A

Project Report

on

"Pneumonia Detection and Classification using Deep Learning"

Submitted

In partial fulfillment of the requirements for the degree

of

Bachelors of Technology

In

Computer Science and Engineering

by

Mohit Jaisal (Enrollment No: 190104041)

Ritik Goyal (Enrollment No: 190104052)

Shikha Rai (Enrollment No: 190104058)

Under the supervision of

Dr. Vandana Dixit Kaushik

Associate Professor

Department of Computer Science and Engineering

To the

School of Engineering

Harcourt Butler Technical University, Kanpur

(Formerly Harcourt Butler Technological Institute, Kanpur)

(May, 2023)

ABSTRACT

The infectious illness known as Pneumonia is regularly a result of contamination because of a bacterium in the alveoli of the lungs. While an infected tissue of the lungs has an infection, it builds up pus in it. To find out if the patient has those illnesses, professionals perform bodily exams and diagnose their patients through Chest X-ray, ultrasound, or biopsy of lungs. Misdiagnosis, erroneous treatment, and if the disease is overlooked will result in the patient's lack of lifestyle. The progression of Deep learning contributes to aid in the decision-making procedure of specialists to diagnose sufferers with these illnesses. The look employs a bendy and efficient technique of deep learning of applying the model of CNN in predicting and detecting a patient unaffected and affected with the sickness using a chest X-ray photograph. The take a look at utilizing an accrued dataset of 20,000 images using a 224x224 photograph decision with 32 batch length is applied to prove the overall performance of the CNN model being educated. The trained-version produced an accuracy charge of 95% at some point of the overall performance training. Based on the end result of the experiment carried out, the research study can detect and are expecting COVID-19, bacterial, and viral-pneumonia sicknesses based totally on chest X-ray photos.

CERTIFICATE

This is to certify that the report presented here, titled "Pneumonia Detection and Classification using Deep Learning", in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology and submitted to Computer Science and Engineering Department of Harcourt Butler Technical University, Kanpur is an authentic record of work carried out under the guidance of Dr. Vandana Dixit Kaushik, Associate Professor, Computer Science and Engineering Department, HBTU Kanpur.

The matter embodied in this report has not been submitted by me for the award of any other degree or diploma.

This is to certify that the above statements made by the candidates are correct to the best of our knowledge.

Date: 04-05-2023

Dr. Vandana Dixit Kaushik
Associate Professor
CSE Department
HBTU Kanpur

ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on

'Pneumonia Detection and Classification using Deep Learning'

We would like to take this opportunity to thank our internal guide **Dr. V.D. Kaushik,** Associate Professor, department of Computer Science and Engineering, for her un- conditional guidance. We are really grateful to them for her kind support and her valuable suggestions were very helpful

An erudite teacher, a magnificent person and a strict disciplinarian, we consider our- selves fortunate to have worked under her supervision

We are highly grateful to **Dr. Narendra Kohli,** Head of Department, Computer Science and Engineering, Harcourt Butler Technical University, for indispensable support, suggestions.

We admit thanks to project coordinators, Department of Computer Science and Engineering, Harcourt Butler Technical University for giving us such an opportunity to carry on such mind stimulating and innovative Project.

Mohit Jaisal (HBTU190104041)

Ritik Goyal (HBTU190104052)

Shikha Rai (HBTU190104058)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS USED

ML – Machine Learning

VGG – Visual Geometry Group

Res Net – Residual Neural Network

IEEE – Institute of Electrical and Electronics Engineers

Colab – Colaboratory

CONTENTS

1. INTRODUCTION TO PNEUMONIA DETECTION USING DEEP LEARNING MODELS

1.1 Pneumonia impacts all the elderly and young people everywhere. With the high growth in the popularity of neural networks, engineers and researchers have been able to find state-of-the-art products for computer vision. Artificial Intelligence helps us to automate analysis techniques, which is only possible now because of the technology of Deep Learning. Exposure to Pneumonia is quite high for many people, mainly in economically underdeveloped and developing countries where the majority are deprived of a nutritious diet. The World Health Organization states that more than 4 million untimely deaths per year occur from diseases caused by air pollution.

The purpose of this project is to build an AI network, which takes the pixel values as input for a given X-Ray image and then proceeds to perform linear operations and activations on each of them. Then by taking all the above operations, and then multiplying them with each layer within the neural network, and the number of nodes. Suddenly you have millions of operations. By applying determination, they can be more efficient in these tasks. The scope is to develop a model that will identify whether a patient is having Pneumonia or not bypassing the chest X-Ray images through the Deep Learning model

1.2 MOTIVATION BEHIND PROJECT TOPIC

The main motivation behind this research is to identify Pneumonia just by using the X-Ray images of the patients. as doctors must do a lot of certain tests to identify if the patient has Pneumonia or not. To solve the cumbersome problem, an ensemble of two deep learning models is developed, to make the work of the doctors simpler.

1.3 AIM AND OBJECTIVE OF THE WORK

Aim:

To provide an efficient and effective solution over the conventional way of detecting pneumonia disease using Deep learning.

Objectives:

To learn different biomedical terms related to pneumonia disease.

To learn the different scenarios of pneumonia disease (viral or bacterial).

To learn different methods of data acquisition. to know about different image processing pre-trained models.

To build a web application to analyse Pneumonia disease using chest X-ray.

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 alveoli fill with fluid or pus, making it hard to respire the system takes benefit of transfer learning on properly-studied of deep learning to know the type of virus and validates the models on a big quantity of data sets, and finally transfers the models and insights learned in training and validation to a brand new data set in a comparable area, which, in this have a look at, is a new infectious fast-growing sickness, 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 analyse the different scenarios of pneumonia disease(viral or bacterial).
- It can be used by any user or Radiologists for testing purpose

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 under - stand.

3. User-Friendly.

4. 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, com- putter 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
- CUDA v11.5
- CuDNN v8.3

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 its made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. There is no one sin- gle 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.

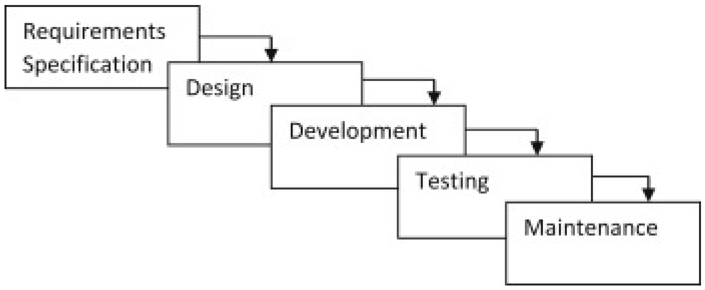


Figure 3.1: Waterfall Model

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 finalised 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.

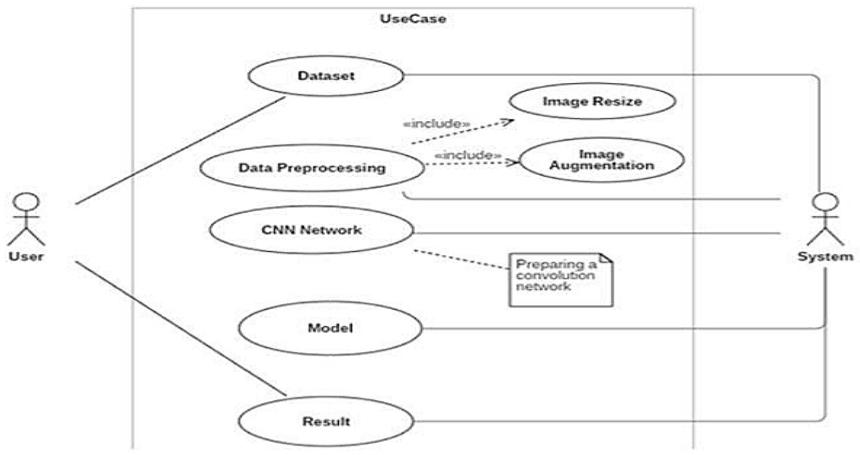
Following UML diagrams have been designed for the project:

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 student, teacher, system, and cloud. The various use cases used in the diagram are teacher login, view attendance, take attendance, capture photo, process image, take another photo.



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 user, student, teacher, Image, Cloud, Face Recognition.

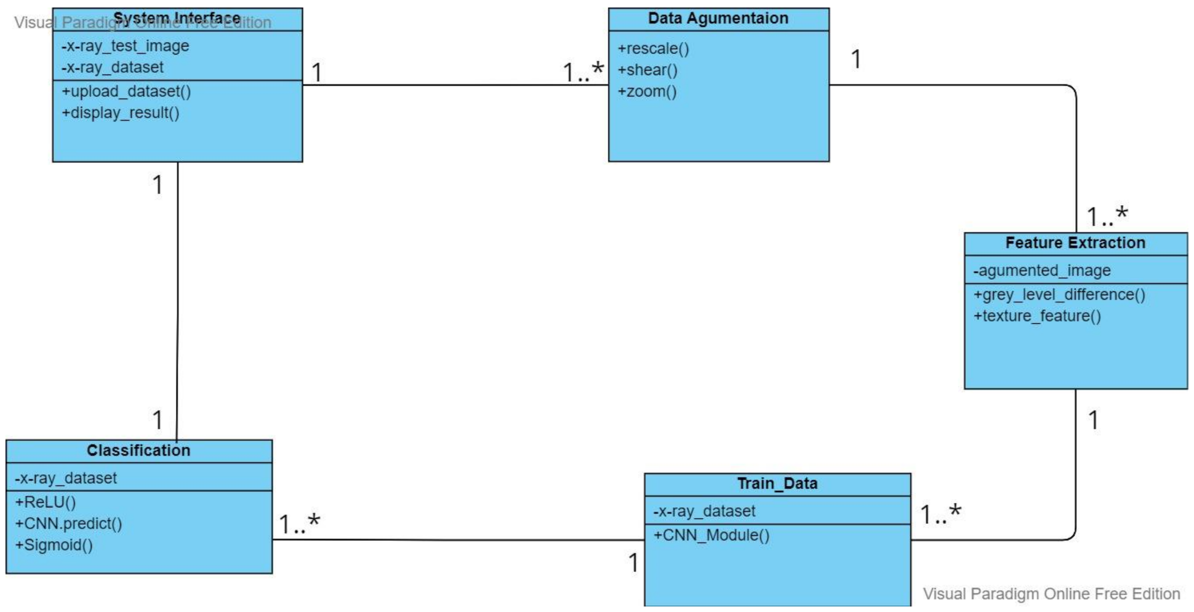


Fig 4.2. Class Diagram

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

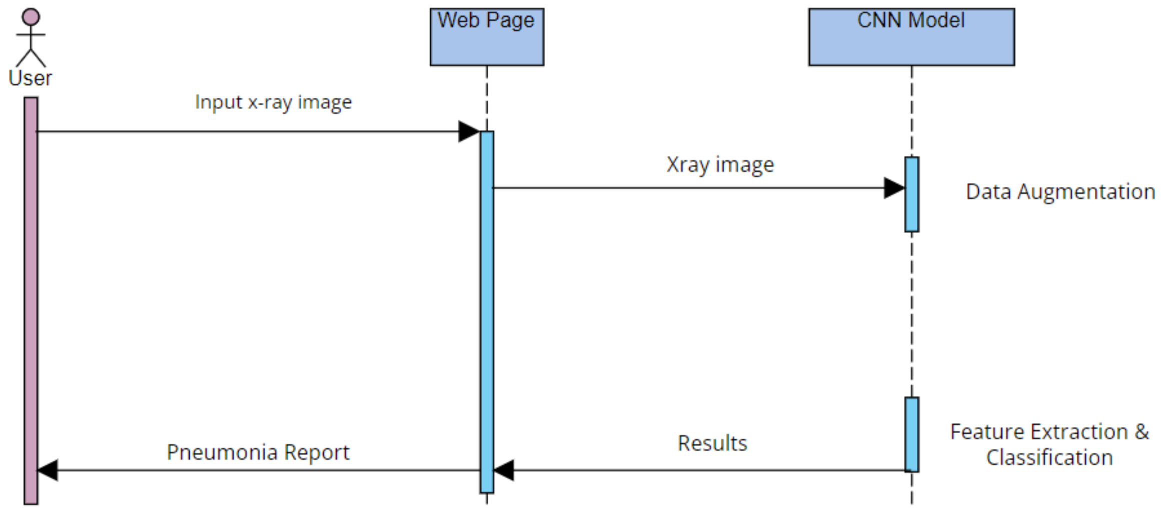


Fig 4.3. Sequence Diagram

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.

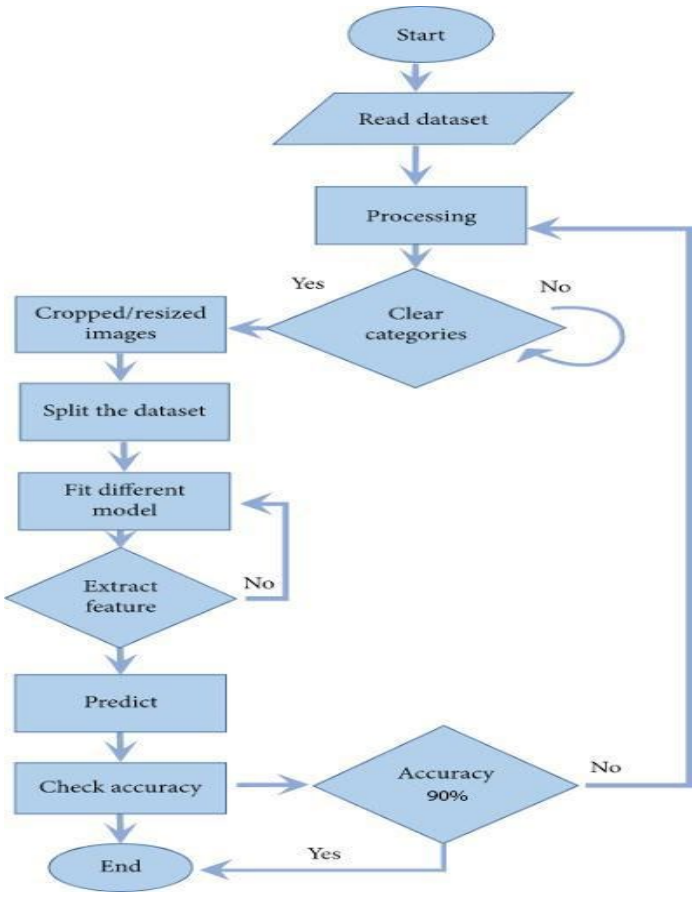


Fig 4.4. Activity Diagram

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.

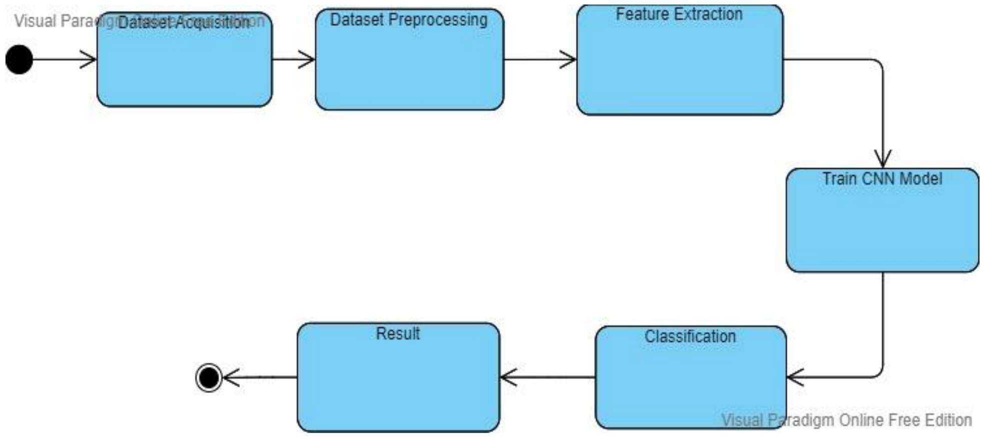


Fig 4.5. State Diagram

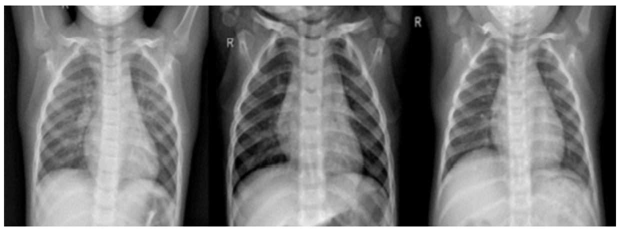
5. ALGORITHM AND METHODS

5.1 Data Acquisition

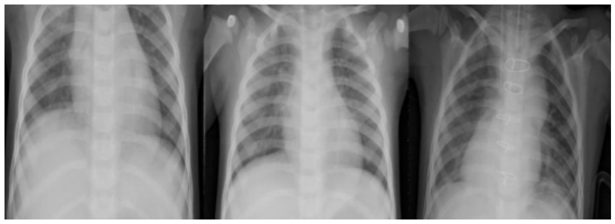
The proposed database used to evaluate the performance of the model contains a total of 5863 X-ray photos from the Kaggle.

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 database 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 by 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./255
Zoom Range	0.2
Shear Range	0.2
Horizontal Flip	True

Table 2

5.3 Proposed Network

In this study, we designed a CNN model to extract the features of chest X-ray images and use those features to detect if a patient suffers from pneumonia. In Our CNN Architecture, we began it with a lower filter value of 32 and increased it layer-wise. Constructed the model with a layer of Conv2D followed by a layer of Carpooling. The kernel size is preferred to be an odd number like 3x3.

Tanh, ReLU, etc. can be used for activation function, but ReLU is the most preferred activation function. input shape takes in image width & height with the last dimension as a color channel. then we Flatten the input after CNN layers and added ANN layers.

$$S(x) = \frac{1}{1 + e^{-x}}$$

f(x)=max(0,x)

S(x) = Sigmoid

f(x) = ReLU

Used activation function as SoftMax for the last layers(ANN Layers) also defined units as the total number of classes and used sigmoid for binary classification and set unit to 1.

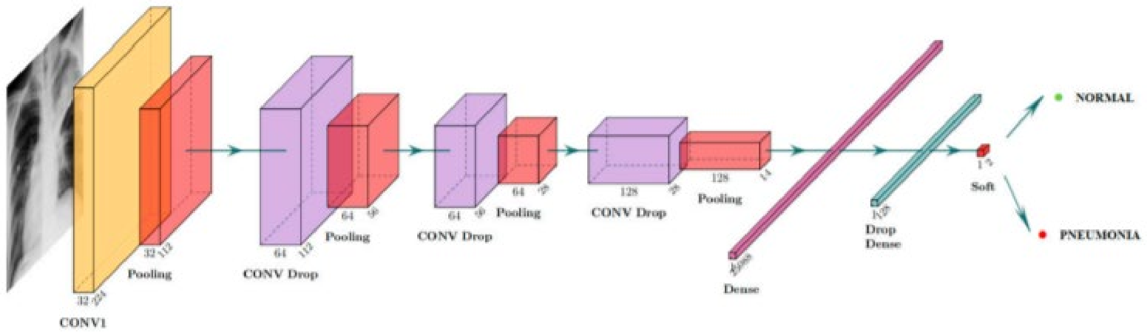


Fig 5.3. Details of proposed DL mode

5.4 WORKING THEORY

Convolutional neural networks refer to a sub-category of neural networks: they, therefore, have all the characteristics of neural networks. However, CNN is specifically designed to process input images. Their architecture is then more specific: it is composed of two main blocks.

Conv Layers

The first block makes the particularity of this type of neural network since it functions as a feature extractor. To do this, it performs template matching by applying convolution filtering operations. The first layer filters the image with several convolution kernels and returns "**feature maps**", which are then normalized (with an activation function) and/or resized.

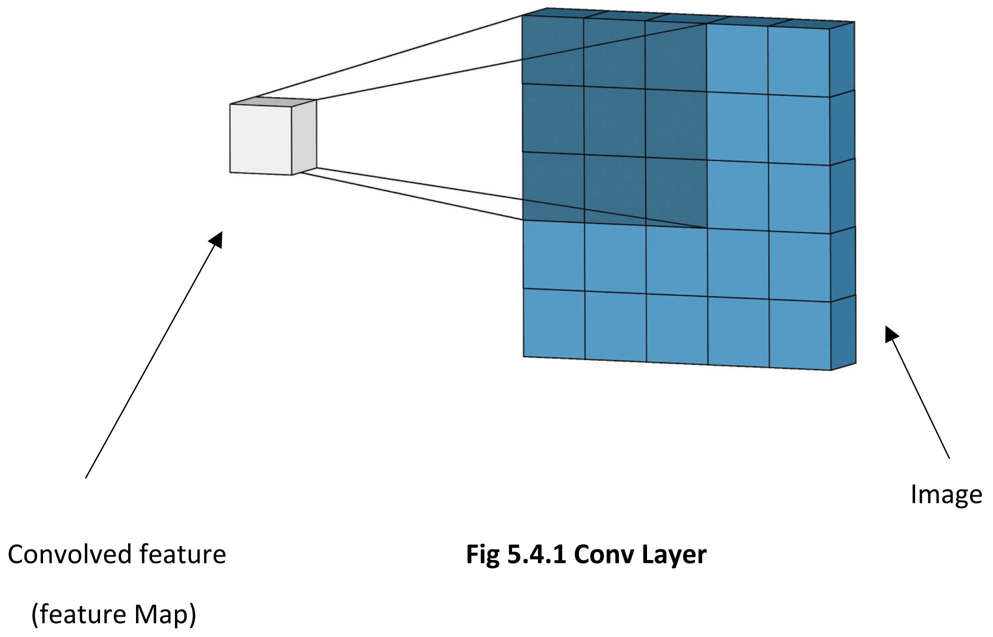


Fig 5.4.1 Conv Layer

Pool Layers

The second block is not characteristic of a CNN: it is in fact at the end of all the neural networks used for classification. The input vector values are transformed (with several linear combinations and activation functions) to return a new vector to the output. This last vector contains as many elements as there are classes: element I represents the probability that the image belongs to class I. Each element is therefore between 0 and 1, and the sum of all is worth 1. These probabilities are calculated by the last layer of this block (and therefore of the network), which uses a **Sigmoid function** (binary classification) or a **RELU function** (multi-class classification) as an activation function.

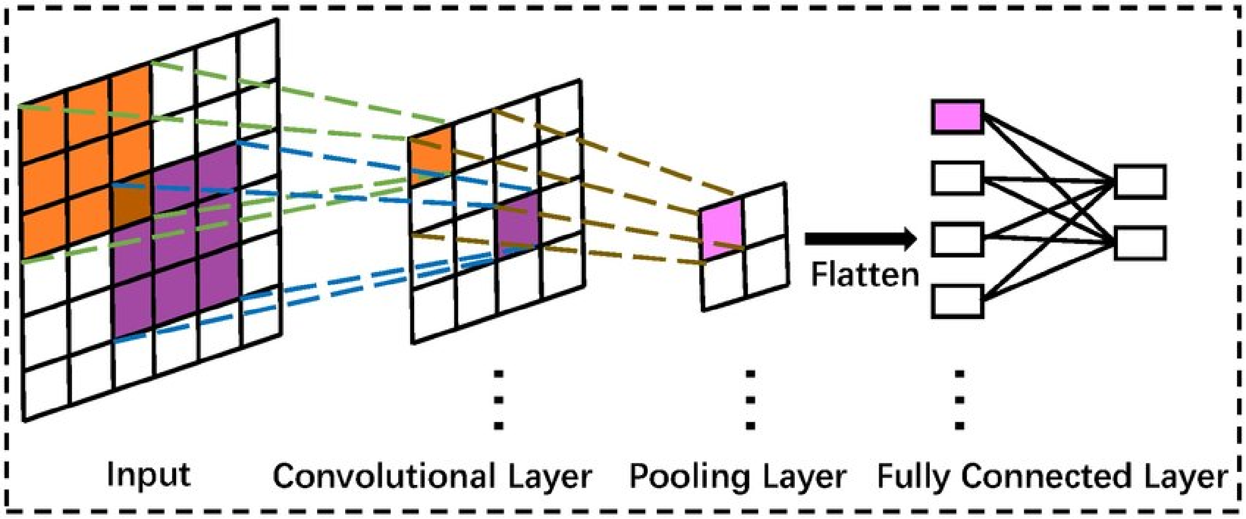


Fig. 5.4.2 Mathematics of Conv and Pooling Layer

Activation Function

An activation function is a mathematical function applied to the output of a neuron in a neural network. It introduces non-linearity to the network, allowing it to learn and model complex relationships between inputs and outputs. Common activation functions include the sigmoid, ReLU, and tanh functions.

ReLU

Rectified linear unit is most widely used and preferred activation function right now which ranges from 0 to infinity, All the negative values are converted into zero.

$$f(x)=max(0,x)$$

Sigmoid

The sigmoid function also called a logistic function. having a characteristic that can take any real value and map it to between 0 to 1. It decides which value to pass as output and what not to pass.

$$S(x) = \frac{1}{1+e^{-x}}$$

VGG16 Model

VGG16 is a popular convolutional neural network architecture for image classification tasks. It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers use small 3x3 filters with a stride of 1 and are followed by max pooling layers. The fully connected layers at the end of the network perform the classification task. The VGG16 model has achieved state-of-the-art performance on several image recognition benchmarks, making it a popular choice for computer vision tasks.

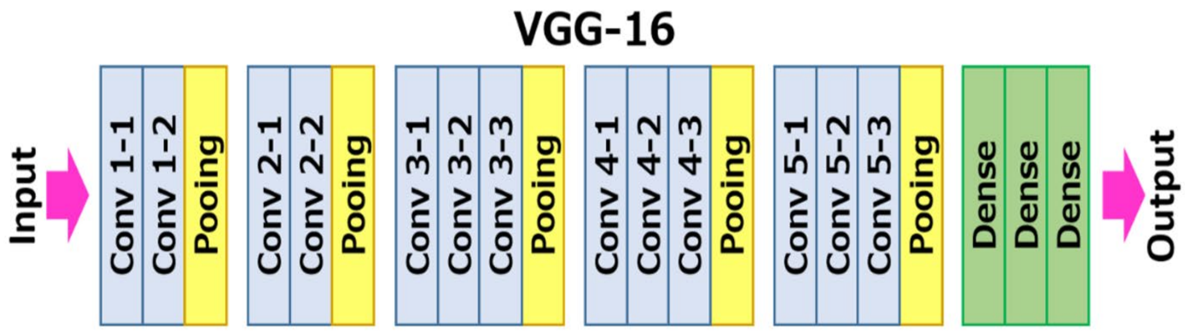
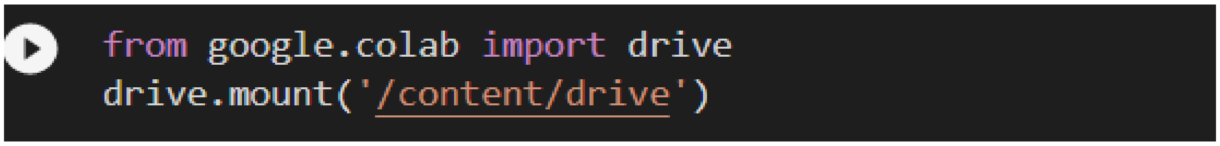


Fig. 5.4.3 VGG16 Model Sequence

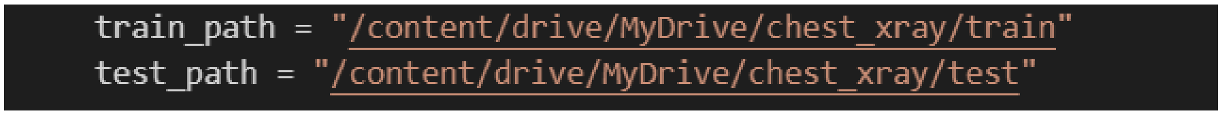
6 Implementation and Testing

6.1 Data Acquisition



```python
from google.colab import drive
drive.mount('/content/drive')
```

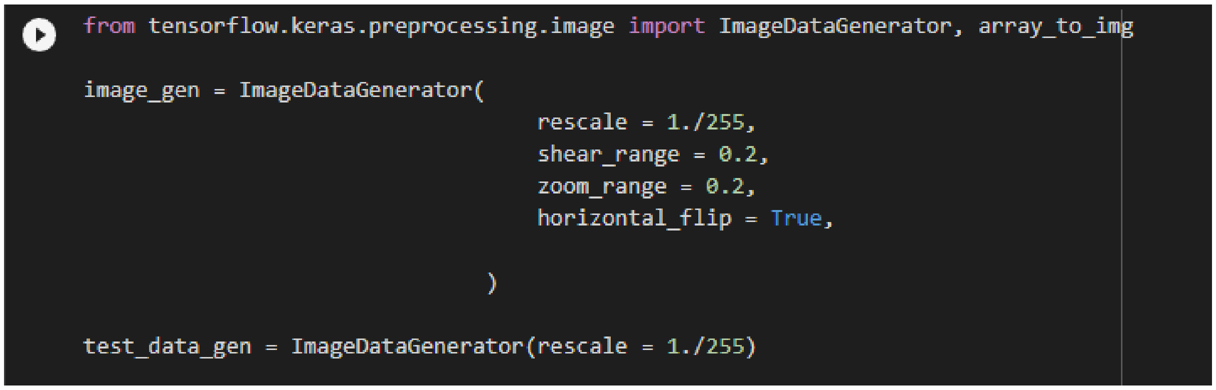
Fig 6.1.1 Mount Drive



```python
train_path = "/content/drive/MyDrive/chest_xray/train"
test_path = "/content/drive/MyDrive/chest_xray/test"
```

Fig 6.1.2 Import Train and Test Data

6.2 Data Augmentation



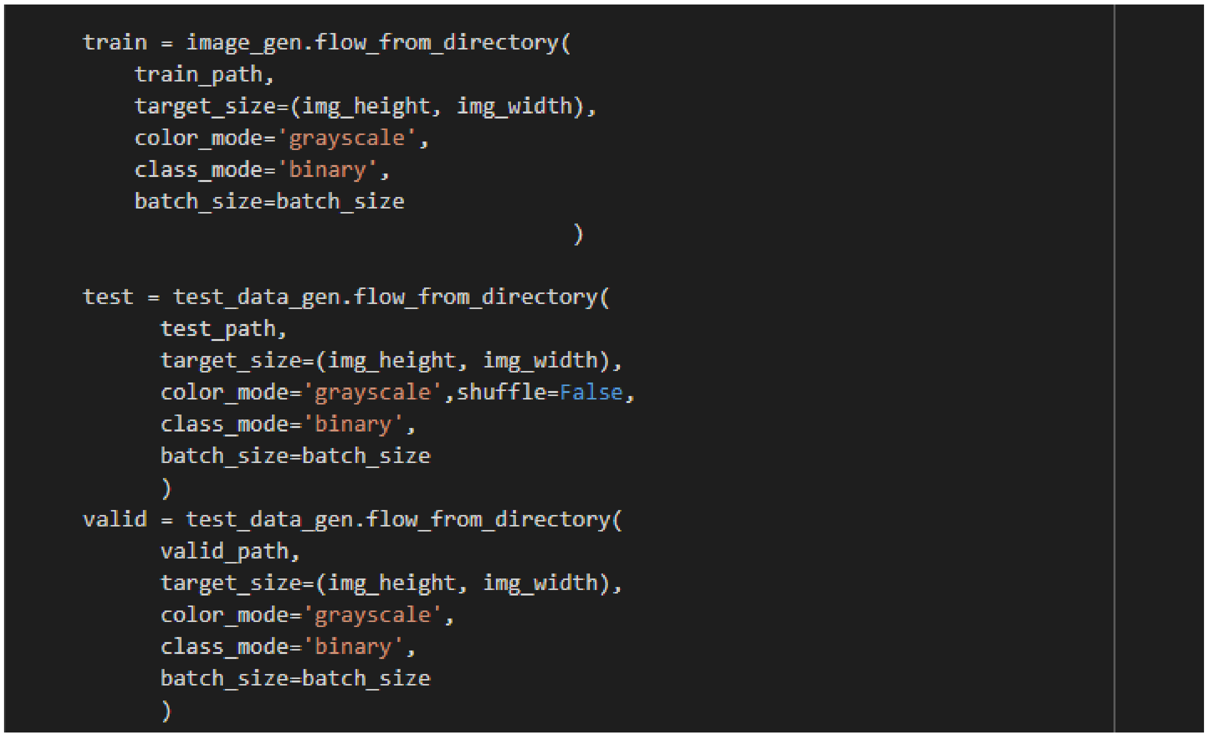
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img

image_gen = ImageDataGenerator(
                                   rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True,

                               )

test_data_gen = ImageDataGenerator(rescale = 1./255)
```

Fig 6.2.1 Defining parameters for Data augmentation



```
train = image_gen.flow_from_directory(
    train_path,
    target_size=(img_height, img_width),
    color_mode='grayscale',
    class_mode='binary',
    batch_size=batch_size
                                      )

test = test_data_gen.flow_from_directory(
      test_path,
      target_size=(img_height, img_width),
      color_mode='grayscale',shuffle=False,
      class_mode='binary',
      batch_size=batch_size
      )
valid = test_data_gen.flow_from_directory(
      valid_path,
      target_size=(img_height, img_width),
      color_mode='grayscale',
      class_mode='binary',
      batch_size=batch_size
      )
```

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.

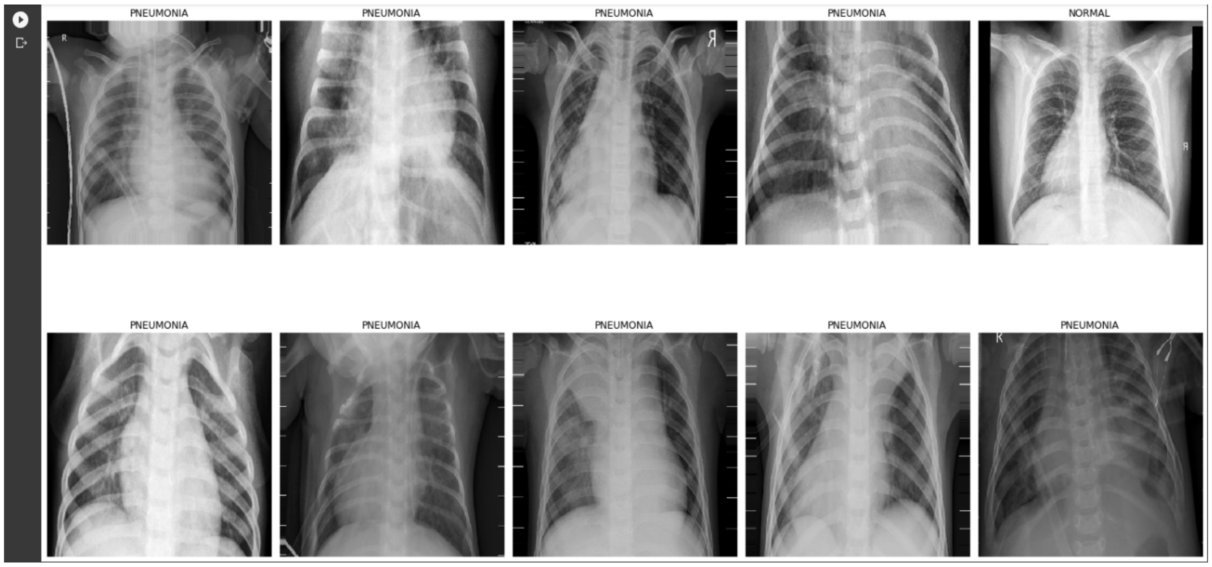


Fig 6.2.3 Augmented Dataset images

We have analyzed the performance of these bellow CNN architectures namely; Alex Net, ResNet-50, and VGGNet-16. [3]

CNN Model	Batch Size	Criterion	Normal	Infected with Pneumonia
VGG 16	32	Precision	0.92	0.89
		Recall	0.81	0.96
		F1-score	0.86	0.92
ResNet-50	32	Precision	0.85	0.85
		Recall	0.87	0.84
		F1-score	0.83	0.80
AlexNet	32	Precision	0.78	1.00
		Recall	1.00	0.88
		F1-score	0.85	0.94

Table 3. Analysis of model Performance for CNN Models with 32 Batch Size

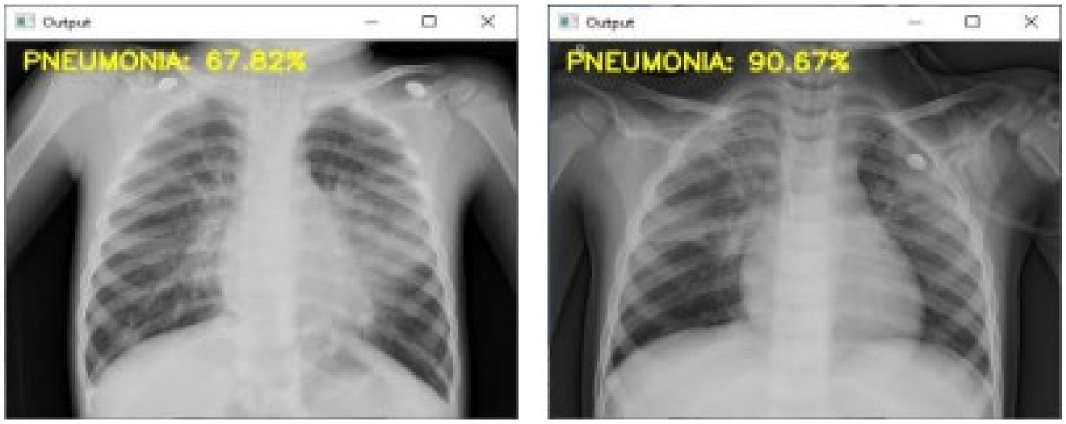


Fig 6.3 ResNet-50

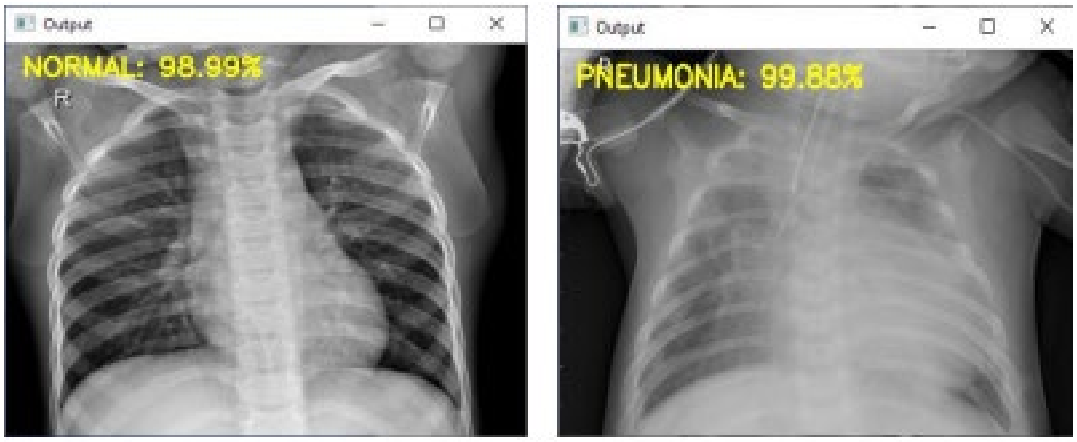


Fig 6.4 AlexNet

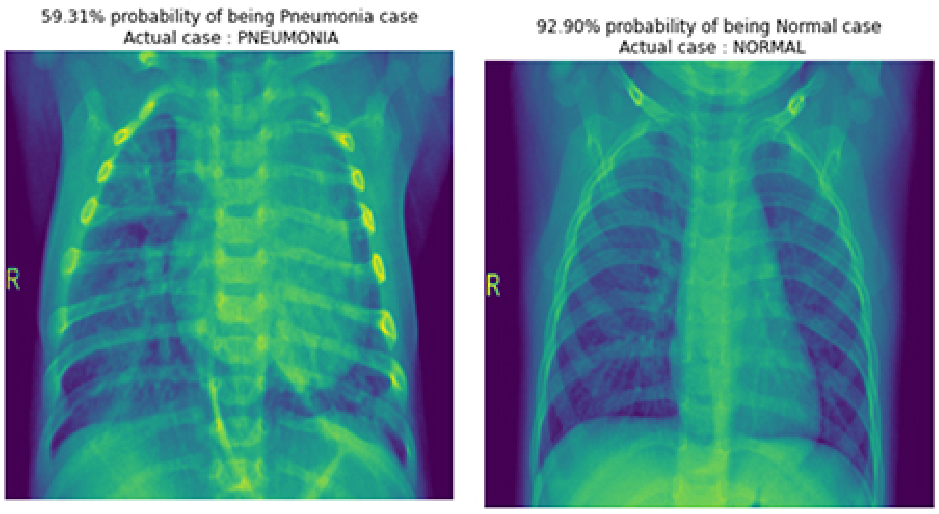

59.31% probability of being Pneumonia case
Actual case : PNEUMONIA
R
92.90% probability of being Normal case
Actual case : NORMAL
R

Fig 6.5 VGG16

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.

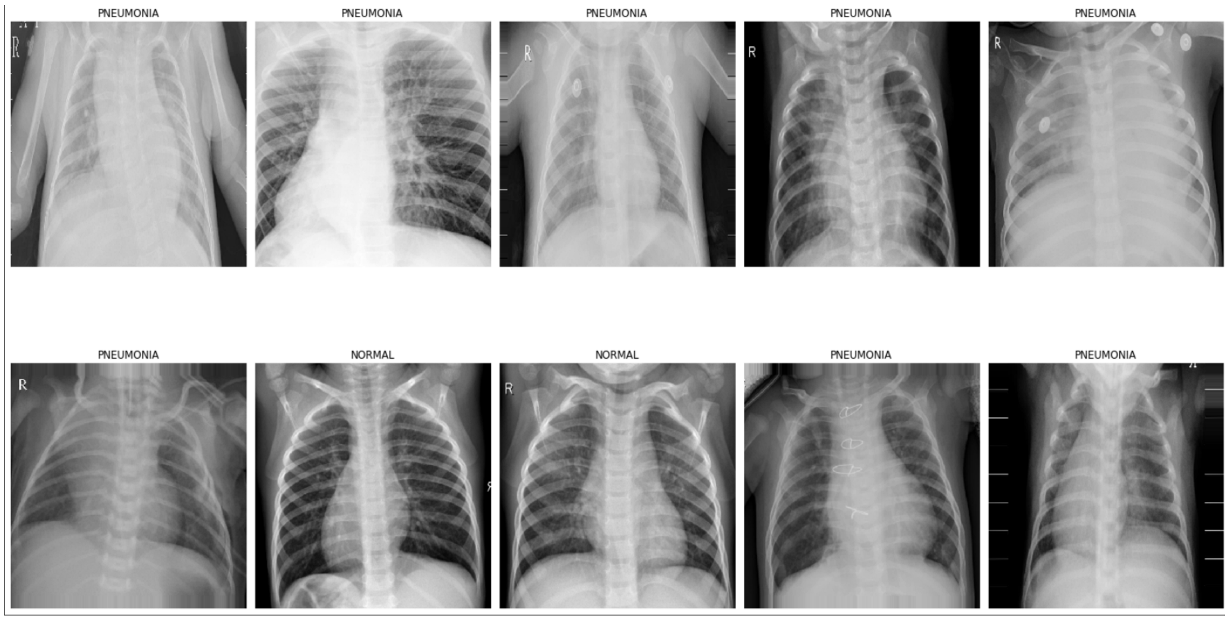


Fig. 7.1 Augmented Dataset Images

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 utilised 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.

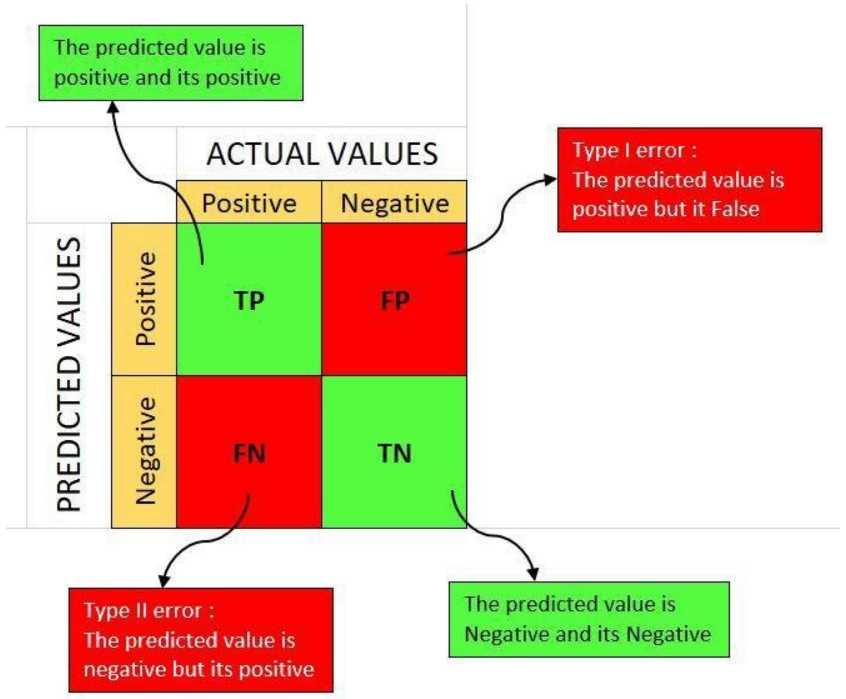
height shift range - Vertically shifts the image.

brightness range - a 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 Confusion Matrix

Let's interpret the output of the confusion matrix. The upper left (TP) denotes the number of images correctly predicted as normal cases and the bottom right (TN) denotes the correctly predicted number of images as cases of pneumonia. As Pneumonia case, the upper right denotes the number of incorrectly predicted images but were actually normal cases and the lower left denotes the number of incorrectly predicted Normal case images but were actually Pneumonia case.


The predicted value is positive and its positive
ACTUAL VALUES
Positive
Negative
Type I error : The predicted value is positive but it False
PREDICTED VALUES
Positive
Negative
TP
FP
FN
TN
Type II error : The predicted value is negative but its positive
The predicted value is Negative and its Negative

7.2.1 Mathematical Model

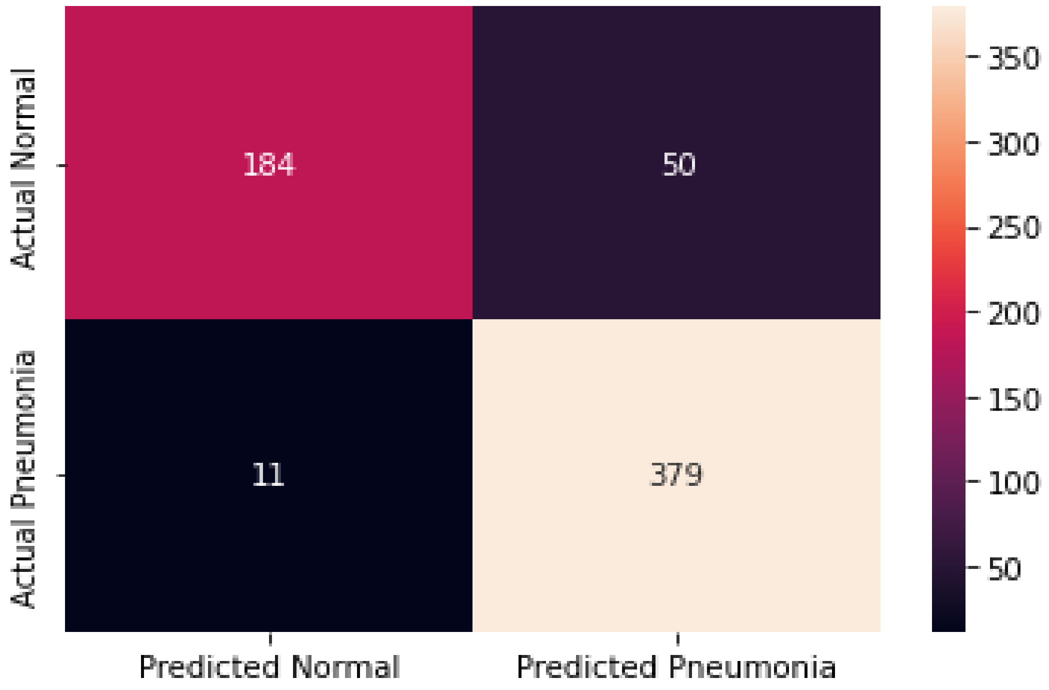


Fig. 7.2.2 Confusion Matrix

7.3 Visualizing some predicted images with percentage %

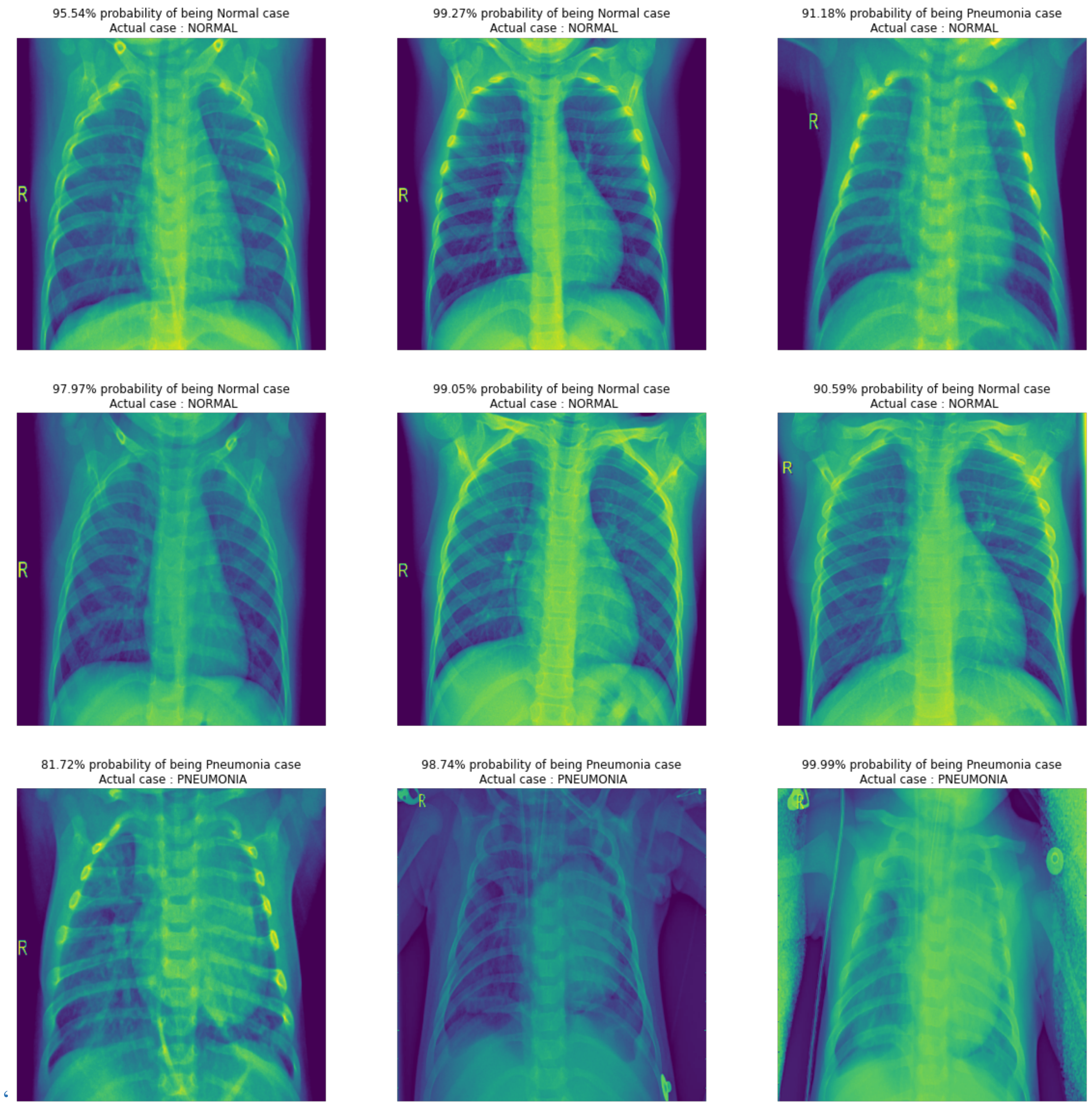

95.54% probability of being Normal case
Actual case : NORMAL
99.27% probability of being Normal case
Actual case : NORMAL
91.18% probability of being Pneumonia case
Actual case : NORMAL
97.97% probability of being Normal case
Actual case : NORMAL
99.05% probability of being Normal case
Actual case : 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.

7.4 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.

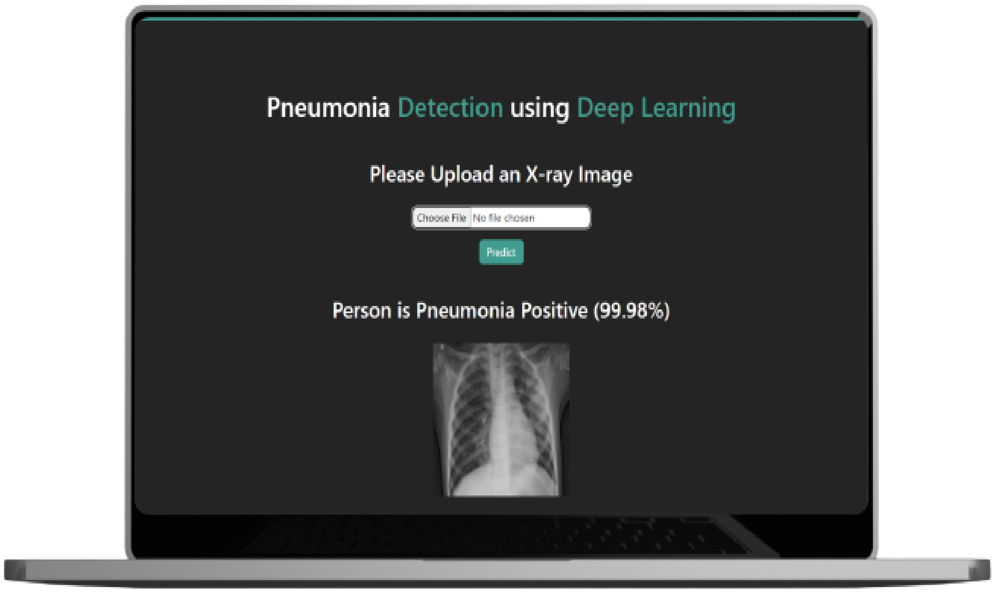

Pneumonia Detection using Deep Learning
Please Upload an X-ray Image
Choose File
No file chosen
Predict
Person is Pneumonia Positive (99.98%)

Fig. 7.4 Frontend User Interface with Flask

8 Conclusion

This study describes a CNN-based model aiming to diagnose pneumonia on a chest X-ray image set. The contributions in this paper are listed as follows. We designed a CNN model to extract the features from original images or previous feature maps, which contained only six layers combining ReLU activation function, drop operation, and max-pooling layers. The results of the obtained accuracy rate of 92.07% and precision rate of 91.41%, shows that our proposed model performs well in comparison to state-of-the-art CNN model architectures. To illustrate the performance of our proposed model, several comparisons of different input shapes and loss functions were provided.

In the future, we will continue the research to explore more accurate classification architectures to diagnose two types of pneumonia, viruses, and bacteria. According to the description discussed above, the CNN-based model is a promising method to diagnose the disease through X-rays.

9. Bibliography

[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.