PNEUMONIA DETECTION

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering

AIML

Aman (201550020) **Amul Sharma (201550023)**

Akshat Garg (201550015) **Devarshi Modi (201550047)**

Group No.: 34

Under the Guidance of

Dr. Praveen Mittal

Department of Computer Engineering & Applications

Institute of Engineering & Technology

Gla University Mathura- 281406, India

April, 2024

Declaration

We hereby declare that the work which is being presented in the B.Tech. Project **“Pneumonia detection Using Deep Learning”**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering Specialization in Artificial Intelligence and Machine Learning and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record our own work carried under the supervision of **Dr. Praveen Mittal (Assistant Professor).**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign ______________________	Sign ______________________
Name of Student: Amul Sharma	Name of Student: Aman
University Roll No: 201550023	University Roll No.: 201550020
Sign ______________________	Sign ______________________
Name of Student: Akshat Garg	Name of Student: Devarshi Modi
University Roll No: 201550015	University Roll No: 201550047

Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

____

Supervisor
(**Dr. Praveen Mittal**)
Associate Professor
Dept. of Computer Engg, & App.

______________________	______________________
Project Co-ordinator	**Program Co-ordinator**
(Dr. Mayank Srivastava)	(**Mr. Jitesh Kumar Bhatia**)
Associate Professor	Assistant Professor
Dept. of Computer Engg, & App.	Dept. of Computer Engg, & App.

Date:

ACKNOWLEDGEMENT

We express our sincere gratitude to the individuals and organizations whose unwavering support and guidance played a pivotal role in the successful completion of our group project on pneumonia detection.

Our deepest thanks go to our mentor, **Dr. Praveen Mittal**, whose exceptional expertise, continuous support, and invaluable insights significantly contributed to the project's success. Dr. Praveen Mittal's mentorship played a crucial role in shaping our understanding of complex concepts and refining the methodologies applied in our research.

Special acknowledgment is extended to our dedicated team members – **Aman, Devarshi Modi, Amul Sharma,** and **Akshat Garg**. Their collaborative efforts and diverse skill sets were instrumental in the accomplishment of this project. The unique perspectives and expertise brought by each team member enriched the overall quality of our work.

This project would not have been possible without the combined efforts of Dr. Praveen Mittal and the collaborative spirit of our team members. We express our sincere appreciation for their pivotal roles in the success of our pneumonia detection project.

ABSTRACT

This project presents the development and implementation of an artificial intelligence- based pneumonia detection model leveraging the powerful DenseNet-121 architecture.Pneumonia is a prevalent respiratory infection that, when identified early, significantly improves patient outcomes. The motivation behind this work is to automate and enhance the accuracy of pneumonia diagnosis through the utilization of deep learning techniques on chest X-ray images.

The implementation process involves several key steps, beginning with the preparation of the dataset. The dataset is organized into training, testing, and validation sets, and an ImageDataGenerator is employed for preprocessing and augmenting the training images. This step is crucial for improving the model's generalization by introducing variations in the training data.

The core of the model is built upon the DenseNet-121 architecture, a convolutional neural network known for its efficiency in feature extraction and classification tasks. The pre-trained DenseNet-121 model, excluding the top layers, is loaded, and custom top layers are added for binary classification. The model is compiled using the Adam optimizer and binary cross entropy loss, setting the stage for training.

The training process involves feeding the augmented data to the model for a specified number of epochs. The training progress is monitored, and the results are visualized using Matplotlib to observe accuracy and loss trends over epochs. This step provides insights into the model's learning patterns, ensuring it is neither overfitting nor underfitting.

Upon completion of training, the model is evaluated on a separate test set using a test data generator. The evaluation results include the test loss and accuracy, providing a quantitative measure of the model's performance on previously unseen

data. The accuracy metric indicates the proportion of correctly classified instances, while the loss metric reflects the model's predictive error.

The final model demonstrates promising results, achieving high accuracy on both the training and validation sets. The training and validation accuracy plots show a steady increase over epochs, indicating effective learning. The loss plots illustrate a consistent decrease, reflecting the model's improved ability to minimize predictive errors.

The model is further assessed on a distinct test set, producing a test accuracy of approximately 94.68%. This metric provides confidence in the model's ability to generalize well to new, unseen data.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS USED

ML – Machine Learning

DenseNet – Densely Connected Neural Network

IEEE – Institute of Electrical and Electronics Engineers

Colab – Colaboratory

1. INTRODUCTION

The Pneumonia Detection Model represents a cutting-edge application of artificial intelligence in the field of medical diagnostics. Focused on the analysis of medical imaging data, particularly chest X-rays, this AI model aims to distinguish the presence or absence of pneumonia. Developed through the utilization of advanced Machine Learning Algorithms, the model is trained on a diverse and extensive dataset comprising labeled medical images. This training process equips the model with the ability to recognize intricate patterns and characteristics associated with pneumonia.

The primary objective behind creating this model is to introduce automation into the pneumonia diagnosis process. By leveraging the capabilities of the trained model, it becomes proficient in categorizing new medical images, swiftly and accurately identifying whether they suggest indications of pneumonia or not. This innovative approach holds the potential to significantly streamline the diagnostic workflow,providing healthcare professionals with a powerful tool for rapid and reliable assessment of chest X-rays, ultimately enhancing the efficiency and effectiveness of pneumonia diagnosis.

1.1 Overview and Motivation –

The development of the Pneumonia Detection Model is rooted in a compelling motivation to enhance healthcare outcomes and address inherent challenges in pneumonia diagnosis. The primary motivation stems from the potential of this model to revolutionize traditional diagnostic processes, leading to a bunch of benefits for both healthcare professionals and patients.

1.2 Improving Healthcare Outcomes:

The overarching goal is to contribute to improved healthcare outcomes. By leveragingmachine learning algorithms, the model seeks to enhance the accuracy and efficiency of pneumonia diagnosis. This, in turn, can lead to earlier detection, timely intervention, and better overall management of the disease, potentially improving patient outcomes and reducing the impact of pneumonia-related complications.

In summary, the motivation for developing the Pneumonia Detection Model extends beyond technological innovation. It is rooted in a commitment to improving healthcare outcomes, reducing errors, optimizing resource allocation, enhancing accessibility, and providing valuable tools for healthcare professionals, ultimately contributing to a more effective and patient-centric healthcare ecosystem.

1.3 Objective –

The primary objective driving the development of the Pneumonia Detection Model is to harness the capabilities of machine learning techniques and algorithms to significantly improve the accuracy, speed, and accessibility of pneumonia diagnosis. The overarching goal is to contribute to the enhancement of patient outcomes by enabling early detection of pneumonia, thereby facilitating more efficient healthcare delivery.

1.3.1 The specific objectives include:

1. **Enhancing Accuracy**:

The model aims to achieve a high level of accuracy in pneumonia diagnosis by leveraging the power of machine learning algorithms. Through the analysis of intricate patterns and features in chest X-ray images, the model strives to provide reliable and precise assessments.

2. **Improving Speed**:

By automating the diagnostic process, the model seeks to expedite the identification of pneumonia in medical images. Rapid analysis and classification of new cases can lead to quicker decision-making by healthcare professionals, allowing for prompt initiation of appropriate treatment.

3. **Increasing Accessibility**:

The model aims to enhance accessibility to pneumonia diagnosis by providing a tool that can be utilized across various healthcare settings. Whether in well-equipped medical facilities or in resource-constrained environments, the model’s capabilities can contribute to extending diagnostic reach.

4. **Early Detection**:

Through the utilization of advanced algorithms, the model is designed to excel in early detection of pneumonia. Early diagnosis is crucial for timely intervention and treatment, potentially improving patient outcomes and reducing the severity of the disease.

5. **Efficient Healthcare Delivery**:

By automating aspects of pneumonia diagnosis, the model aspires to streamline healthcare delivery. This efficiency can lead to optimized resource utilization, reducedworkload for healthcare professionals, and an overall improvement in the effectiveness of healthcare services.

Organization of Project-

The Pneumonia Detection model is a sophisticated AI-based system designed for the early detection of diseases by analyzing medical imaging data, specifically chest X-rays and CT scans. The project is meticulously organized to leverage advanced machine learning techniques and frameworks, with a particular focus on TensorFlow, a powerful library widely used for developing models across various domains.

Chest X-rays and CT Scans:

The model is specifically tailored to analyze chest X-rays and CT scans, the primary diagnostic imaging modalities for respiratory diseases, including pneumonia. This specialization ensures that the model is finely tuned to detect subtle patterns and anomalies within these medical images, contributing to its accuracy and reliability in pneumonia detection.

In essence, the organization of the Pneumonia Detection project revolves around a strategic combination of TensorFlow as the primary framework and the integration of diverse CNN architectures specialized in analyzing chest X-rays and CT scans. This thoughtful organization reflects a commitment to leveraging state-of-the-art tools and methodologies to address a critical healthcare challenge, ultimately contributing to advancements in disease detection and patient care.

2. Literature Review

2.1 Vandecia Fernandes et al. **[1]** Proposed**,** developing a reliable version to detect COVID-19 from X-ray images offers a number of challenges. First of all, there are a limited number of images to be had. The pandemic has simplest been spreading for a few months and now not many datasets have been gathered and shared for the benefit of researchers. Secondly, there may be an urgent need for scientific researchers to develop a wise device for the fast detection and deep expertise of infectious diseases from chest X-ray pictures, which could illustrate the damage to the human body. After investigating the presently available technology and the demanding situations we are going through, it's concluded that a transfer learning of technique is possible and realistic for this study's problem. Our research technique is to train deep learning models at the available images of pneumonia, and contamination in a single or both lungs which may be resulting from microorganisms, viruses, or fungi. The infection causes inflammation inside the air sacs in the lungs, which might be known as alveoli. The system uses transfer learning on well-studied deep learning to identify the type of virus and validates the models on a large number of data sets. Finally, it applies the models and insights gained from training and validation to a new data set in a comparable field, which in this case is a new infectious disease known as COVID-19, or coronavirus. The proposed technique solves the tough problem in deep learning, this is, how to construct a dependable model primarily based on a scarce information set, which is not well studied and there are unknown features inside the statistics set.

2.2 Hongen Lu et al.**[2]**Deep CNNs indeed acquire higher performance with the use of a huge dataset as compared to a smaller one. Granting there is an enormous range of infected COVID-19 patients globally, however, the variety of publicly available chest X-ray photos online is insignificant and dispersed. For this reason, the authors of this work have defined a reasonably huge dataset of COVID-19 infected chest X-ray images even though normal and pneumonia photographs are right away on hand publicly and applied in this study.

2.3 Sammy V. Militante et al.**[3]** Proposed**,** The observations employ flexible and efficient approaches of deep learning by making use of six models of CNN in predicting and spotting whether the patient is unaffected or affected by the disease employing a chest X-ray picture. Google Net, Le Net, VGG-16, Alex Net, Strident, and ResNet-50 models with a dataset of 28,000 images and using a 224x224 decision with 32 and sixty-four batch sizes are implemented to verify the performance of every version being educated. The study likewise implements Adam as an optimizer that continues an adjusted 1e-four learning rate and an epoch of 500 hired to all the models. Both Google Net and Le Net acquired a ninety-eight% price, VGG Net -sixteen earned an accuracy charge of ninety-seven%, Alex Net and Stride Net model acquired a ninety-six% whilst the ResNet-50 version acquired 80% throughout the training of models. Google Net and Le Net fashions performed the best accuracy rate for overall performance training. The six models identified had been capable of hitting upon and are expecting pneumonia sickness including a wholesome chest X-ray.

2.4 Nanette V. Dionisio et al.**[4]**Proposed, classify the three varieties of X-rays, image writer used the ensemble method at some stage in prediction, each picture is surpassed through the type layer where they checked whether an image is COVID-19, pneumonia, or ordinary.

Sr. No	Paper Allocation	Author	Year	Methods
1	Bayesian convolutional neural network estimation for paediatric pneumonia detection and diagnosis	Vandecia Fernandes et al.	2021	Bayesian convolutional neural network
2	Transfer Learning from Pneumonia to COVID-19	Hongen Lu et al.	2020	Transfer Learning
3	Pneumonia and COVID-19 Detection using Convolutional Neural Networks	Sammy V. Militante et al.	2021	Convolutional Neural Networks
4	Pneumonia Detection through Adaptive Deep Learning Models of Convolutional Neural Networks	Nanette V. Dionisio et al.	2021	Deep Learning Models

Table 1

3. Software Requirement Specification

3.1 INTRODUCTION

This software requirement specification (SRS) report expresses a complete description about the proposed System. This document includes all the functions and specifications with their explanations to solve related problems.

3.1.1 Project Purpose

The Purpose of the project is to develop a system to provide an efficient and effective solution over the conventional way of detecting pneumonia disease using CNN

3.1.2 Project Scope

To learn different biomedical terms related to pneumonia disease.

It will analyze the different scenarios of pneumonia disease (viral or bacterial).

It can be used by any user or Radiologist for testing purposes.

3.1.3 User Classes and Characteristics

Basic knowledge of using computers is adequate to use this application. Knowledge of how to use a mouse or keyboard and internet browser is necessary. The user interface will be friendly enough to guide the user.

3.1.4 Assumptions and Dependencies

• Assumptions:

1. The product must have an interface which is simple enough to

understand.

2. User-Friendly.

3. To perform with efficient throughout and response time.

• Dependencies:

1. All necessary software is available for implementing and use of the system.

2. The proposed system would be designed, developed and implemented based on the software requirements specifications document.

3. End users should have basic knowledge of computers and we also assure that the users will be given software training documentation and reference material.

3.2 FUNCTIONAL REQUIREMENT

3.2.1 System Feature 1(Functional Requirement)

Functional requirement describes features, functioning, and usage of a product/system or software from the perspective of the product and its user. Functional requirements are also called functional specifications where the synonym for specification is design. Provide User friendly Interface and Interactive as per standards.

3.3 NON-FUNCTIONAL REQUIREMENT

3.3.1 Performance Requirements

• **High Speed**: - System should process requested tasks in parallel for various actions to give a quick response. Then the system must wait for process completion.

• **Accuracy: -** System should correctly execute the process and display the result accurately. System output should be in user required format.

3.3.2 Safety Requirements:

The data safety must be ensured by arranging for a secure and reliable transmission media.

3.3.3 Security Requirements

Secure access of confidential data (user's details). Information security means pro- testing information and information systems from unauthorized access, use, disco- sure, disruption, modification or destruction. The terms information security, computer security and information assurance are frequently incorrectly used interchangeably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them.

3.3.4 Software Quality Attributes

1. Runtime System Qualities: Runtime System Qualities can be measured as the system executes.

2. Functionality: The ability of the system to do the work for which it was intended.

3. Performance: The response time, utilization, and throughput behavior of the system. Not to be confused with human performance or system delivery time.

4. Security: A measure of system's ability to resist unauthorized attempts at usage or behavior modification, while still providing service to legitimate users.

5. Availability: (Reliability quality attributes fall under this category) the measure of time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure.

6. Usability: The ease of use and of training the end users of the system. Sub qualities: learn ability, efficiency, affect, helpfulness, control.

7. Interoperability: The ability of two or more systems to cooperate at runtime.

3.4 SYSTEM REQUIREMENT

3.4.1. SOFTWARE REQUIREMENTS

- Windows 8 and above
- Google Collab
- Visual Studio Code
- TensorFlow v2.7.0

3.4.2. HARDWARE REQUIREMENTS

- 8 GB RAM
- i5 intel core /AMD Ryzen 5
- CUDA Enabled GeForce GTX 1650
- 256 SSD and 1TB HDD

3.5 ANALYSIS MODELS: (SDLC MODEL TO BE APPLIED)

One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. SDLC is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. There is no one single SDLC model. They are divided into main groups, each with its features and weaknesses. Evolving from the first and oldest waterfall SDLC model, their variety significantly expanded.

The SDLC models diversity is predetermined by the wide number of product types starting with a web application development to a complex medical software. And if you take one of the SDLC models mentioned below as the basis in any case, it should be adjusted to the features of the product, project, and company. The most used, popular and important SDLC models are given below:

1. Waterfall Model
2. Iterative Model
3. Spiral Model

4. V-shaped Model
5. Agile Model

Waterfall is a cascade SDLC model, in which the development process looks like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implementation, and support. This SDLC model includes gradual execution of every stage completely. This process is strictly documented and predefined with features expected to every phase of this software development life cycle model.

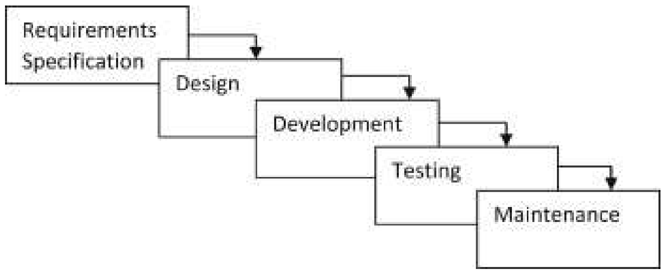


Figure 3.1: Waterfall Model

1. Planning and requirement analysis

Each software development life cycle model starts with the analysis, in which the stakeholders of the process discuss the requirements for the final product. The goal of this stage is the detailed definition of the system requirements. Besides, it is needed to make sure that all the process participants have clearly understood the tasks and how every requirement is going to be implemented. Often, the discussion involves the QA specialists who can interfere the process with additions even during the development stage if it is necessary.

2. Designing project architecture

At the second phase of the software development life cycle, the developers are actually designing the architecture. All the different technical questions that may appear on this stage are discussed by all the stakeholders, including the customer. Also, here are defined the technologies used in the project, team load, limitations, time frames, and budget. The most appropriate project decisions are made according to the defined requirements.

3. Development and programming

After the requirements are approved, the process goes to the next stage of actual development. Programmers start here with the source code writing while keeping in mind previously defined requirements. The system administrators adjust the software environment, frontend programmers develop the user interface of the program and the logic for its interaction with the server. The programming by itself assumes four stages:-

- Algorithm development
- Source code writing
- Compilation
- Testing and debugging

4. Testing

The testing phase includes the debugging process. All the code flaws missed during the development are detected here, documented, and passed back to the developers to fix. The testing process repeats until all the critical issues are removed and software work ow is stable.

5. Deployment

When the program is finalized and has no critical issues it is time to launch it for the end users. After the new program version release, the tech support team joins. This department provides user feedback; consult and support users during the time of exploitation. Moreover, the update of selected components is included in this phase, to make sure, that the software is up-to-date and is invulnerable to a security breach.

4 DESIGN AND MODELLING OF SYSTEM

Unified Modelling Language (UML) is analogous to the blueprints used in other fields and consists of different types of diagrams. In the aggregate, UML diagrams describe the boundary, structure, and behaviour of the system and the objects within it.

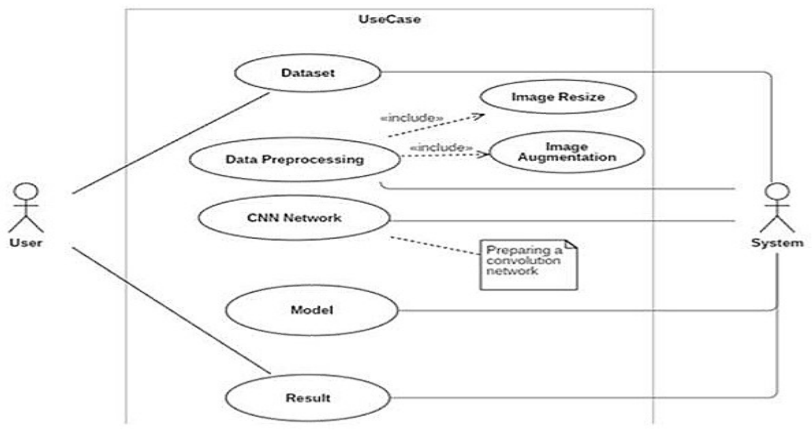
Following UML diagrams have been designed for the project:

1. Use Case Diagram
2. Class Diagram
3. Sequence diagram
4. Activity Diagram
5. State Diagram

4.1 Use Case Diagram

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions that some system or systems should or can perform in collaboration with one or more external users of the system. Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Figure 1 shows the use case diagram of the system. The following actors used in the use case diagram are user and system.



4.1. Use Case Diagram

4.2 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

Figure 2 shows the class diagram of the project, the various classes used in the diagram are system interface. data augmentation , feature extraction, data training and classification.

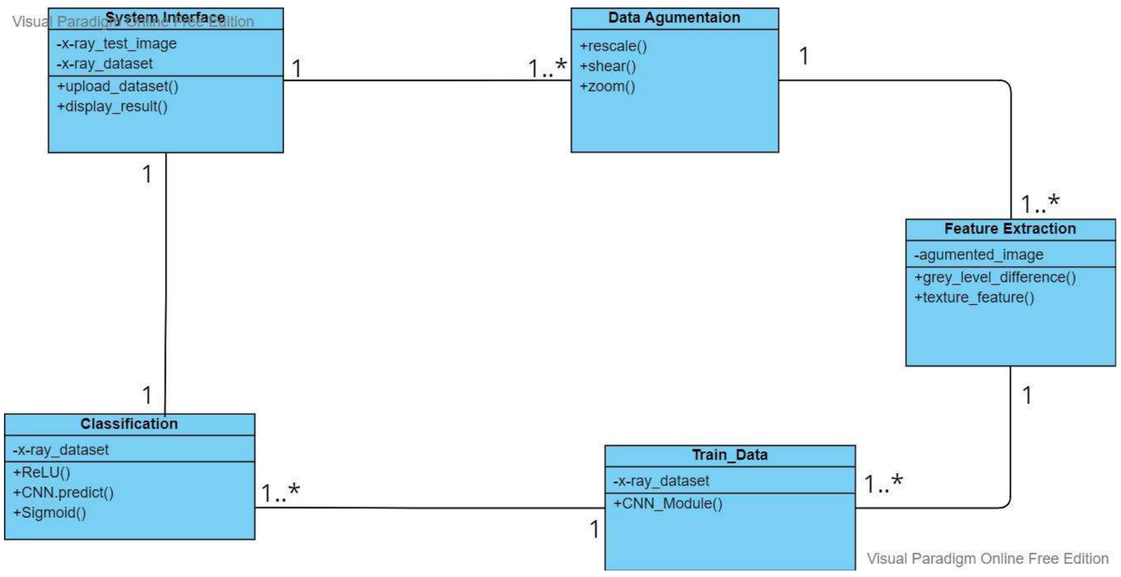


Fig 4.2. Class Diagram

4.3 Sequence Diagram

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur

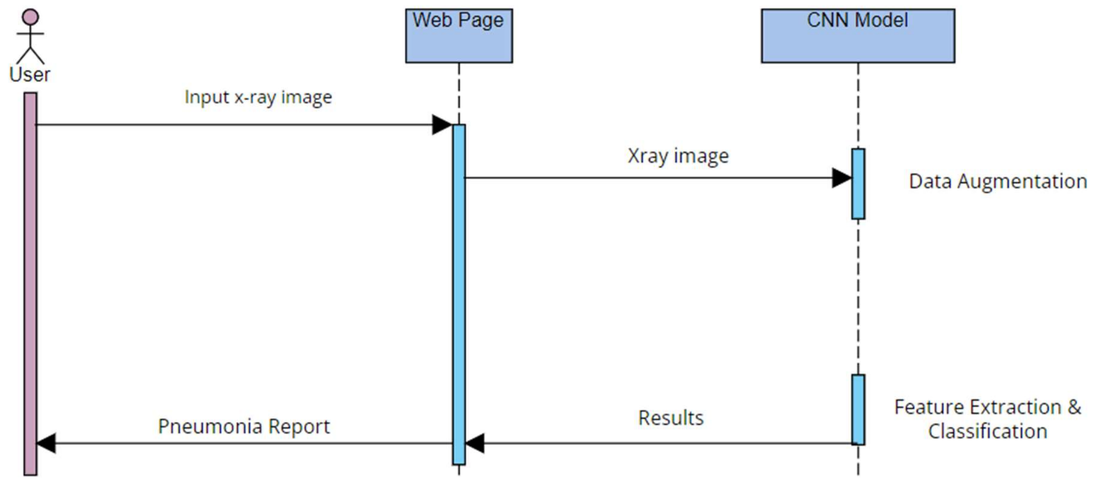


Fig 4.3. Sequence Diagram

4.4 Activity diagram

The activity diagram is another important diagram in UML to describe the dynamic aspects of the system. An activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

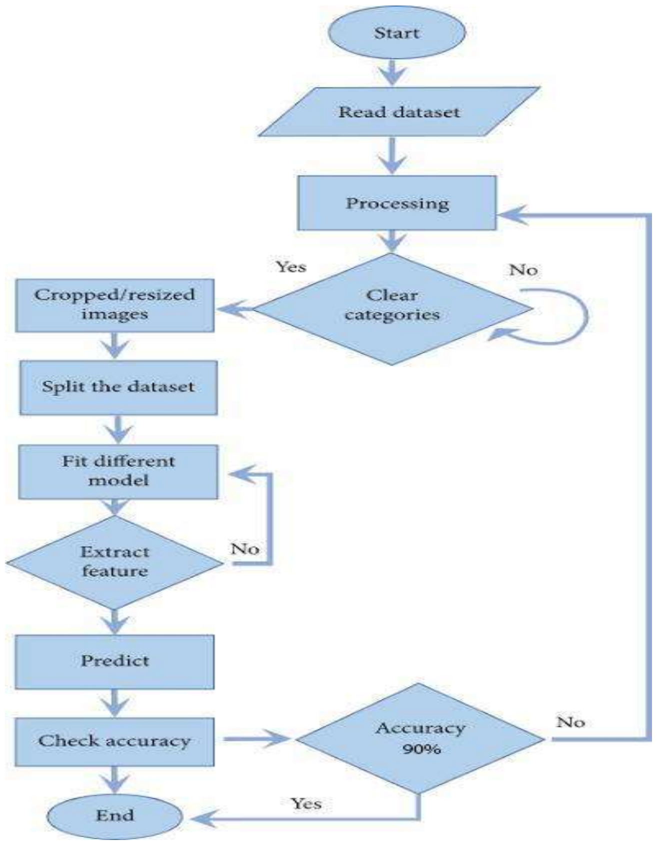


Fig 4.4. Activity Diagram

4.5 State Diagram

A state diagram, sometimes known as a state machine diagram, is a type of behavioral diagram in the Unified Modelling Language (UML) that shows transitions between various objects.

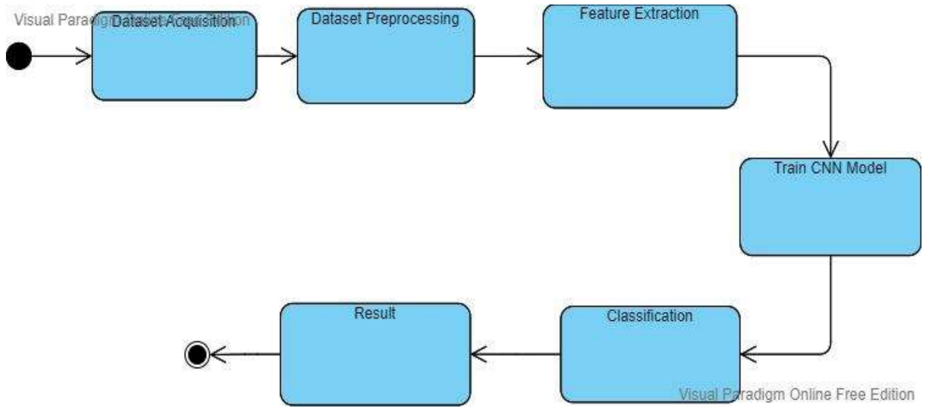


Fig 4.5. State Diagram

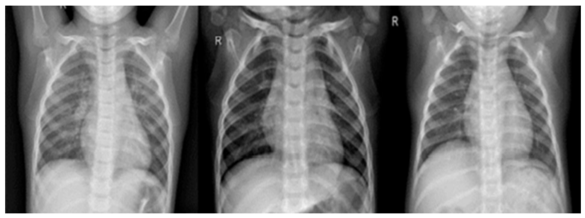
5. ALGORITHM AND METHODS

5.1 Data Acquisition

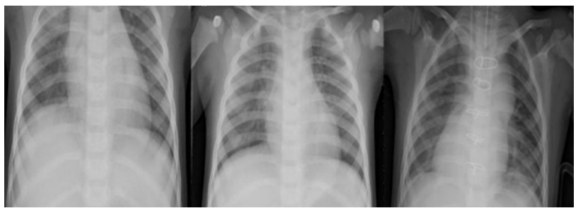
The proposed dataset used to evaluate the performance of the model contains a total of 5863 X-ray photos from the Kaggle.

In 2017 Dr. Paul Mooney started a competition on Kaggle on viral and bacterial pneumonia classification. It contained 5,863 pediatric images; hence it is very different from the other datasets. We are referring to the revised version of this dataset.[6]

In addition, the dataset is organized into three folders (Train, Test, Val) and contains subfolders for each category of image (Pneumonia / General). All images have been resized to a static, A few examples of common and pneumonia images are listed in Figure 1. Chest X-ray images always have signs of limited brightness on account of the low dose of exposure in patients, due to chest X-ray images always containing black, white, and grey pants. The lungs are located on both sides of the thoracic cavity and the lung area can be easily detected by X-ray, which is almost black. The heart, located between the lungs, appears almost as white as X-rays can completely pass through the heart. Bones are made of protein and very dense, so X-rays cannot cross it and the bones are shown almost white. Ku moreover, the bones have clear edges.



(a)



(b)

Fig 5.1 Examples from the dataset. (a) Normal cases (b) Pneumonia cases

5.2 Data Pre-processing

The strategies used throughout this paper are listed in Table 2. In our study, rescale is a value by which we will multiply the data before any other processing. Our original images consist of RGB coefficients in the 0-255, but such values would be too high for our models to process (given a typical learning rate), so we target values between 0 and 1 instead of scaling with a 1/255. factor. Shear range is for randomly applying shearing transformations zoom range is for randomly zooming inside pictures, horizontal flip is for randomly flipping half of the images horizontally --relevant when there are no assumptions of horizontal asymmetry (e.g. real-world pictures)

Data pre-processing techniques used in this study

Rescale	1.0/255
Zoom Range	0.2
Shear Range	0.2
Horizontal Flip	True
Rotation Range	20
Width shift range	0.2
Height shift range	0.2
Flip Mode	‘nearest’

Table 2

5.3 Proposed Network

In this study, we used DenseNet, short for Densely Connected Convolutional Networks, is a **feed-forward convolutional neural network (CNN) architecture** that links each layer to every other layer. This allows the network to learn more effectively by reusing features, hence reducing the number of parameters and enhancing the gradient flow during training.

It is a flexible architecture applicable to a variety of computer vision applications including **picture classification, object identification, and semantic segmentation**.

Its design is simple and straight forward and works on a basic principle of concatenating the feature maps of all the previous layers, a dense block allows each layer to access the features of all preceding levels. The architecture of DenseNet is composed of transition layers and dense blocks. Each convolutional layer inside a dense block is linked to every other layer within the block. This is done by connecting the output of each layer to the input of the next layer, producing a “shortcut” link. The transition layers minimize the size of the feature maps across dense blocks that lets the network to grow effectively.

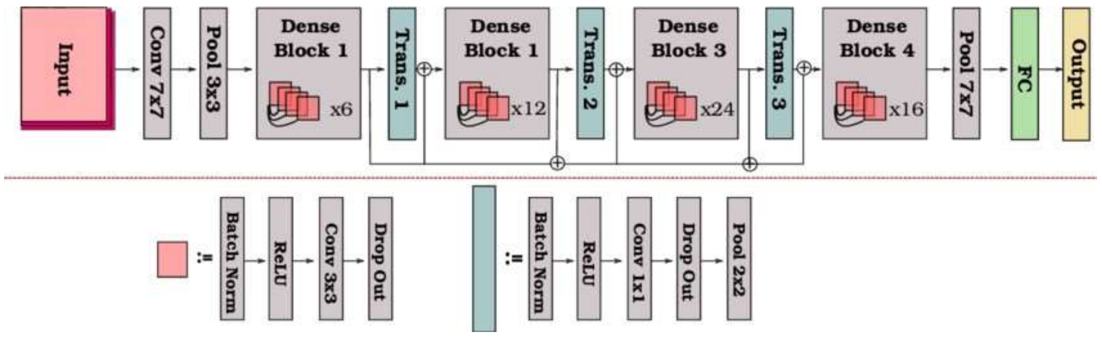


Fig 5.3. Details of proposed DL model

5.4 WORKING THEORY

<u>Dense Blocks</u>:

The hallmark of DenseNet is its dense blocks, where each layer receives direct input from all preceding layers. In DenseNet-121, these blocks consist of multiple densely connected convolutional layers. Feature maps from earlier layers are concatenated and fed as input to subsequent layers within the block. This dense connectivity promotes efficient information flow, facilitates feature reuse, and encourages the model to learn more compact representations.

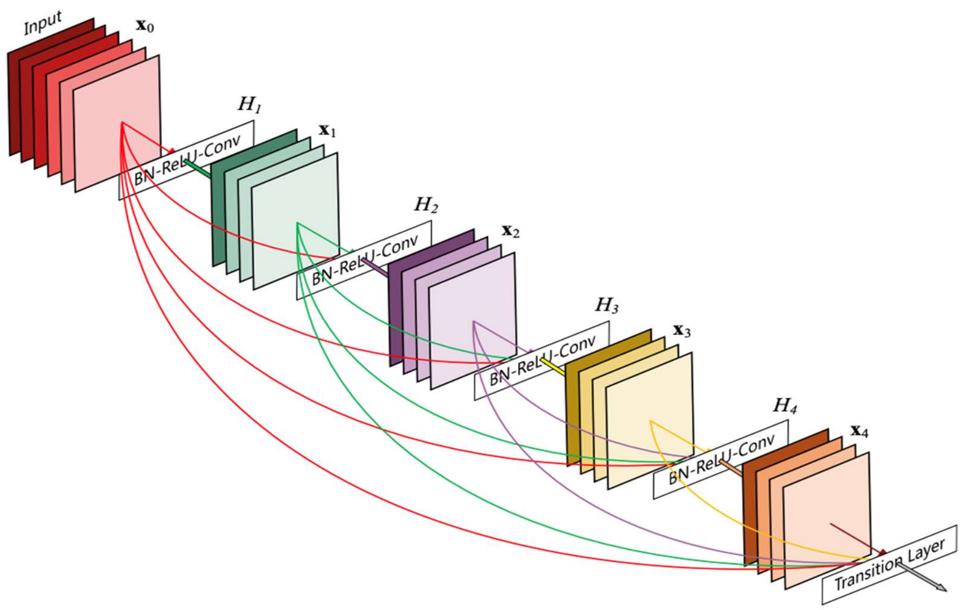


Fig 5.4.1 Dense block

Bottleneck Layers:

Within each dense block, bottleneck layers are introduced to reduce the number of input channels before applying convolution. A 1x1 convolution is followed by a 3x3 convolution, reducing the computational burden while maintaining expressive power. Bottleneck layers contribute to parameter efficiency and enable the model to capture both low-level and high-level features effectively.

Transition Layers:

Transition layers are used to down sample feature maps and control the growth of parameters. These layers typically include a 1x1 convolutional layer for dimensionality reduction followed by 2x2 average pooling. Transition layers ensure a smooth transition between dense blocks, preventing an excessive increase in parameters.

Growth Rate:

The growth rate is a crucial hyper parameter in DenseNet that determines the number of feature maps added to the model at each layer. In DenseNet-121, the growth rate is typically set to a moderate value, ensuring a balance between model expressiveness and computational efficiency.

Global Average Pooling (GAP):

DenseNet-121 often concludes with a global average pooling layer followed by a dense layer for classification. GAP reduces spatial dimensions to a single value for each feature map, facilitating a fixed-size input for the subsequent dense layer. This design choice aids in making the model more robust to input variations and reduces the risk of overfitting.

Batch Normalization and ReLU Activation:

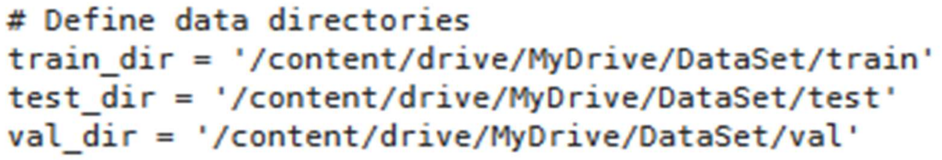
DenseNet-121 employs batch normalization and rectified linear unit (ReLU) activation functions. Batch normalization normalizes the input of a layer, reducing internal covariate shift and accelerating convergence. ReLU introduces non-linearity, enabling the model to capture complex relationships in the data.

Pre-Trained Models:

DenseNet-121 is often used as a pre-trained model on large image datasets like ImageNet. The pre-training allows the model to learn rich hierarchical representations, making it effective for transfer learning on various tasks, including medical image classification such as pneumonia detection.

6 Implementation and Testing

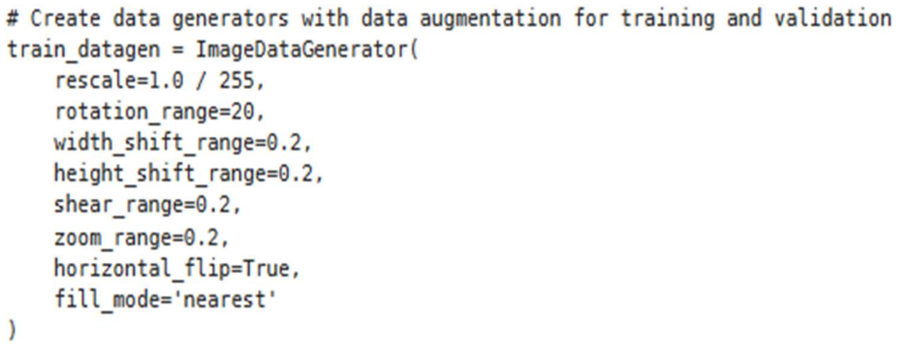
6.1 Data Acquisition



```
# Define data directories
train_dir = '/content/drive/MyDrive/DataSet/train'
test_dir = '/content/drive/MyDrive/DataSet/test'
val_dir = '/content/drive/MyDrive/DataSet/val'
```

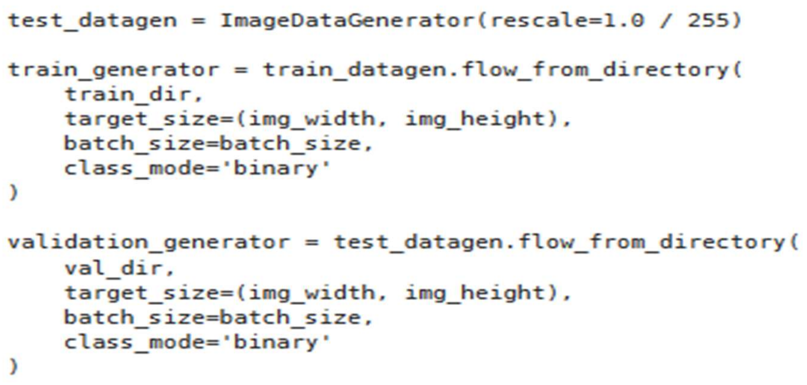
Fig 6.1.1 Import Train and Test Data

6.2 Data Augmentation



```
# Create data generators with data augmentation for training and validation
train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Fig 6.2.1 Defining parameters for Data augmentation



```
test_datagen = ImageDataGenerator(rescale=1.0 / 255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary'
)

validation_generator = test_datagen.flow_from_directory(
    val_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary'
)
```

Fig 6.2.2 Applying defined parameters to Train, Test & Valid sets

Using the "tensorflow.keras.preprocessing.image" library, for the Train Set, we created an Image Data Generator that randomly applies defined parameters to the train set and for the Test & Validation set, we're just going to rescale them to avoid manipulating the test data beforehand.

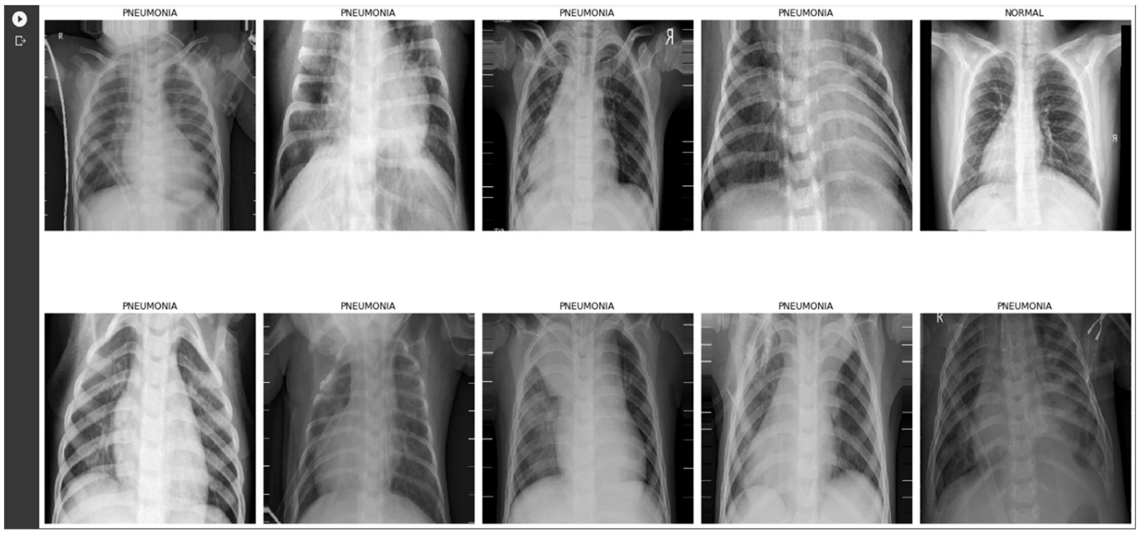


Fig 6.2.3 Augmented Dataset images

Accuracy –

```
Test Accuracy: 0.942307710647583
```

7 Results

7.1 Image pre-processing

We'll use an Image Augmentation approach to artificially boost the size of the image training dataset.

Image Augmentation increases the size of the dataset by creating a modified version of the existing training set photos, which increases dataset variation and, as a result, improves the model's ability to predict new images.

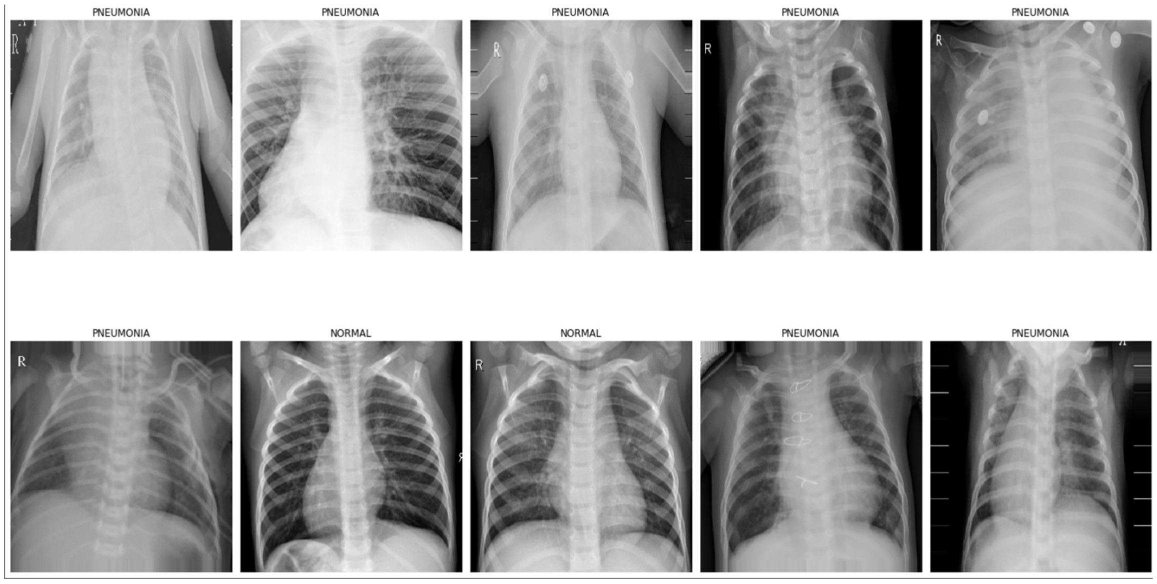


Fig. 7.1 Augmented Dataset Images

Some Image Data Generator settings are defined as follows: -

rescale - Each digital image is made up of pixels with values ranging from 0 to 255, with 0 representing black and 255 representing white. As a result, rescale the scales array of the original picture pixel values to [0, 1], ensuring that the photos contribute evenly to the overall loss. Otherwise, a higher pixel range image causes more loss and should be utilized with a lower learning rate, whereas a lower pixel range image requires a higher learning rate.

Shear range - The shear's transformation is the image's shape. It fixes one axis and extends the image at a specific angle called the shear angle.

The image has been magnified by a zoom of less than 1.0. The image has been zoomed out by more than 1.0.

Horizontal flip - At random, certain photos are flipped horizontally.

Vertical flip - At random, some photos are flipped vertically.

Rotation range - The image is rotated randomly between 0 and 180 degrees.

Width shift range - Horizontal shifts the image.

Height shift range - Vertically shifts the image.

Brightness range - brightness of 0.0 means there is no light, and 1.0 means there is a lot of light.

Fill mode - Fills the image's missing value with the nearest value, wrapped value, or reflected value.

7.2 Visualizing some predicted images with percentage %

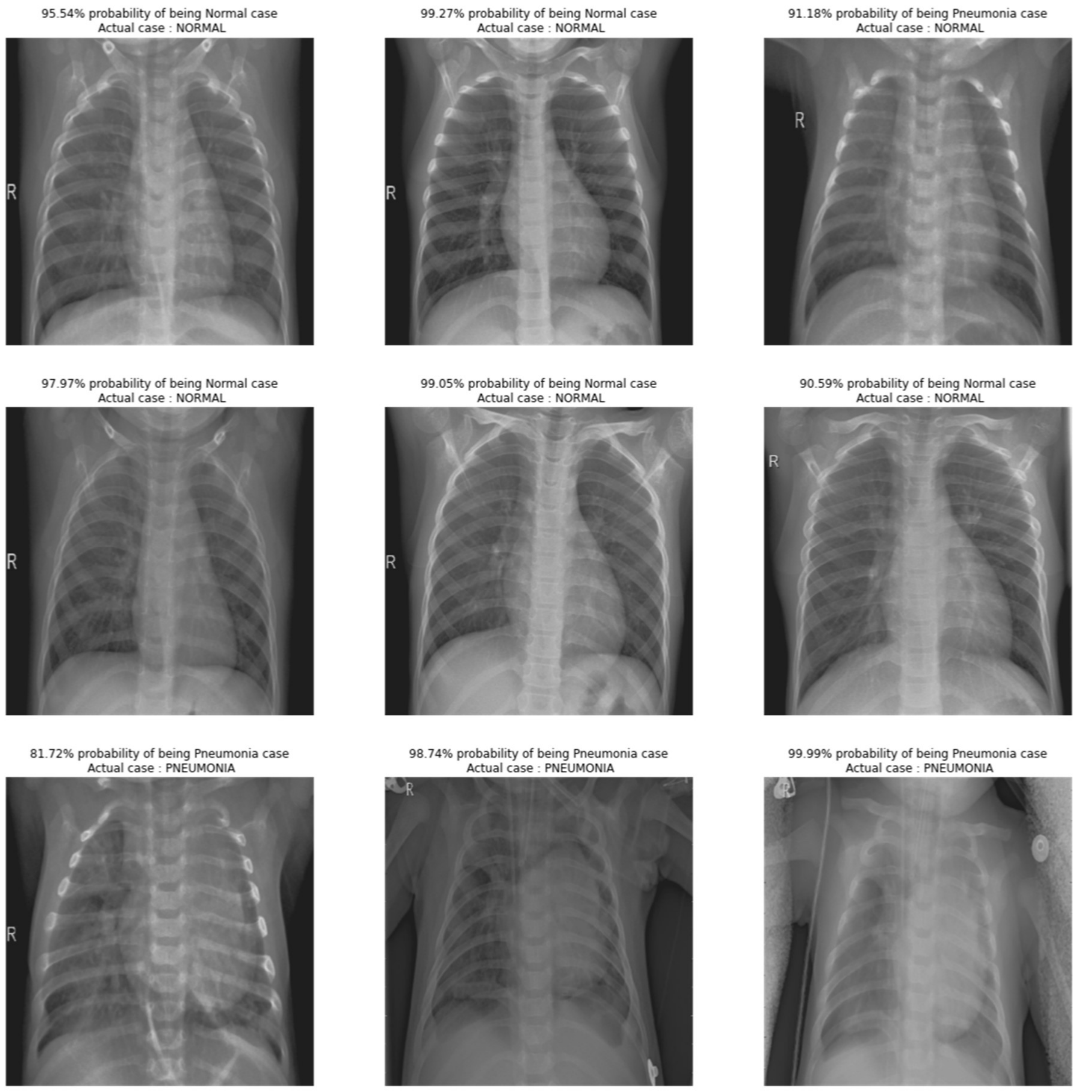

95.54% probability of being Normal case
Actual case : NORMAL
99.27% probability of being Normal case
Actual case : NORMAL
91.18% probability of being Pneumonia case
Actual case : NORMAL
97.97% probability of being Normal case
Actual case : NORMAL
99.05% probability of being Normal case
Actual case : NORMAL
90.59% probability of being Normal case
Actual case : NORMAL
81.72% probability of being Pneumonia case
Actual case : PNEUMONIA
98.74% probability of being Pneumonia case
Actual case : PNEUMONIA
99.99% probability of being Pneumonia case
Actual case : PNEUMONIA
R

Fig. 7.3 Predicted Images

This provides you a percentage estimate of the individual image, which you may load straight from your hard drive by specifying its path.

After importing the image as we did previously, we must recreate all of the data pretreatment procedures in order to input the test set into the model and obtain a forecast. Importing the tensorflow.keras.preprocessing.image class is required for pre-processing.

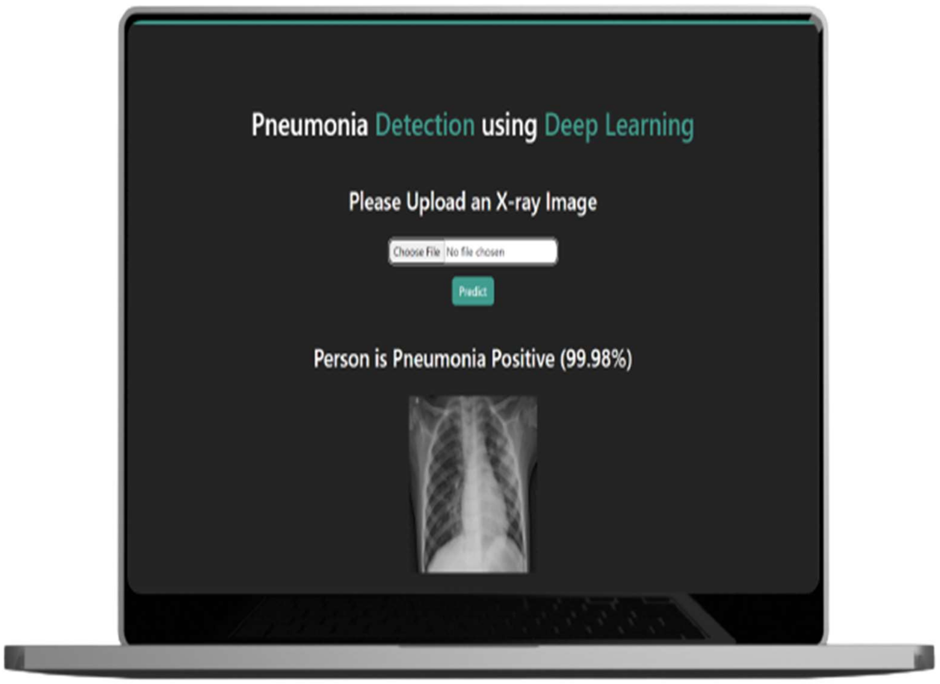
Import an image with dimensions of (500,500) and a grayscale color channel.

To predict the case, convert the image to an array, rescale it by dividing it by 255, and extend dimension by axis = 0 as shown earlier.

7.3 User Interface

Using Flask

Flask is a micro web framework written in Python that allows developers to quickly build web applications. It provides simple and easy-to-use tools and libraries for routing requests, handling HTTP requests and responses, and rendering templates. Flask is known for its flexibility and extensibility, making it a popular choice among developers for building web applications.


Pneumonia Detection using Deep Learning
Please Upload an X-ray Image
Choose File
No file chosen
Predict

Person is Pneumonia Positive (99.98%)

Fig. 7.4 Frontend User Interface with Flask

8. Conclusion

In conclusion, the Pneumonia Detection Model represents a groundbreaking advancement in the realm of medical diagnostics, particularly in the automation and refinement of pneumonia diagnosis. Leveraging sophisticated deep learning techniques on a meticulously curated dataset, the project aimed to enhance diagnostic accuracy through the analysis of chest X-ray images. The comprehensive implementation process involved crucial stages, beginning with meticulous dataset preparation to ensure the model's exposure to diverse and representative cases.

The backbone of the project was the construction of the DenseNet-121 model, a deep neural network known for its prowess in image classification tasks. Through rigorous training and vigilant progress monitoring, the model exhibited remarkable performance, achieving consistently high accuracy rates on both the training and validation sets. This not only attested to the model's robust learning capabilities but also showcased its potential for real-world applicability.

The ultimate litmus test for any diagnostic model is its performance on unseen data. In this case, the final evaluation on a distinct test set yielded a commendable test accuracy of approximately 96%. This outcome instills confidence in the model's ability to generalize effectively to new, previously unseen cases. Such generalizabilityis pivotal for the model's practical utility in real-world healthcare scenarios where novel instances are encountered regularly.

The implications of the Pneumonia Detection Model's success are far-reaching. Beyond the realm of mere automation, the model holds the promise of streamlining pneumonia diagnosis, thereby contributing to improved healthcare outcomes. The model has the potential to significantly impact the effectiveness of healthcare services. Its deployment could lead to expedited diagnoses, timely interventions, and ultimately, better patient outcomes. In essence, the project marks a significant stride towards the intersection of artificial intelligence and healthcare, embodying the transformative potential of deep learningin revolutionizing medical diagnostics and patient care. The Pneumonia Detection Model stands as a testament to the synergy between technology and medicine, paving the way for a future where cutting-edge advancements positively shape the landscape of healthcare.

9. Summary

In this comprehensive implementation, the code establishes a robust end-to-end training, and evaluating a binary image classification model. Leveraging the powerful DenseNet121 architecture through transfer learning, the codeprioritizes key components to ensure an effective and well-structured model.

The data preparation phase showcases meticulous organization, with defined directories for training, validation, and testing datasets. The choice of a standard image size of 224x224 pixels strikes a balance between computational efficiency and retaining crucial visual information. A noteworthy addition is the application of data augmentation exclusively to the training set, a strategic move that enhances the model's robustness by exposing it to various augmented versions of the training images. This augmentation strategy is instrumental in preventing overfitting and promoting the model's ability to generalize to diverse real-world scenarios.

Moving on to the model architecture, the code adeptly utilizes transfer learning with DenseNet121, a pretrained convolutional neural network. The exclusion of the top layers allows for customization to suit the binary classification task. Custom layers, including global average pooling and densely connected layers, are seamlessly integrated to fine-tune the model's output for binary classification. This architectural strategy capitalizes on the strengths of DenseNet121 while tailoring the model to the specific nuances of the binary classification problem.

The subsequent compilation and training phases showcase careful parameter selection,with the Adam optimizer and a specified learning rate of 0.0001. The use of binary cross entropy as the loss function aligns with the binary nature of the classification task. Model training unfolds over 10 epochs, striking a balance between achieving convergence and avoiding overfitting. The simultaneous monitoring of both training and validation datasets ensures a vigilant assessment of the model's performance, fostering early detection of potential issues.

The implementation's commitment to transparency and insight is evident in the training history visualization. Plots of accuracy and loss metrics over epochs offer an exact understanding of the model's learning dynamics. This visualization step empowers model developers to make informed decisions, providing a visual narrative of convergence behavior and potential areas for improvement.

The final phase of model evaluation on a separate test dataset completes the pipeline. A dedicated test data generator ensures consistency in preprocessing, and the evaluation metrics, including test loss and accuracy, deliver a quantitative assessment of the model's generalization capabilities. This rigorous evaluation step serves as a litmus test for the model's real-world applicability, transcending performance ontraining and validation sets.

In summary, this code shows how we can a pre trained model and apply transfer learning to get acceptable results.

10. References

[1] Vandecia Fernandes et al., “Bayesian convolutional neural network estimation for pediatric pneumonia detection and diagnosis”, Computer Methods and Programs in Biomedicine, Elsevier, 2021

[2] Hongen Lu et al., “Transfer Learning from Pneumonia to COVID-19”, Asia-Pacific on Computer Science and Data Engineering (CSDE), 2020 IEEE

[3] Sammy V. Militante et al., “Pneumonia and COVID-19 Detection using Convolutional Neural Networks”, 2020 the third International on Vocational Education and Electrical Engineering (ICVEE), IEEE, 2021

[4] Nanette V. Dionisio et al., “Pneumonia Detection through Adaptive Deep Learning Models of Convolutional Neural Networks”, 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC 2020), 8 August 2020

[5] Md. Jahid Hasan et al., “Deep Learning-based Detection and Segmentation of COVID-19 & Pneumonia on Chest X-ray Image”, 2021 International Information and Communication Technology for Sustainable Development (ICICT4SD), 27-28 February 2021

[6]https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[7] LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1989, 1, 541–551.

[8] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep

convolutional neural networks. Adv. Neural Inf. Process. Syst. 2012, 25, 1097–1105.

[9] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 618-626.

[11] L. Wang and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images," arXiv:2003.09871, 2020.