model is trained, fine-tuning is done to improve accuracy and reliability.

Hyperparameters such as learning rate, batch size, and number of epochs are adjusted for optimal results. Transfer learning techniques may be applied to enhance model generalization. The fine- tuning process helps in reducing false positives and negatives.

Model Efficiency:

The efficiency of the trained model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score. The generated images are compared with real medical data to assess their quality. Model robustness is tested on unseen datasets to verify generalization capabilities. Efficient models help in faster and more reliable pneumonia diagnosis.

Forecasting Results:

The final stage involves utilizing the trained model to predict pneumonia progression in patients. By generating intermediate images of the disease at different stages, the model provides valuable insights for medical professionals. The forecasting results assist in early diagnosis, treatment planning, and decision-making. The system’s predictions can be integrated into healthcare applications for real-time monitoring.

3.3 TECHNIQUE USED OR ALGORITHM USED

3.3.1 EXISTING TECHNIQUE

The existing algorithm primarily relies on Convolutional Neural Networks (CNNs) for image classification tasks, specifically for detecting pneumonia in chest X-ray images. CNNs are deep learning models that excel at image-related tasks by automatically learning spatial hierarchies of features. In the case of pneumonia detection, CNNs are trained to distinguish between normal and pneumonia-infected lung images. These networks typically consist of convolutional layers, pooling layers, and fully connected layers, designed to process image data and produce a classification output. The existing systems generally rely on manual data annotation and do not focus on generating intermediate images or modeling disease progression.

3.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED

The proposed system employs a combination of Cycle GANs and Conditional GANs to generate intermediate progression images of pneumonia from chest X-ray data. GANs are composed of two neural networks: a generator and a discriminator, which work in opposition to each other. The generator creates synthetic images, while the discriminator evaluates whether the images are real or fake. Conditional GANs extend GANs by adding conditional information to the generation process, allowing for more controlled output, such as generating specific stages of pneumonia progression. In this framework, the generator receives a conditional vector to generate pneumonia progression images at different stages between normal and infected images. This method leverages the flexibility of GANs to simulate the continuous progression of pneumonia and improve the realism and usefulness of generated medical images.

3.4 REQUIREMENTS ENGINEERING

3.4.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, ourresults are very competitive.

3.4.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are sed by software engineers as the starting point for the system design. It should

what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 500 GB

3.4.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

3.4.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture..

3.4.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows:

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

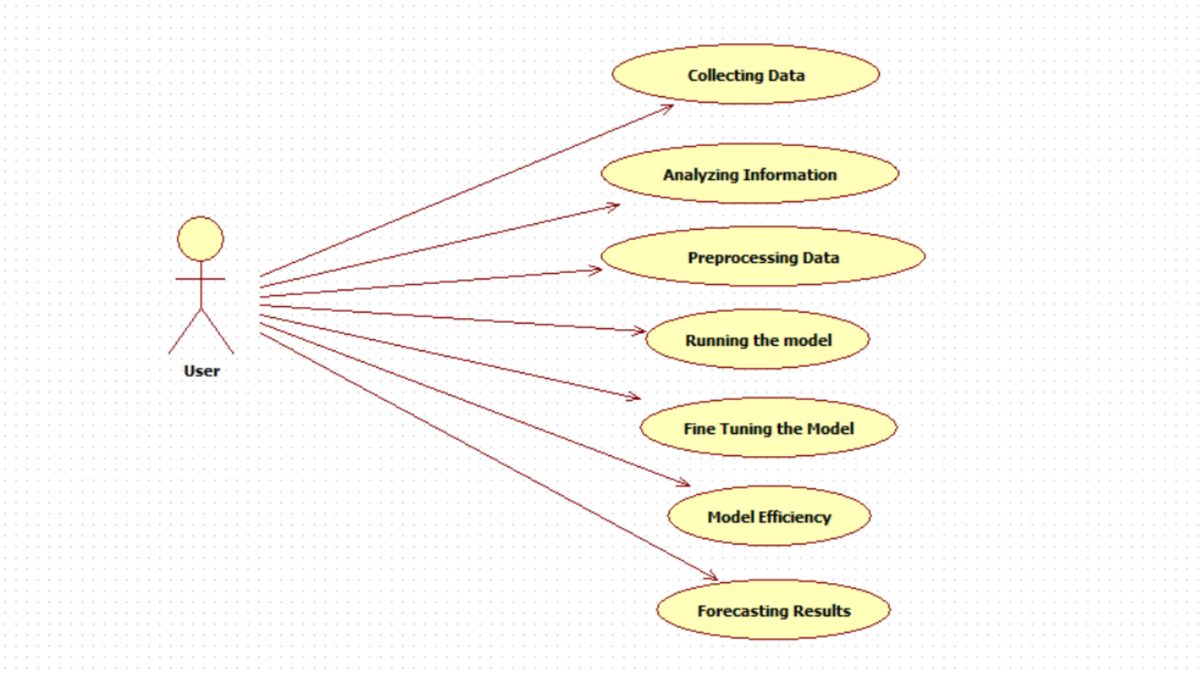
3.5 DESIGN ENGINEERING

3.5.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

3.5.2 UML DIAGRAMS

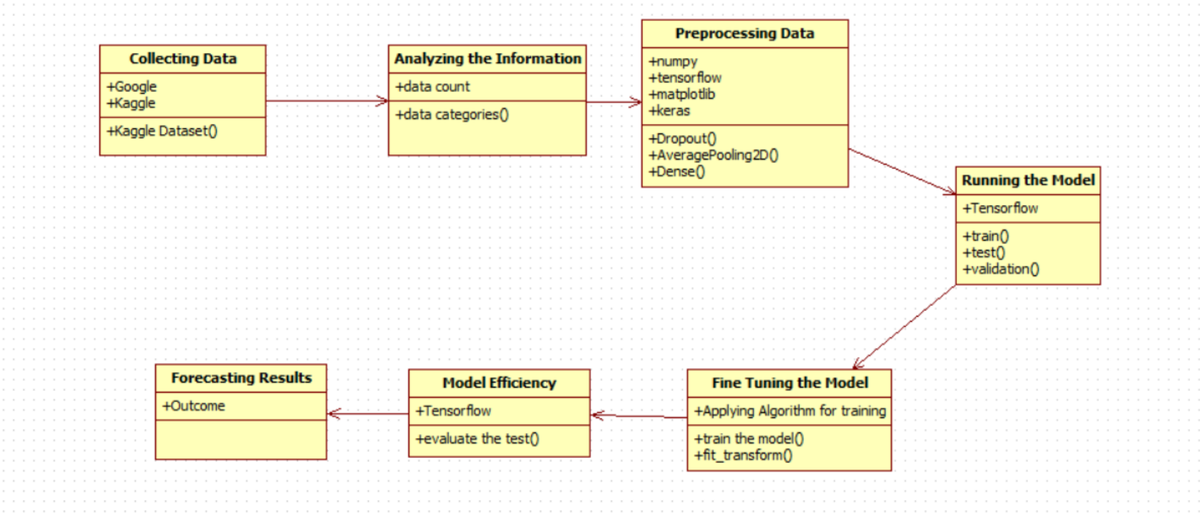
3.5.2.1 USE CASE DIAGRAM



EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

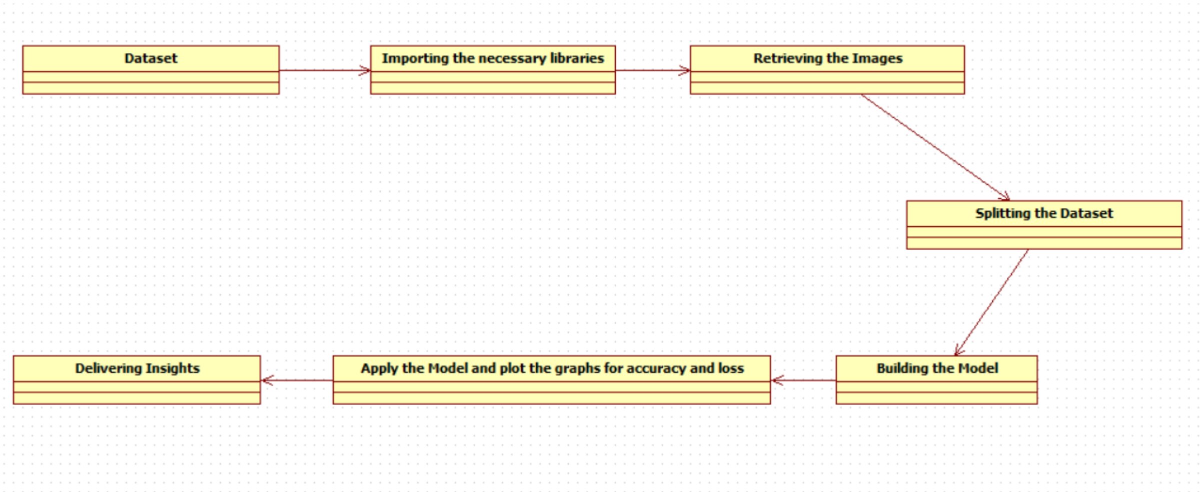
3.5.2.2 CLASS DIAGRAM



EXPLANATION:

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

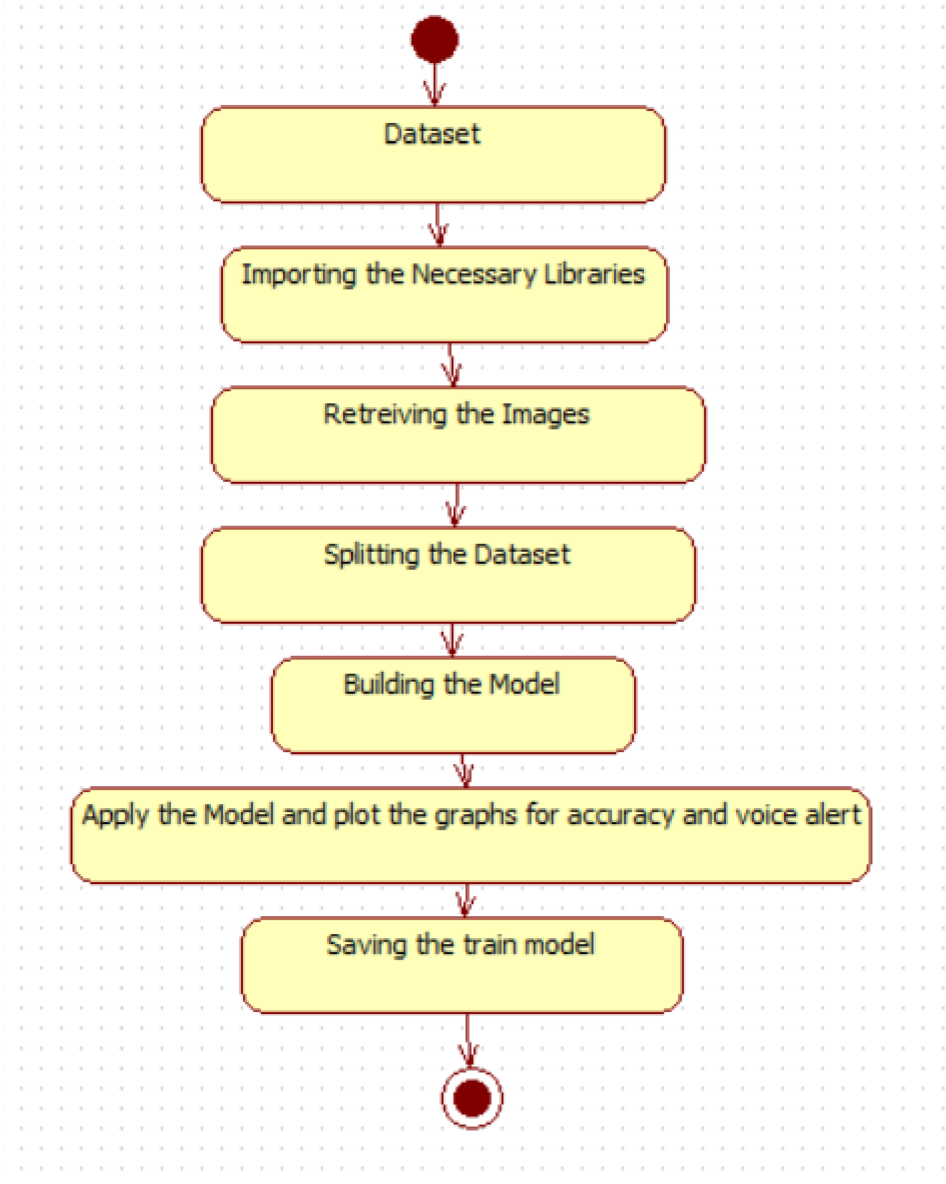
3.5.2.3 OBJECT DIAGRAM



EXPLANATION:

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

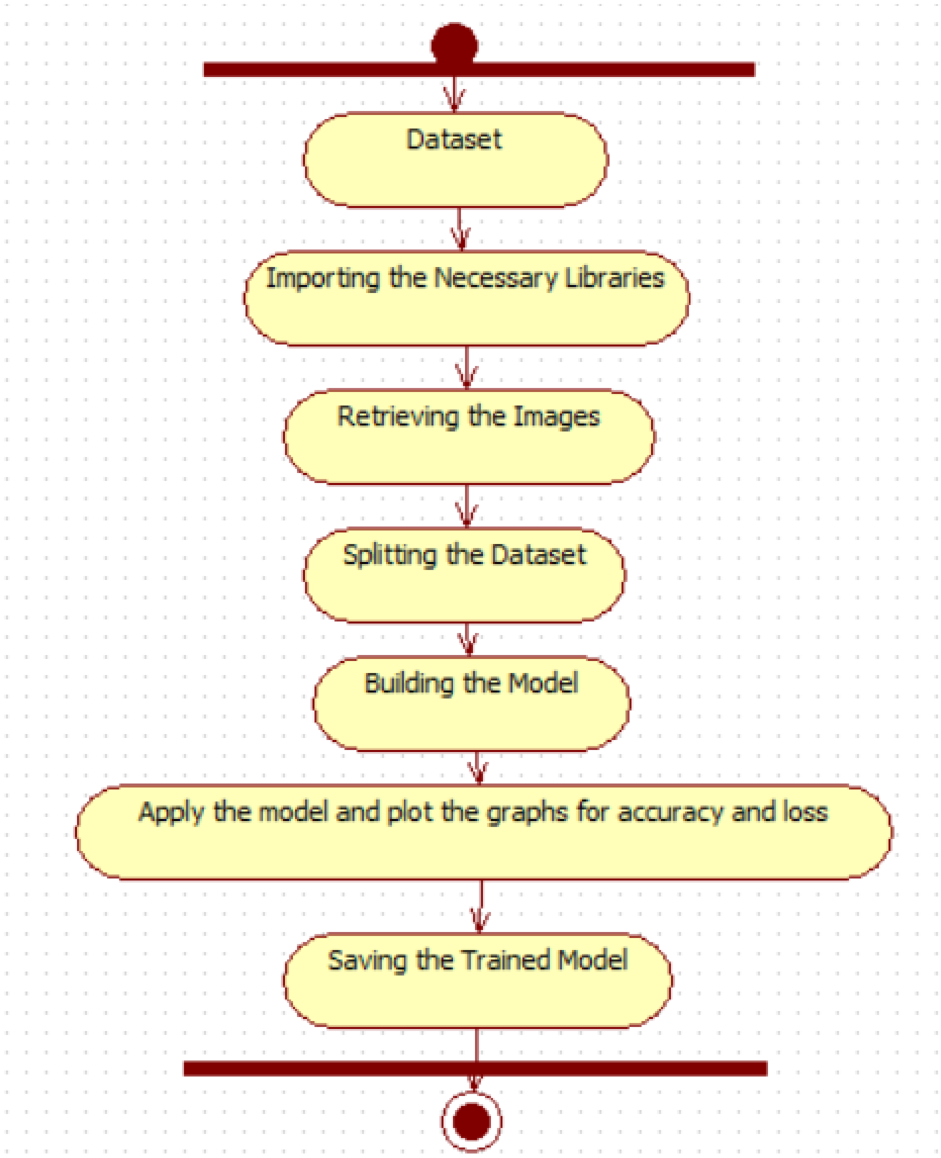
3.5.2.4 STATE DIAGRAM



EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

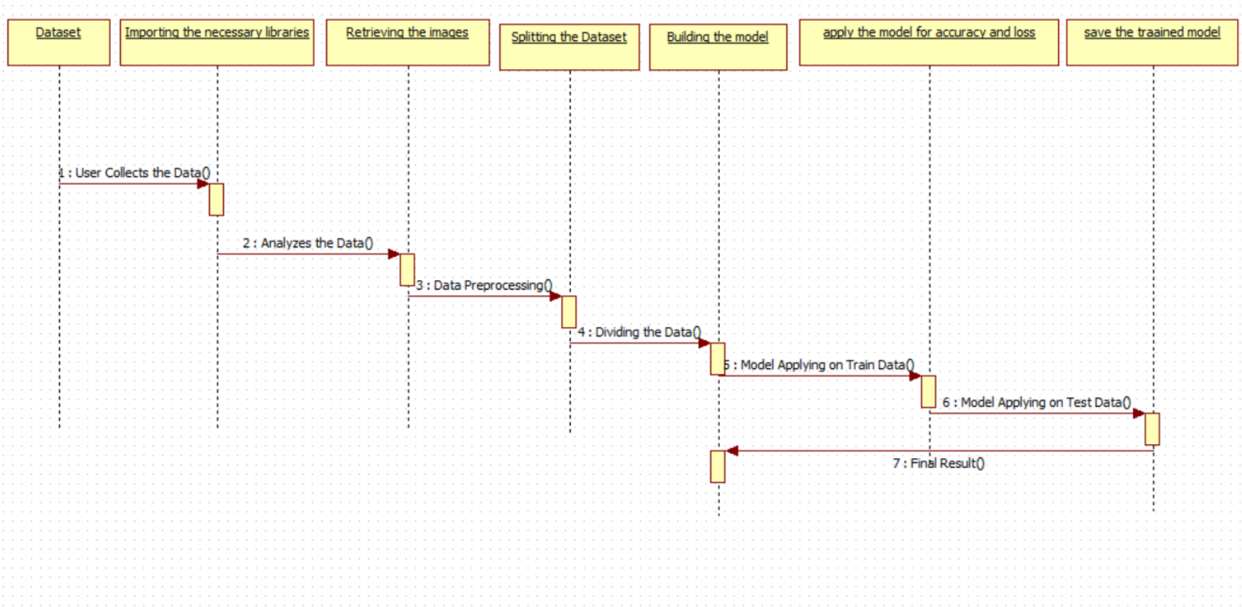
3.5.2.5 ACTIVITY DIAGRAM



EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

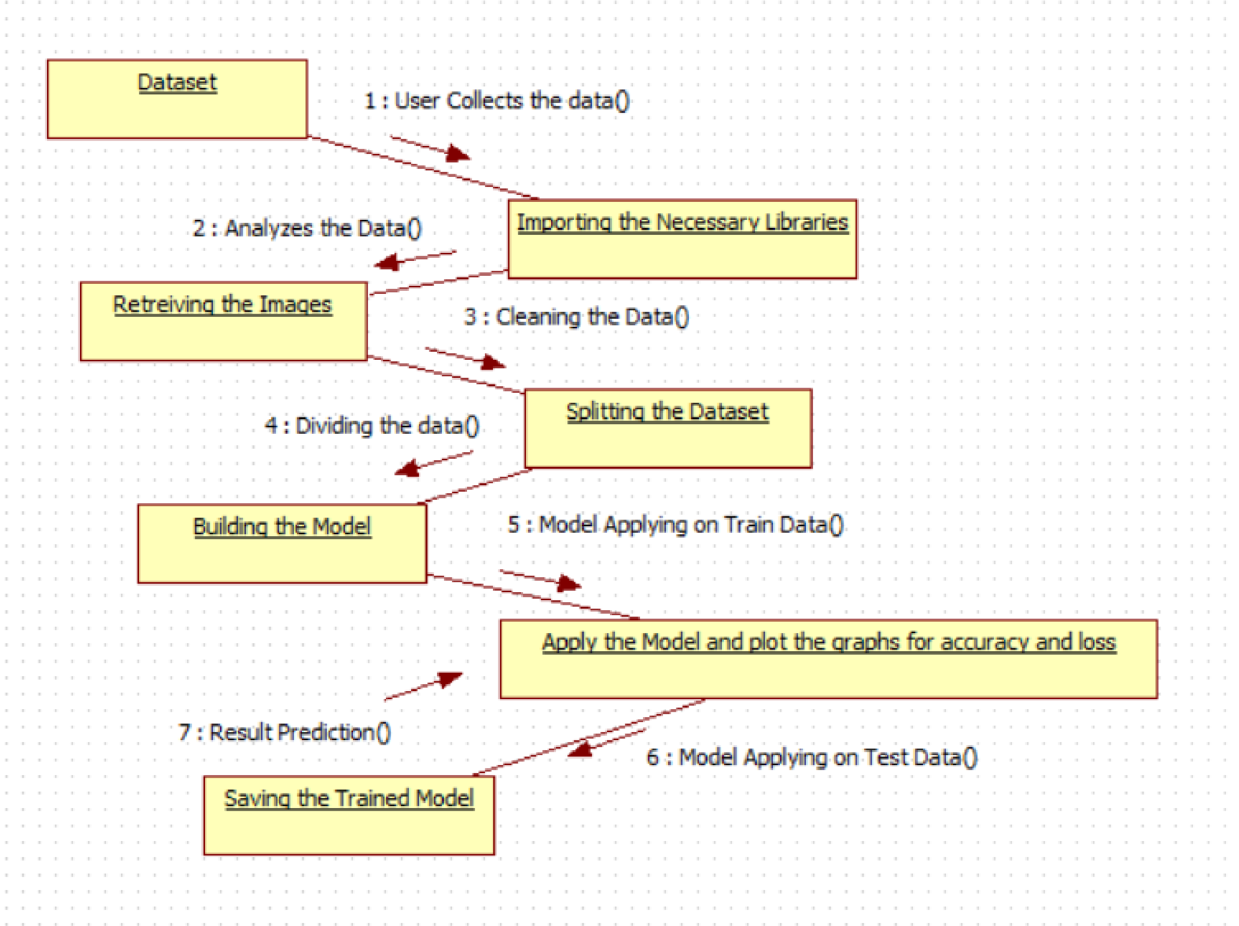
3.5.2.6 SEQUENCE DIAGRAM



EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

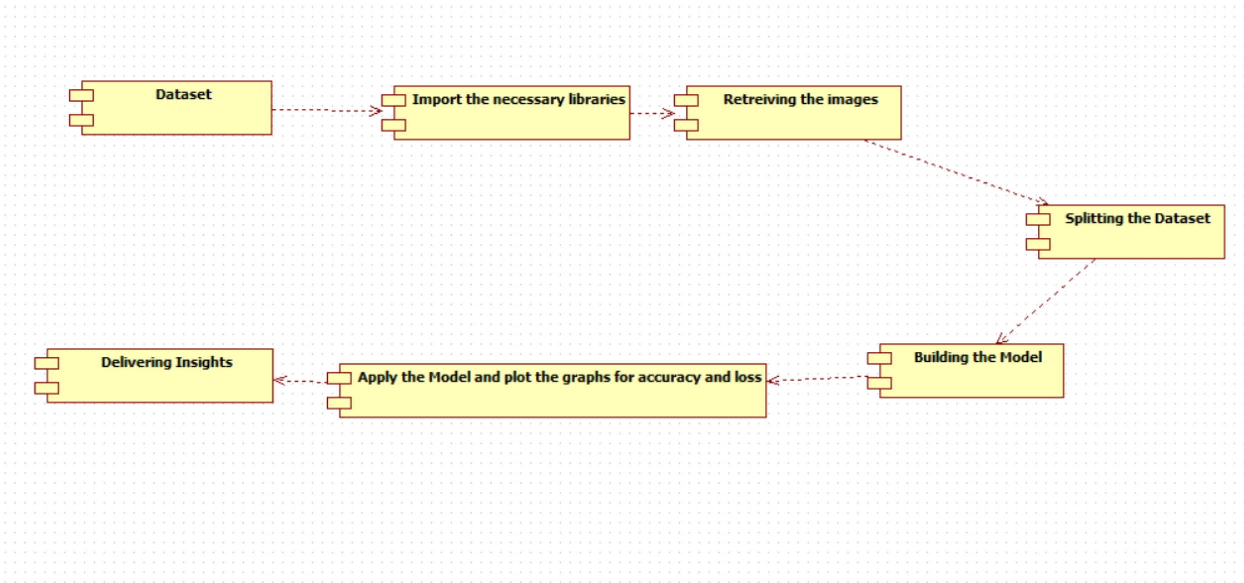
3.5.2.7 COLLABORATION DIAGRAM



EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

3.5.2.8 COMPONENT DIAGRAM

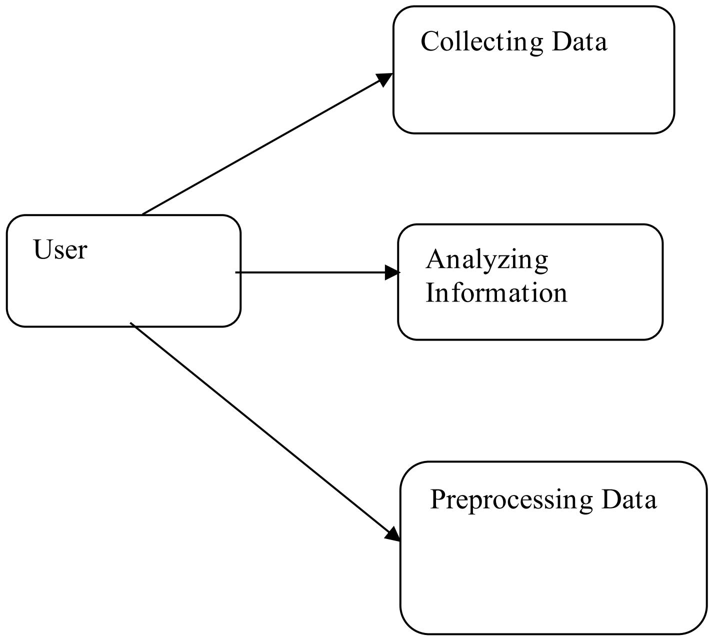


EXPLANATION:

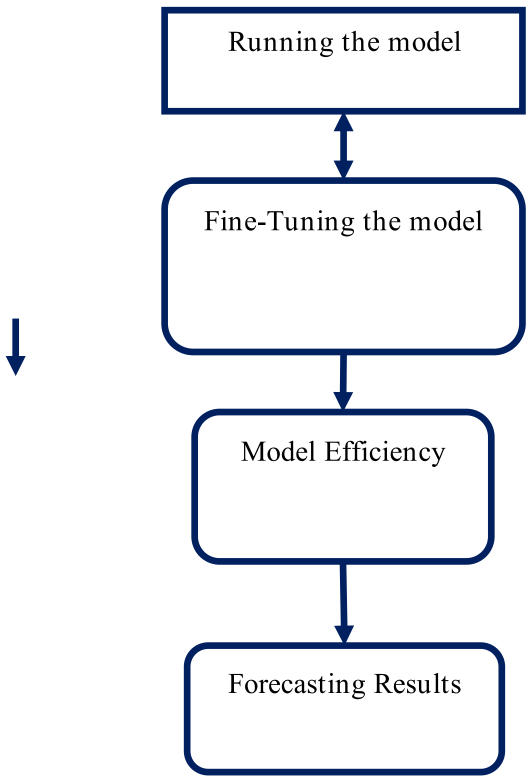
In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

3.5.2.9 DATA FLOW DIAGRAM

Level 0



Level 1

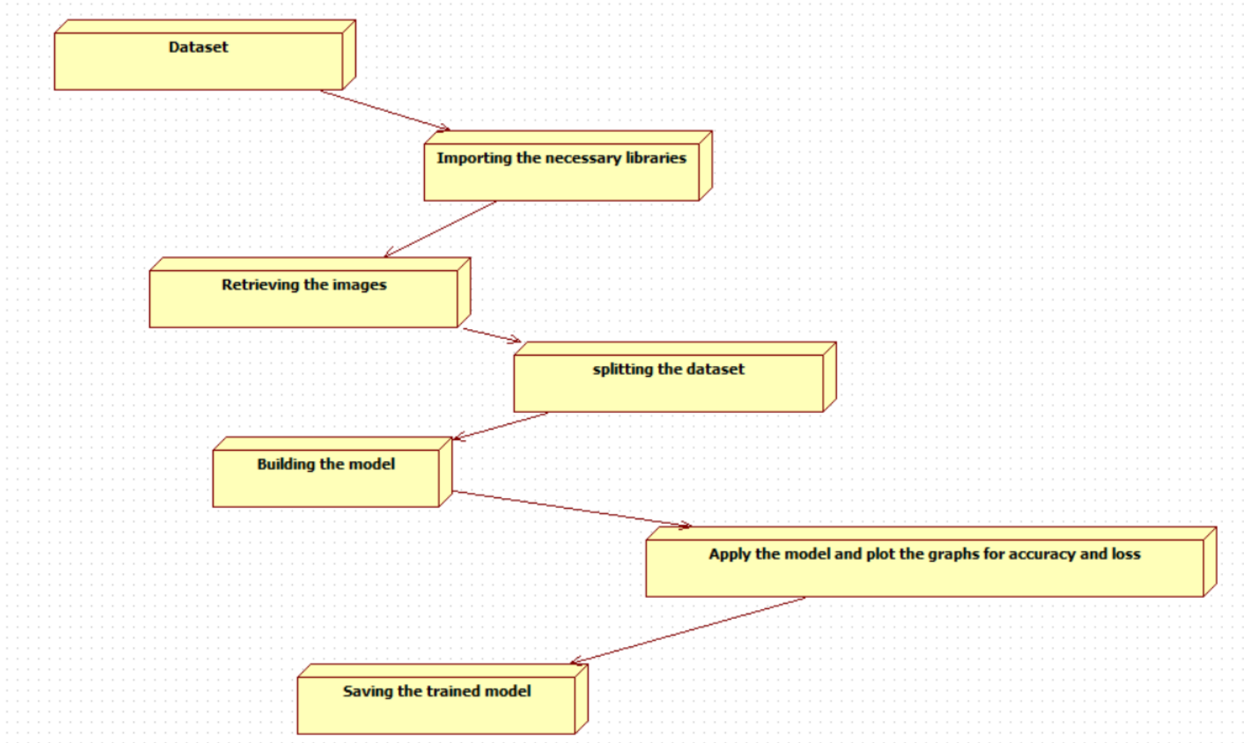


EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

3.5.2.10 DEPLOYMENT DIAGRAM



EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

3.5.2.11 SYSTEM ARCHITECTURE DIAGRAM

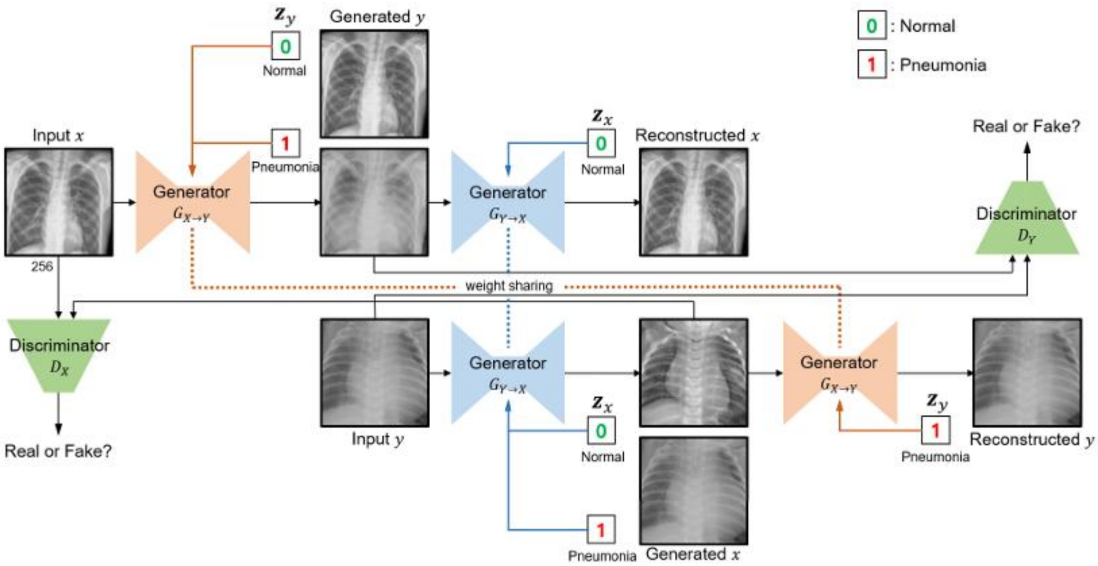


Fig 4.11: System Architecture

3.6 DEVELOPMENT TOOLS

3.6.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

3.6.2 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.6.3 IMPORTANCE OF PYTHON

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

3.6.4 FEATURES OF PYTHON

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

3.6.5 LIBRARIES USED IN PYTHON

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

3.7 IMPLEMENTATION

3.7.1 GENERAL

The Implementation is nothing but source code of project.

3.7.2 CODE

app.py

```
from flask import Flask, render_template, request, redirect, url_for, session, flash

import os

from PIL import Image, ImageOps

import numpy as np

import tensorflow as tfp

from tensorflow.keras.models import load_model

app = Flask(__name__)

app.secret_key = 'abcd123'

model = load_model('MobileNetV2_x-ray.h5')

class_names = ['NORMAL','PNEUMONIA']

users = {}

ALLOWED_EXTENSIONS  = {'png', 'jpg', 'jpeg', 'gif', 'bmp', 'DAT'}

MAX_CONTENT_LENGTH  = 30 * 1024 * 1024

app.config['MAX_CONTENT_LENGTH']  = MAX_CONTENT_LENGTH

def allowed_file(filename):
```

```
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

def import_and_predict(image_path, model):
    image = Image.open(image_path).convert('RGB')
    image = ImageOps.fit(image, (224, 224), Image.LANCZOS)
    img = np.asarray(image) / 255.0
    img = img[..., :3] if img.shape[-1] == 4 else img
    img_reshape = np.expand_dims(img, axis=0)
    predictions = model.predict(img_reshape)
    predicted_class_idx = np.argmax(predictions)
    return class_names[predicted_class_idx], predictions[0][predicted_class_idx]


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


@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
```

```
        if username in users:
            flash('Username already exists.', 'error')
            return redirect(url_for('signup'))
        users[username] = password
        flash('Signup successful!', 'success')
        return redirect(url_for('login'))
    return render_template('signup.html')


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if username in users and users[username] == password:
            session['username'] = username
            flash('Login successful!', 'success')
            return redirect(url_for('predict'))
        flash('Invalid username or password.', 'error')
        return redirect(url_for('login'))
    return render_template('login.html')
```

```
@app.route('/index')

def index():

    if 'username' in session:

        return render_template('index.html')

    flash('You need to log in first', 'error')

    return redirect(url_for('login'))


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

def predict():

    if request.method == 'POST':

        file = request.files['file']

        if file and allowed_file(file.filename):

            try:

                file_path = os.path.join('static/uploads', file.filename)

                file.save(file_path)

                predicted_class, accuracy = import_and_predict(file_path, model)

                return render_template('result.html', disease=predicted_class,
accuracy=round(accuracy * 100, 2), real_image_path=f'/static/uploads/{file.filename}')

            except Exception as e:

                flash(f'Error: {str(e)}', 'error')
```

```
                return redirect(url_for('index'))

        flash('Invalid file format or file is too large.', 'error')

        return redirect(url_for('index'))

    return render_template('index.html')



@app.route('/performance')

def performance():

    labels = ['NORMAL','PNEUMONIA']

    values = [1341, 3875]

    return render_template('performance.html', labels=labels, values=values)



@app.route('/logout')

def logout():

    session.pop('username', None)

    flash('You have been logged out.', 'info')

    return redirect(url_for('login'))

if __name__ == '__main__':

    app.run(port=5000, debug=True)
```

3.8 SOFTWARE TESTING

3.8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

3.8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

3.8.3 TYPES OF TESTS

3.8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

3.8.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

3.8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

3.8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

3.8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

3.8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 Build the test plan

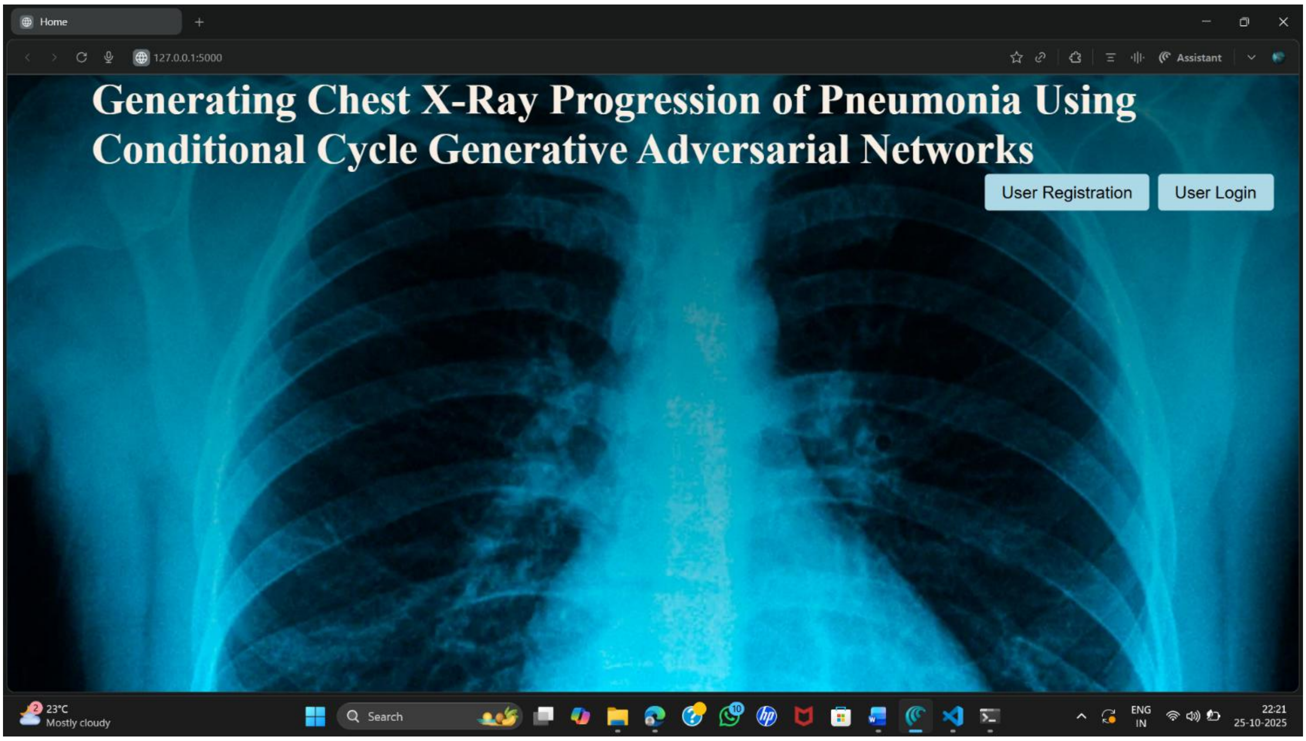
Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

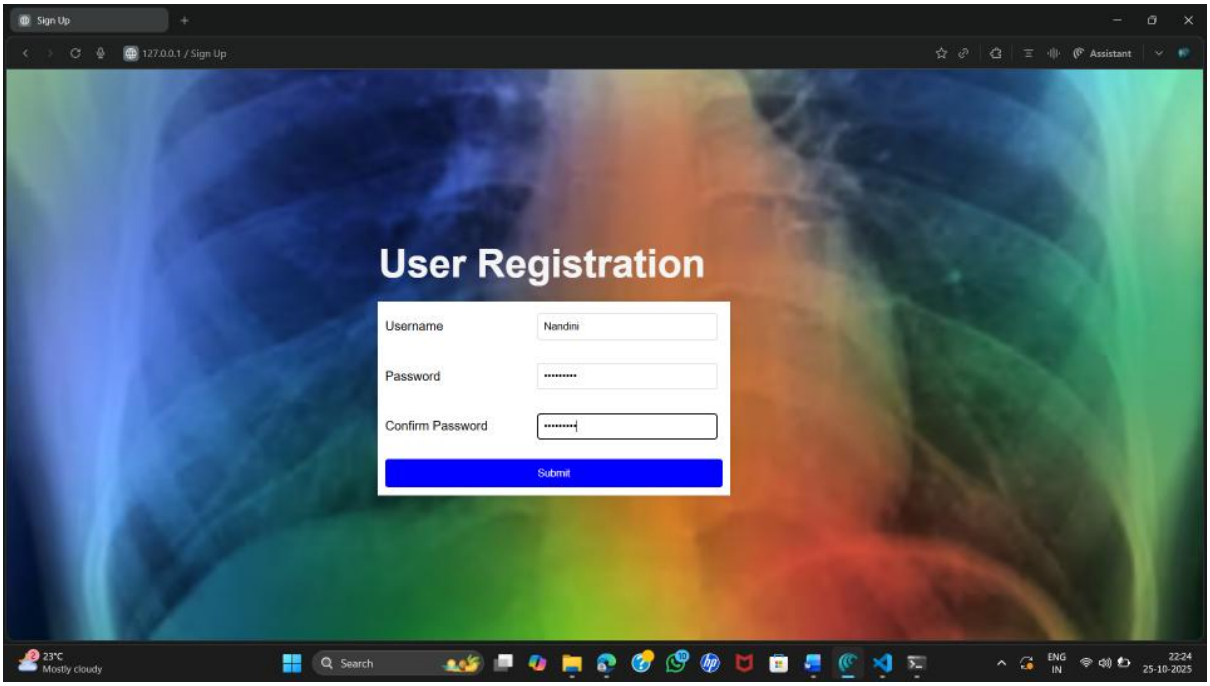
3.9 SNAPSHOTS

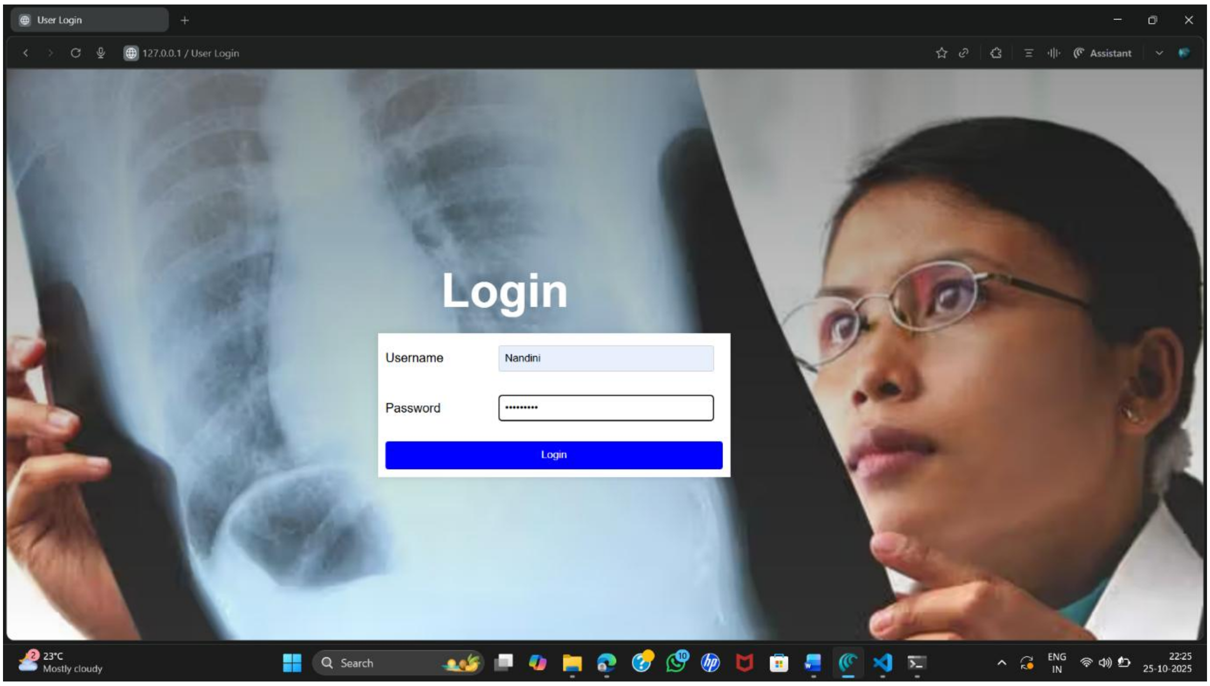
3.9.1 GENERAL

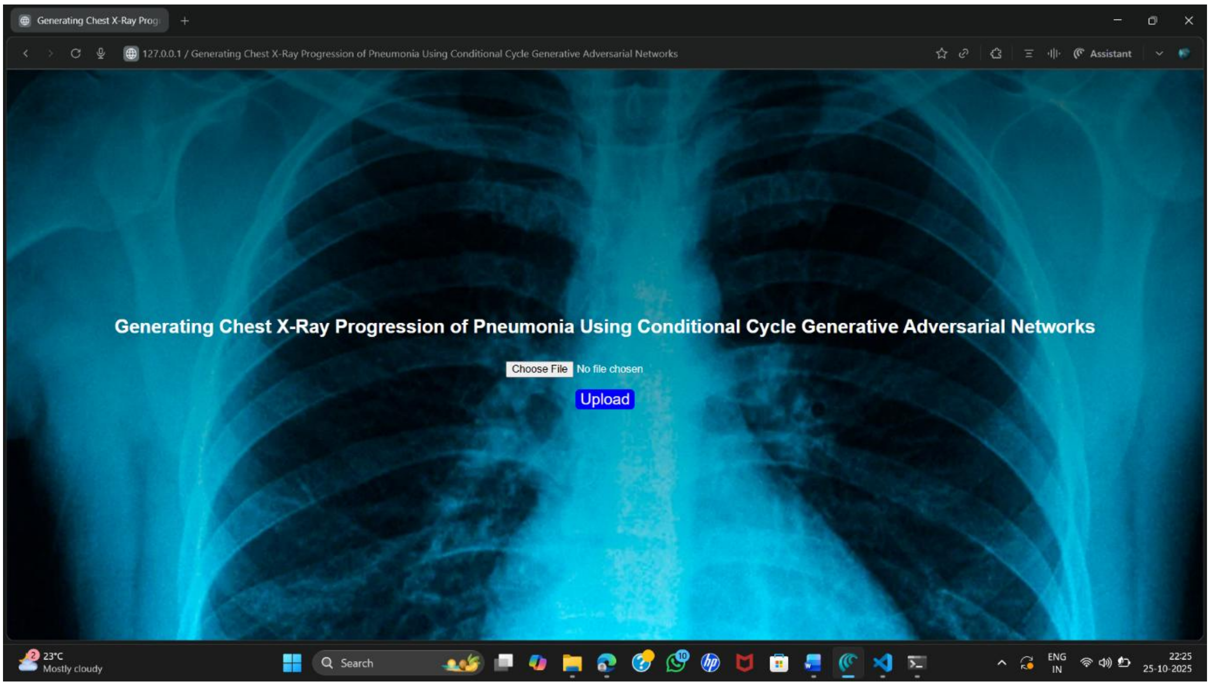
The system efficiently classifies facial expressions into seven categories using a CNN model for real-time emotion detection. Results are visually presented through performance graphs, highlighting key metrics like accuracy. This output demonstrates reliable recognition suitable for practical, real-world applications.

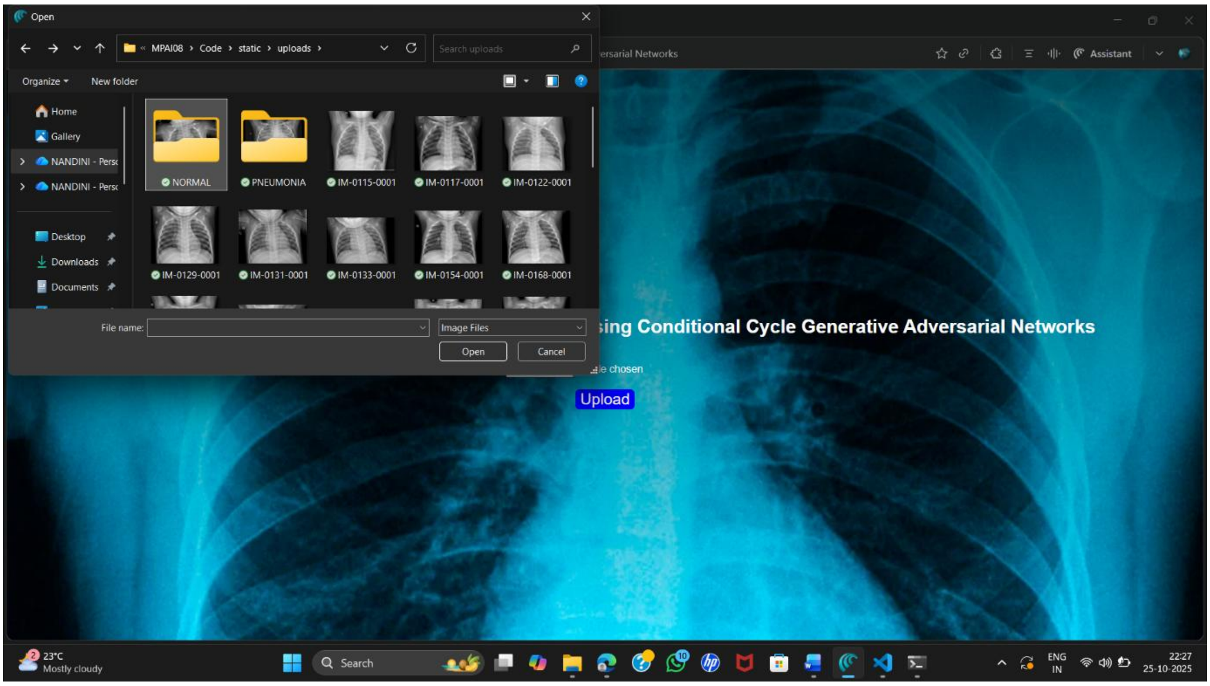
3.9.2 VARIOUS SNAPSHOTS

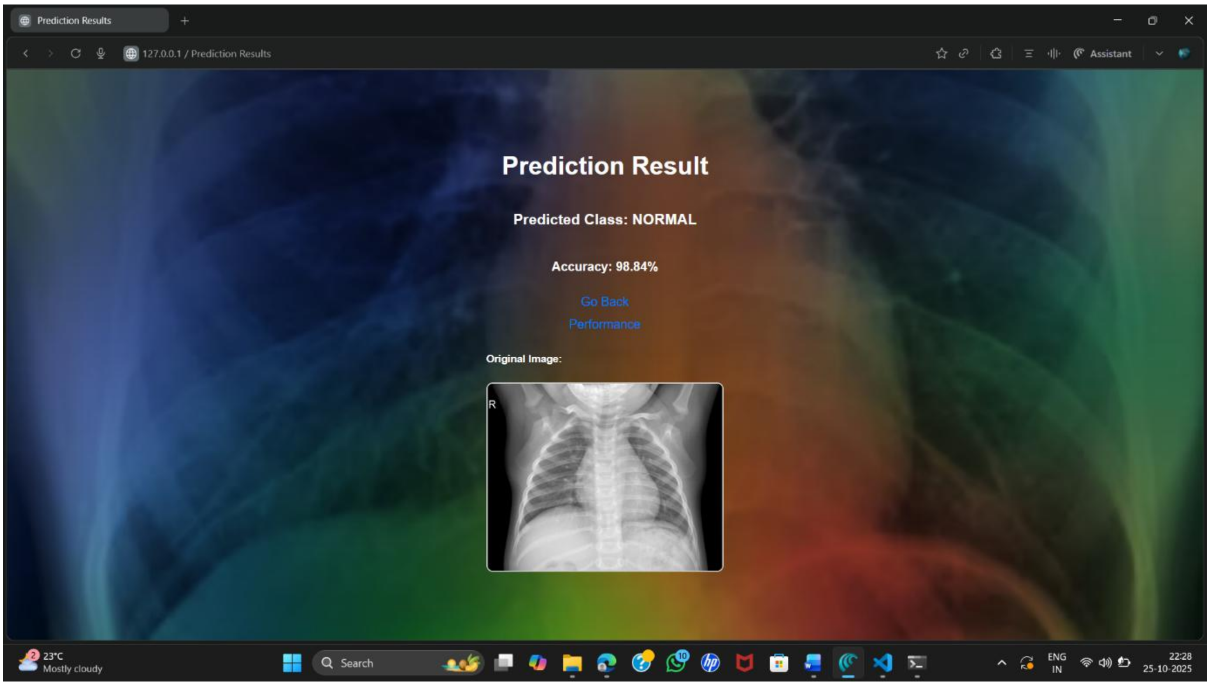


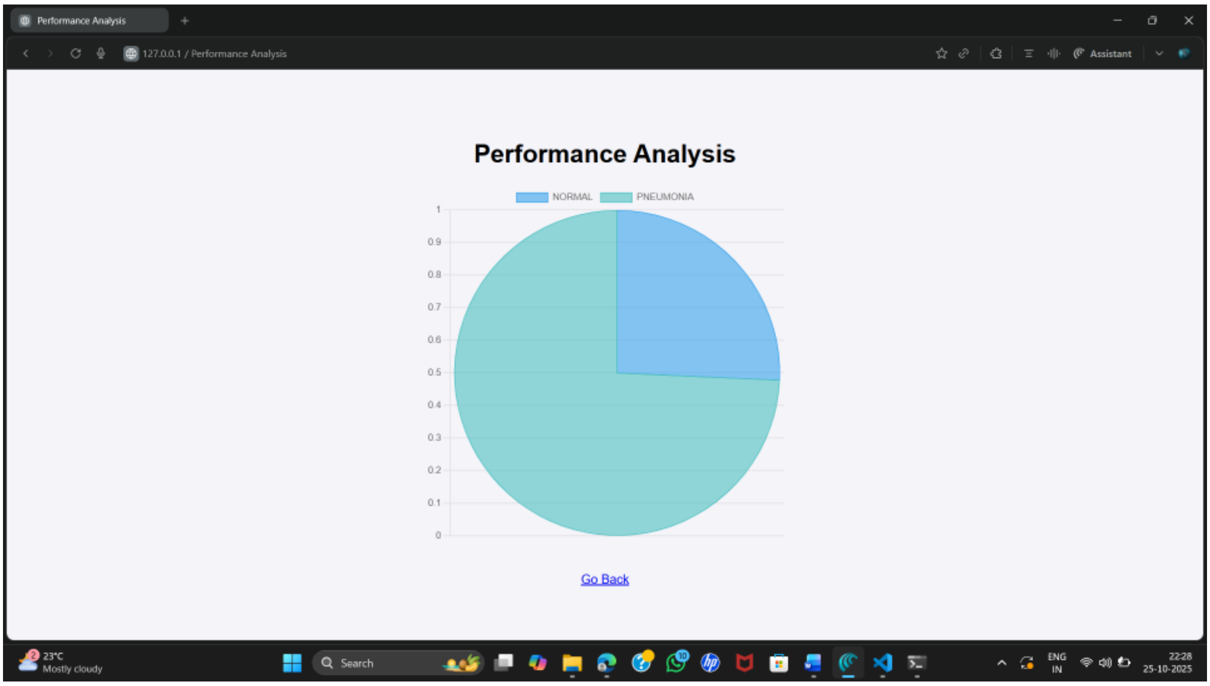

Sign Up
127.0.0.1 / Sign Up
User Registration
Username
Nandini
Password
Confirm Password
Submit
23°C
Mostly cloudy
Search
22:24
25-10-2025

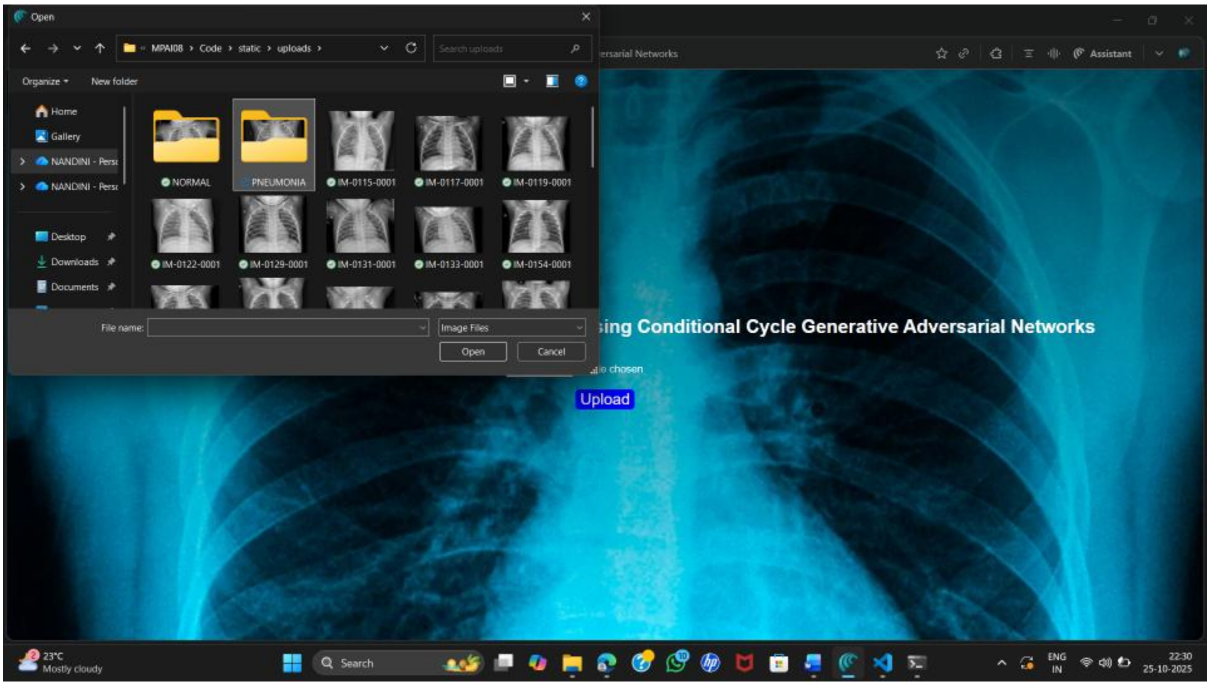

User Login
127.0.0.1 / User Login
Login
Username
Nandini
Password
Login
23°C
Mostly cloudy
Search
22:25
25-10-2025

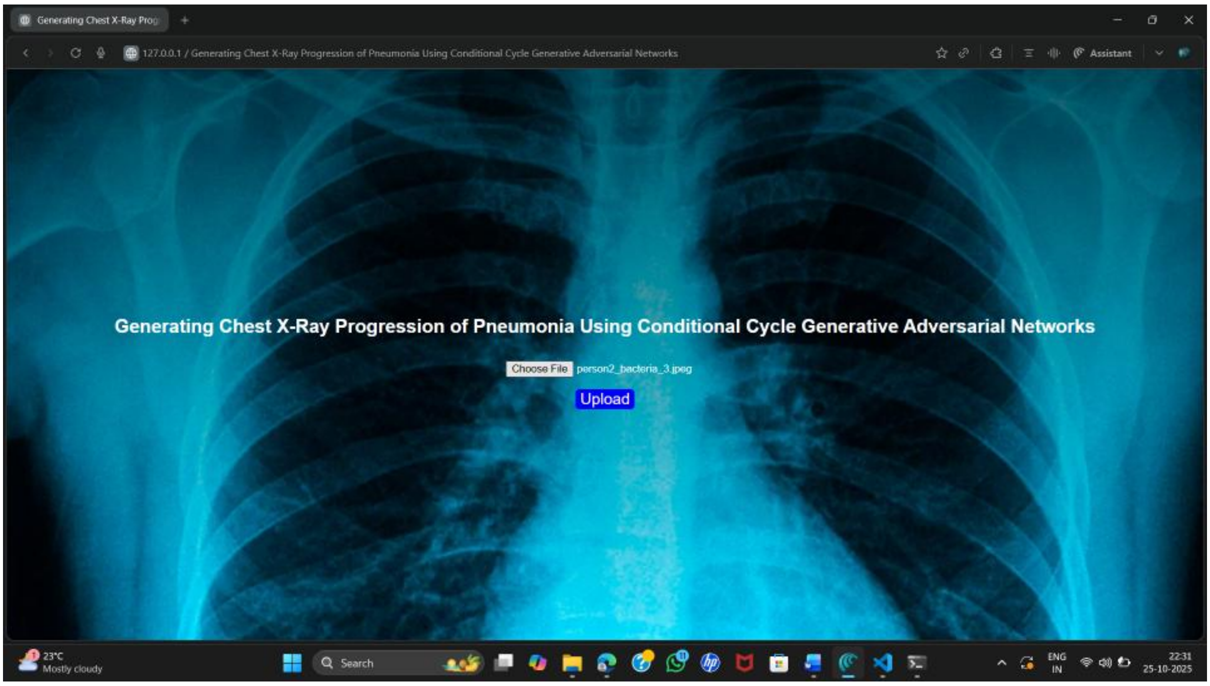

Generating Chest X-Ray Prog
127.0.0.1 / Generating Chest X-Ray Progression of Pneumonia Using Conditional Cycle Generative Adversarial Networks
Assistant
Generating Chest X-Ray Progression of Pneumonia Using Conditional Cycle Generative Adversarial Networks
Choose File
No file chosen
Upload
23°C
Mostly cloudy
Search
ENG
IN
22:25
25-10-2025

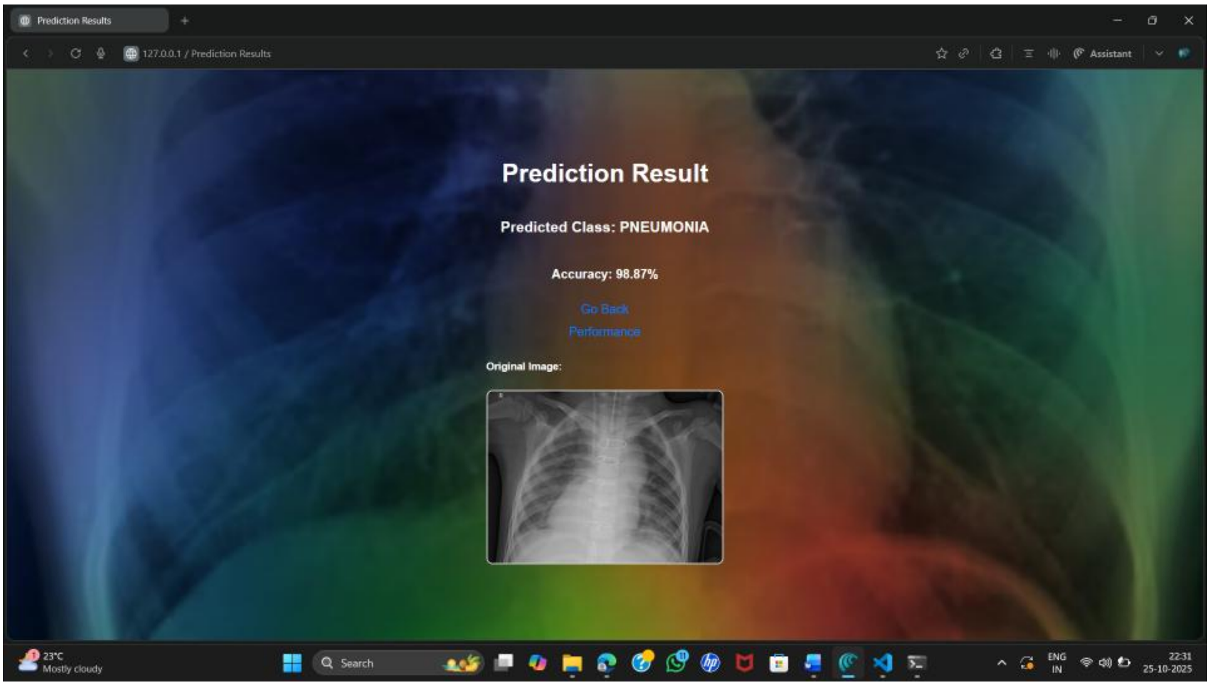

Open
MPAI08 › Code › static › uploads ›
Search uploads
Organize
New folder
Home
Gallery
Desktop
Downloads
Documents
NORMAL
PNEUMONIA
IM-0115-0001
IM-0117-0001
IM-0122-0001
IM-0129-0001
IM-0131-0001
IM-0133-0001
IM-0154-0001
IM-0168-0001
File name:
Image Files
Open
Cancel
sing Conditional Cycle Generative Adversarial Networks
Upload
23°C
Mostly cloudy
Search
ENG
IN
22:27
25-10-2025

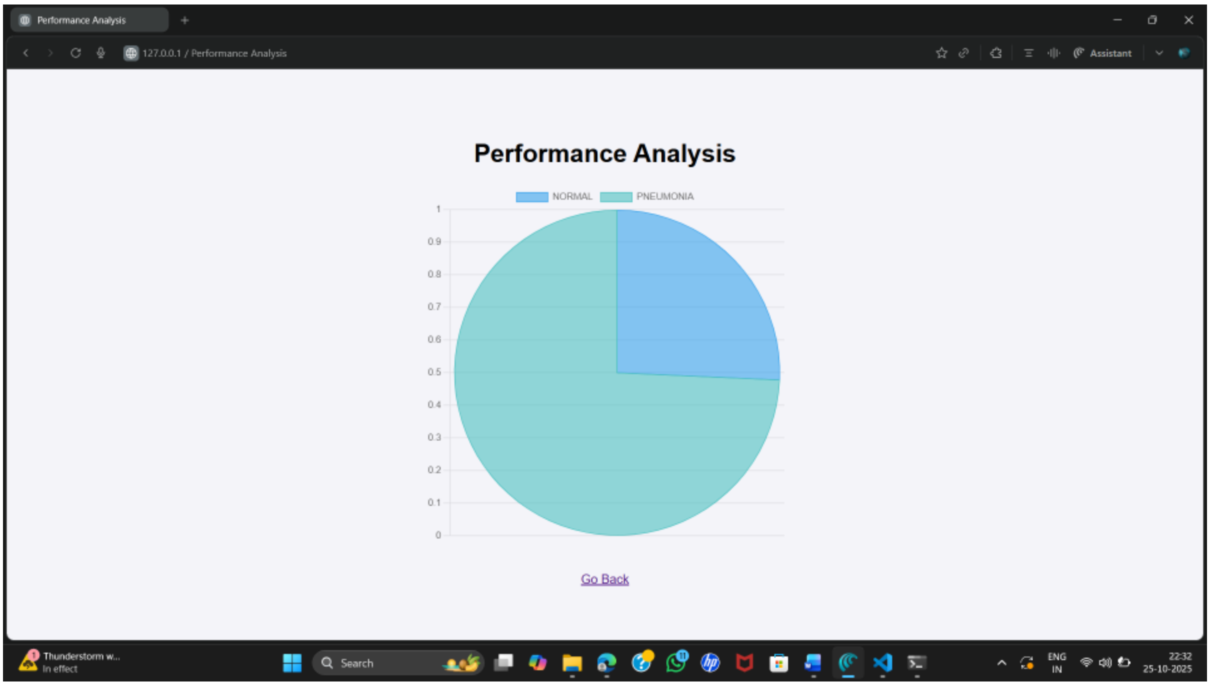

Prediction Results
127.0.0.1 / Prediction Results
Prediction Result
Predicted Class: NORMAL
Accuracy: 98.84%
Go Back
Performance
Original Image:


Performance Analysis
127.0.0.1 / Performance Analysis
Performance Analysis
NORMAL
PNEUMONIA
Go Back


Open
MPAI08 › Code › static › uploads ›
Organize
New folder
Home
Gallery
Desktop
Downloads
Documents
NORMAL
PNEUMONIA
IM-0115-0001
IM-0117-0001
IM-0119-0001
IM-0122-0001
IM-0129-0001
IM-0131-0001
IM-0133-0001
IM-0154-0001
File name:
Image Files
Open
Cancel
sing Conditional Cycle Generative Adversarial Networks
Upload


127.0.0.1 / Generating Chest X-Ray Progression of Pneumonia Using Conditional Cycle Generative Adversarial Networks
Generating Chest X-Ray Progression of Pneumonia Using Conditional Cycle Generative Adversarial Networks
Choose File person2_bacteria_3.jpeg
Upload


Prediction Results
127.0.0.1 / Prediction Results
Prediction Result
Predicted Class: PNEUMONIA
Accuracy: 98.87%
Go Back
Performance
Original Image:


Performance Analysis
127.0.0.1 / Performance Analysis
Performance Analysis
NORMAL
PNEUMONIA
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Go Back

CHAPTER 4

FUTURE ENHANCEMENT

4.1 FUTURE ENHANCEMENTS

The proposed model has demonstrated its effectiveness in diagnosing pneumonia and simulating disease progression using GANs. However, future enhancements can further improve its accuracy and usability. Integrating more diverse datasets from various demographic groups can help make the model more robust and generalizable. Additionally, incorporating multi-modal data such as patient history, symptoms, and laboratory reports alongside X-ray images can enhance diagnostic precision. Future work can also explore using 3D medical imaging techniques like CT scans to provide a more comprehensive view of lung infections. Advancements in explainable AI (XAI) methods can make the model's decision-making process more transparent, enabling better trust and adoption in clinical settings. Real-time deployment through cloud-based applications can allow doctors to access the model remotely, facilitating global healthcare support. Moreover, adding self-learning capabilities to continuously improve the model based on new medical data can enhance its adaptability. Collaborative research with medical experts can ensure the model aligns with real-world clinical requirements.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

This study presents a deep learning-based framework for pneumonia diagnosis and progression monitoring using GANs and ResNet152. By leveraging synthetic image generation, the model aids in understanding how pneumonia evolves over time, providing valuable insights for medical professionals. The results demonstrate the effectiveness of GANs in producing realistic X-ray images that closely resemble actual disease progression. The classifier successfully distinguishes between normal and pneumonia-affected images, contributing to early detection and improved patient care. The proposed system offers a promising AI-driven approach to assist radiologists in diagnosing pneumonia efficiently and accurately. Despite some limitations, the model's adaptability and potential for future enhancements make it a valuable addition to medical imaging technologies. With continued advancements and integration into healthcare systems, this framework can play a crucial role in improving pneumonia detection and treatment planning worldwide.

CHAPTER 6

REFERENCES

6.1 JOURNALS

[1] O. Ruuskanen, E. Lahti, L. C. Jennings, and D. R. Murdoch, ''Viral pneumonia,'' Lancet, vol. 377, no. 9773, pp. 1264–1275, 2011.

[2] K. K. Yadav and S. Awasthi, ''The current status of community-acquired pneumonia management and prevention in children under 5 years of age in India: A review,'' Therapeutic Adv. Infectious Disease, vol. 3, nos. 3–4, pp. 83–97, Jun. 2016.

[3] Y. Esayag, I. Nikitin, J. Bar-Ziv, R. Cytter, I. Hadas-Halpern, T. Zalut, and A. M. Yinnon, ''Diagnostic value of chest radiographs in bedridden patients suspected of having pneumonia,'' Amer. J. Med., vol. 123, no. 1, pp. 88.e1–88.e5, Jan. 2010.

[4] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, ''CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,'' 2017, arXiv:1711.05225.

[5] H. Salehinejad, S. Valaee, T. Dowdell, E. Colak, and J. Barfett, ''Generalization of deep neural networks for chest pathology classification in X-rays using generative adversarial networks,'' in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 990–994.

[6] C. Jozef Saul, D. Y. Urey, and C. D. Taktakoglu, ''Early diagnosis of pneumonia with deep learning,'' 2019, arXiv:1904.00937.