MEDISAFE – IDENTIFY EXACT LUNG DISEASE AMONG OTHER LUNG DISEASE USING IMAGE PROCESSEING

2022 - 143

Final Report

B.Sc. (Hons) Degree in Information Technology
Specializing in Information Technology

Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

September 2022

MEDISAFE – IDENTIFY EXACT LUNG DISEASE AMONG OTHER LUNG DISEASE USING IMAGE PROCESSEING

2022 - 143

Final Report

Rasuni Wageesha H.A

IT19015040

Mr. Supunya Swarnakantha

B.Sc. (Hons) Degree in Information Technology

(Specialization in Information Technology)

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2022

Declaration

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student Id	Signature
Rasuni Wageesha H.A	IT19015040	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

………………………………………………….. ……………………………………….

Signature of the Supervisor Date

Abstract

The biggest issue in medicine is image diagnosis. Modern medical facilities provide early and accurate disease diagnosis, reducing the need for needless treatment procedures. In this approach, the image processing and identify of the which disease affected. The human body, which is made up of many parts including legs, hands, bones, and lungs, has many distinct health issues. A radiologist may advise patients to get an x-ray of their lungs for diagnostic purposes because lung illness is one of the key issues that might arise from excessive lung use. The most used method of screening for heart and lung failure is a simple X-ray, which accounts for more than 500,000 fatalities per year in the countries. One of the most interesting areas of software development has always been figuring out how to automate diagnosis from chest x - rays. The main purpose means identify lung diseases. Mainly use Machine Learning (ML), Image Processing, and Convolutional Neural networks (CNN). CNN used to analyse visual imagery and ML like a part of Artificial Intelligence (AI)because ML is the study of computer algorithms that improve automatically through experiences. Finally, 1000 various lung images in the user acceptance testing. The overall component accuracy scored 97%. That accuracy was dependent on the lung x ray images. Then overall accuracy was assigned based on the user acceptability testing results and system testing results. This System trains the lung diseases by using lung disease x ray images and identify the lung diseases when user insert the x ray images of lung image. Patients getting various diseases like lung opacity, covid19 and pneumonia.

key words: Medical image processing, lung segmentation, Convolutional Neural network,

Acknowledgement

It is with great pleasure that I express my gratitude for the guidance and help of everyone who contributed to the success of my research.

First and foremost, I want to thank my supervisor, Mr. Ravi Supunya, and my co-supervisor, Ms. Samantha Rajapaksha, for their great supervision, expertise, fast constructive criticism, and useful suggestions that helped me complete this project successfully.

I feel very fortunate to have them as my research supervisors, and I owe them a great debt of gratitude. Also, I want to express my heartfelt gratitude to my team members for their tireless efforts in making this research project a success.

I should last like to thank my family and friends for their financial support and other means of effectively carrying out this research

Table of content

List of Figures

List of Tables

List of abbreviations

Abbreviation	Description
CNN	Convolutional Neural Network
DL	Deep Learning
IDE	Integrated Development Environment
ML	Machine Learning

1. Introduction

1.1 Background & Literature survey

A society is prosperous when all its citizens are healthy. If one wants to be happy, it is crucial to preserve their health. Only a healthy body can be mentally healthy and having both has a beneficial effect on how well people perform. People are less conscious of their health today. They neglect to take the proper precautions to maintain their health because of their busy lives, and they have less awareness of their health situation. People place little value on their health and find it time-consuming to visit hospitals for check-ups. Health has no place in an overscheduled existence. The majority of those who make up the working class of the population state that their busy schedules prevent them from getting regular check-ups and that they ignore any symptoms of discomfort in their bodies until they get too severe.

A healthy person can then have a better lifestyle and make a bigger contribution to a better society. But because of the hectic lifestyles people lead in today's culture, their health has become a concern that is frequently disregarded in favour of the pressure to earn money and finish the enormous amounts of work they are assigned.

By training lung diseases using sample lung disease x-ray images, this system can identify lung diseases when a user inserts x-ray images of the lungs. Patients are being diagnosed with disorders such as lung opacity, COVID-19, and pneumonia.

Phat Nguyen Kieu, Hai Son Tran and Thai Hoang Le discovered the suggested model employs several Convolutional Neural Networks, often known as multi-CNNs, to choose the input image. The ConvnetJS package was used to create each convolutional neural network that makes up the multi-CNN [1]. The given model's output is Normal/Abnormal density. In this study, we also suggest a technique that we call Fusion rules for combining the outputs of the model's individual parts.

Detection of Pneumonia clouds in Chest X-ray using Image processing approach by Abhishek Sharma, Daniel Raju, and Sutapa Ranjan the method described in this article uses solely image processing methods to identify pneumonia clouds in chest X-rays (CXR). For we worked on 40 analog chest CXRs from individuals who were both healthy and who had pneumonia [2]. Where there is a lack of skilled radiologists, computer assisted illness detection using CXR is always very beneficial. Such systems can be incredibly helpful by automatically identifying patients who require immediate medical attention and additional diagnosis in nations like India where we do not have skilled radiologists in rural areas.

In this research paper detecting COVID-19 and community acquired pneumonia using chest CT scan images with deep learning for the purpose of identifying COVID19 and community acquired pneumonia (CAP) using chest computed tomography (CT) scan pictures, we suggest a two-stage Convolutional Neural Network (CNN) based classification framework [3].The suggested two-stage framework had a classification accuracy of 89.3% for the fine-grained three-class classification of COVID-19, CAP, and normal, and over 94% for the binary classification task of infectious vs. non-infectious in the CT scan pictures.

The purpose of this study is to create new image processing models that may be applied to describe individual opacities and opacity density within chosen regions of interest (ROI) [4]. The AM-FM paradigm has demonstrated a lot of promise for describing specific opacities. The results show strong agreement with the ILO requirements, as well as the regular Pattern Spectrum and the radial Design Spectrum both offer an automated way to measure opacity density.

X-ray scans of the chest will be used to locate the pneumonia and assess its severity using deep learning. Methods: RSNA Pneumonia Detection Challenge data [5]. In order to determine the severity percentage in a lung x-ray image of pneumonia, it is intended to take use of previous research and create a more effective, highly accurate deep learning model.

The goal is to examine how three contrast enhancement methods—histogram equalization (HE), contrast constrained adaptive HE, and exposure fusion framework—affect the ability to diagnose pneumonia [6]. HE, CLAHE, and EFF are only a few of the image enhancement methods that have been used in this study. In order to identify pneumonia using a CXR, these three improved images serve as the input images for the VGG16-based CNN. Using EFF image enhancement as a pre-processing method result in the best performance in terms of training and validation losses and accuracies.

For the automatic identification of pneumonia from chest X-ray images, we have proposed a combination technique in our study that combines Image Processing and either VGG-16 or VGG-19, variations of Deep Convolutional Neural Network. For the purpose of assessing our model, we used the Mendeley OCT and Chest X-Ray dataset [7]. In this study, we used the VGG-16 and VGG-19 networks, following by ubiquitous image processing, to identify pneumonia. These networks outperformed the transition InceptionV3 by 3.4% and 3.1%, respectively. In the future, we'd like to experiment with different convolution network topologies to create a pneumonia detector that is more effective.

A tried-and-true machine learning (ML) method is employed in this study. In order to classify data, the model was developed using the SVM (Support Vector Machine) method. This study aimed to investigate how quickly or accurately COVID19 chest X-ray or CT pictures might be identified using machine learning, image processing, image segmentation, and feature extraction [8]. Global healthcare systems are nearing breaking point due to the significant number of deaths brought on by the coronavirus pandemic. By detecting the COVID-19 virus earlier, a faster, easier, and less expensive technology could potentially save lives and lighten the workload of medical personnel. Machine learning has been critical in the detection of COVID-19 by using IP techniques on X-ray images. Furthermore, IS was applied K-Means clustering. In this study, an ML system for the highly accurate and straightforward identification of COVID-19 was devised and implemented by combining the features gathered using the PCA approach.

The process of having a professional diagnose pneumonia using chest X-ray pictures is time-consuming and less precise. To extract features from chest X-ray images and classify the images to determine whether a person has pneumonia, we suggest various deep convolution neural network (CNN) architectures in this research [9]. Using images from chest X-rays, we offer two CNN architectures in this study that were created from the ground up to diagnose pneumonia. Techniques for data augmentation are employed to prevent overfitting.

This proposed system by Tejasvi Raj Pant, Ravi Kiran Aryal, Tribikram Panthi, Milan Maharjan and Basanta Joshi for this study uses a convolution neural network to analyse chest X-rays to diagnose patients' illnesses[10].In this study, the diagnosis of seven thoracic diseases—Atelectasis, Consolidation, Effusion, Mass, Nodule, Pleural Thickening, and Pneumothorax—was made with an accuracy of more than 75% when individual diseases were taken into account and more than 60% when the problem was viewed as a multilabel classification problem (multi-classification problem).

1.2 Research Gap

The research papers mentioned above can only distinguish one kind of lung disease. The following common areas have generated a lot of research: One disease is the focus of most research papers.

- Get some images of diseases by going.
- By way of one disease

That crucial portion of the input images is covered by the suggested web application. With the help of this procedure, you may determine the specific type of lung disease that has been identified in cases of lung opacity, wise, pneumonia, and COVID-19, as well as provide suggestions for the patient.

The purpose of this study was to demonstrate whether the chest X-ray image produced by contrast improvement was useful in identifying pneumonia. A convolutional neural network is the approach that has been suggested for diagnosing pneumonia. With the help of 8 CNN models, the CLAHE corrected chest X-ray images was trained [11]. The Convolutional Neural Networks technique was used to train and diagnose pneumonia from the findings of contrast-enhanced chest X-rays.

In this paper Jianpeng An, Qing Taizhong Qu and Zhonghe Gao proposed to improve the cross-domain lung segmentation task as the prior knowledge to the classification network, we present a multi-scale adversarial domain adaptation network (MS-AdaNet) [12]. With the use of this study, the diagnosis of seven thoracic disorders—Atelectasis, Consolidation, Effusion, Mass, Nodule, Pleural Thickening, and Pneumothorax—can be made with an accuracy of over 60% when it comes to multilabel classification tasks and above 75% when it comes to individual diseases (multi-classification problem).

Table 1:Comparison Table

Research products	Features		
	Identify lung issues	Identify several lung diseases	Provide suggestions
Research A	✓	✗	✗
Research B	✗	✗	✗
Research C	✓	✗	✓
Proposed system (MediSafe)	✓	✓	✓

2. Research Problem

Lung-related illnesses are currently recognized. As an illustration, take COVID-19, which is currently paired with lung opacity and pneumonia. Each disease has its own unique peculiarities in how it spreads, although lung-related disorders remain invisible to the human eye. As a result, I'd like to describe a technique called image processing in this component to find lung-related disorders.

This method is accessible to all of you. Then upload this system after taking an x-ray of the chest. Once you know what type of lung illness you have, you can take appropriate action. Are you aware that you have the option to do this before seeing your specialist? The unique aspect of lung disease is the ability to identify the specific lung disease from which you are suffering.

Most of the systems that you are familiar with only look for one disease. By using this technique, you can determine the type of disease for which you have received a diagnosis.

The 2019 coronavirus disease, also known as COVID-19, has spread worldwide. Severe Acute Respiratory Syndrome Coronavirus 2 is the beta coronavirus responsible for the illness (SARS-CoV-2). The enormous number of fatalities and affected people around the world provide insight into the disease's severity. A rapid diagnosis will enable improved illness management. Diagnostic tests are readily available, but their availability of testing equipment and their turnaround time are limitations. Ct Scanning (CT) is a type of radiological examination that can be used to diagnose disease. In particular, chest X-ray scans can be examined to determine whether a patient has COVID-19 [13]. Using chest X-ray pictures, the research described an automated approach for COVID-19 detection. The use of spectral analysis in the construction of an enhanced depth wise convolution network. A simplified case of filtering and down sampling is used to rewrite the pooling and convolution layers.

According to the survey following are the responses gathered from people who are known and unknown about lung diseases affecting.

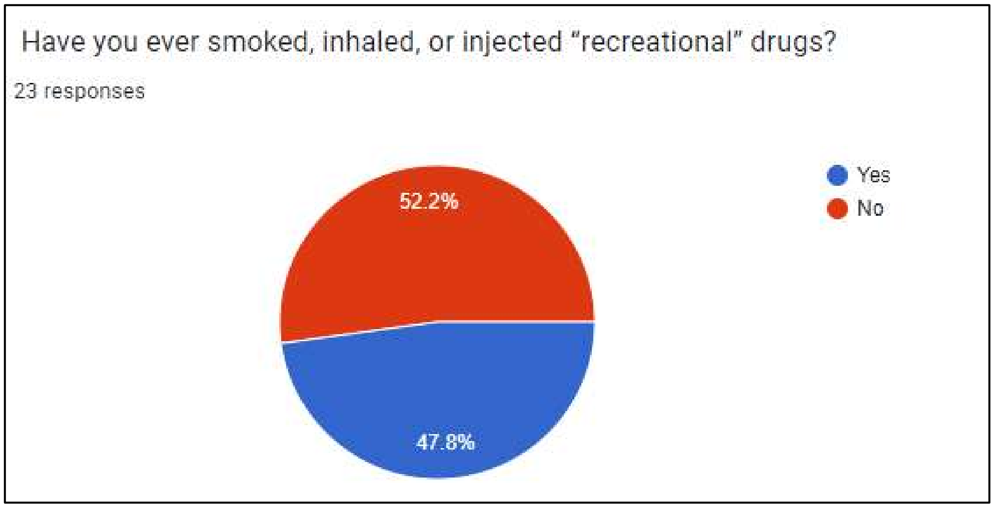

Figure 1:Have you ever smoked?

According to the above information (Figure 1) 52.2% person are not to get smoked or any drugs.

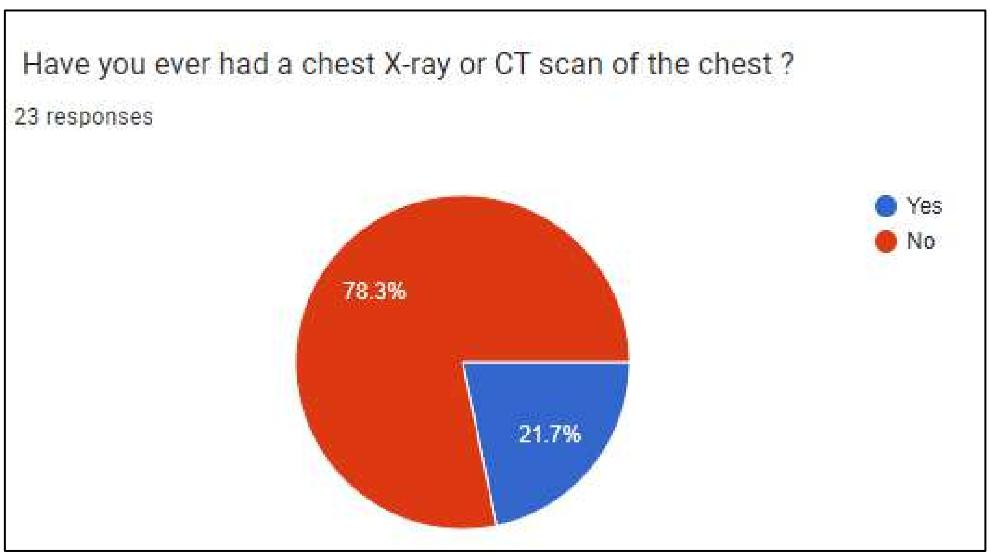

Figure 2:Have you had a X -ray or CT scan?

Considering the figure (Figure 2), patients are not to take any chest X rays in previously. Chest X-ray (CXR) is an important tool for diagnosing for lung diseases and many clinical decisions rely heavily on its radiological findings.

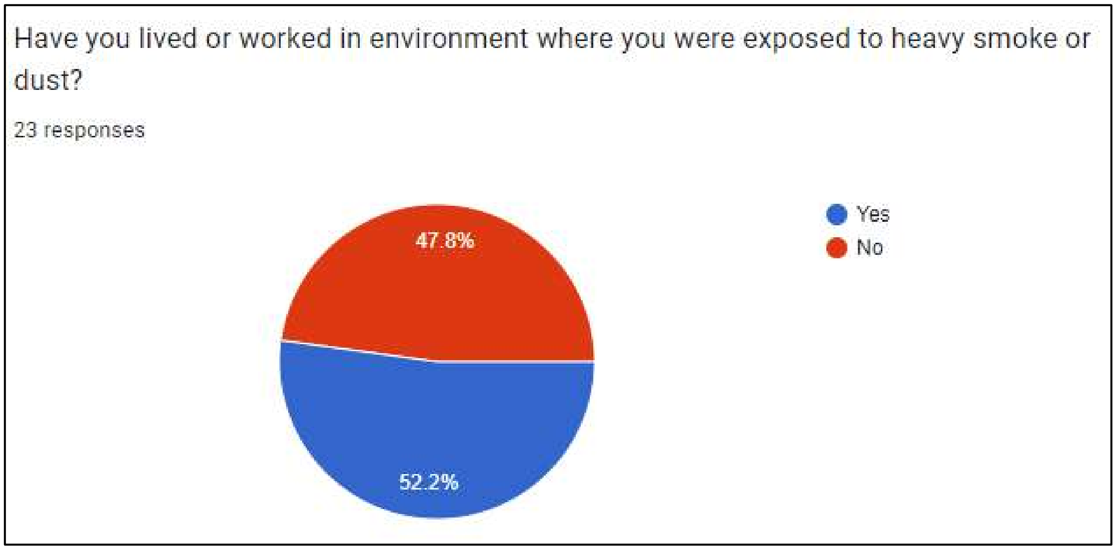

Figure 3:Worked Environment

This graph shows (Figure 3) most of the patients are living to the dusty area.

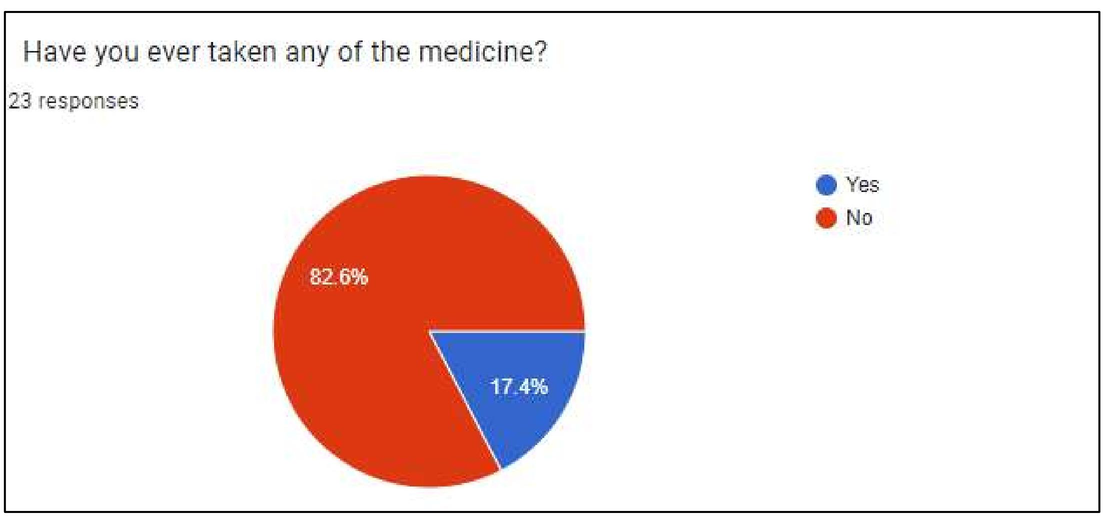

Figure 4:Get any medicine

From in this figure (Figure 4) most of the patients are not to take any medicine or any provide suggestions.

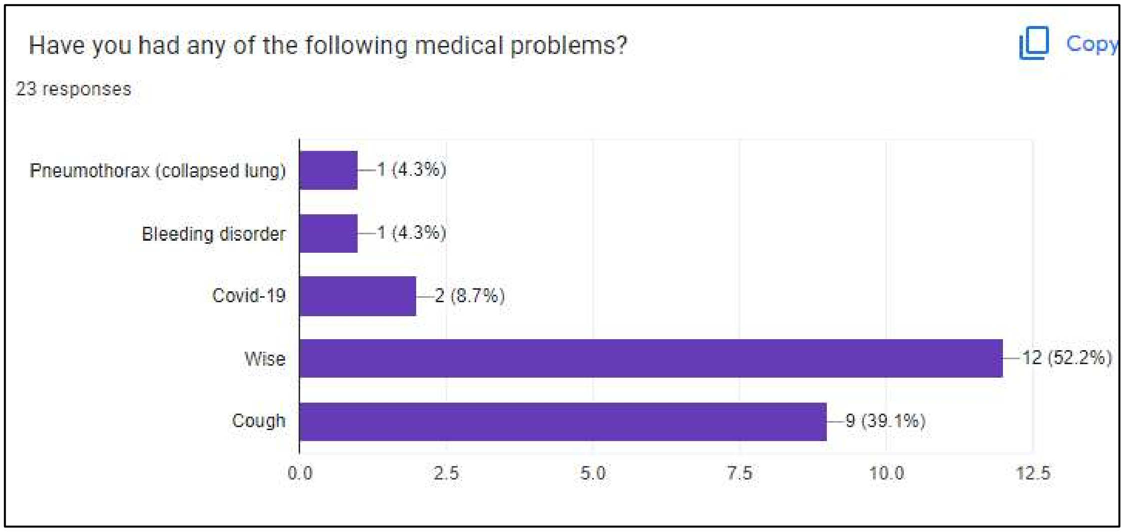

Figure 5:Diseases

This graph shows (Figure 5) many people affected by diseases, mostly affected by 52% (wise). Due to the current situation, some people have COVID-19, so that disease's bad side is lung issues.

3. Research Objective

Following the identification of the requirement for a system like that which is proposed in this document, the main objective of this document is to identify an exact lung disease among other lung diseases and give suggestions to patients.

To achieve the main goal, the following goals must be accomplished.

- Generate suggestions for every patient.

  Most of the systems do not generate suggestions for their patients. But in that system, it provides the suggestions.

4. Methodology

4.1 System Overview

The human body's lung is an important organ. Lungs become infected with illnesses for a variety of environmental, biological, and behavioural reasons. In the modern world, thoracic diseases such atelectasis, consolidation, mass, nodule, etc. are among the life-threatening illnesses. One of the primary methods for determining the patient's sickness is a chest X-ray. In general, a skilled medical expert will manually evaluate a patient's body to diagnose the disease. The tedious task takes up the experts' time. A major issue is the dearth of skilled medical personnel who are capable of quickly and accurately analysing these X-rays.

The topic and background were briefly introduced in the earlier section, which also covered the literature review and the system. This chapter will go into how the proposed system is to be implemented, including the technologies, frameworks, libraries, programming languages, and integrated development environments (IDEs) that are chosen. This chapter will also include explanations of the system's supporting images and modifications.

This chapter will also go over the various testing carried out on the finished system. Functional and non-functional needs have mostly been covered using some testing methods to make it apparent about the features incorporated and the quality of the system. In addition, this testing chapter will discuss testing techniques.

4.2 System Architecture

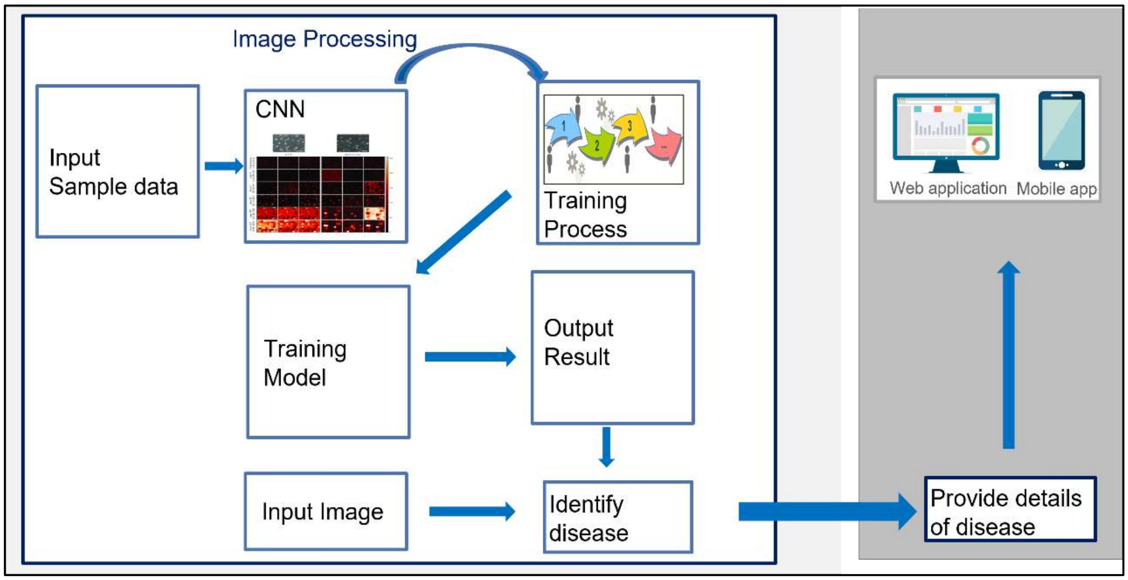

Figure 6:System Architecture

Above figure (Figure 6) demonstrates the overall system architecture of the lung disease analysis. In initial step get the sample data set using the CNN algorithm and train model. Finally, you can add image and check whether which kind of lung diseases affected to the upload person.

4.3 Flow Chart

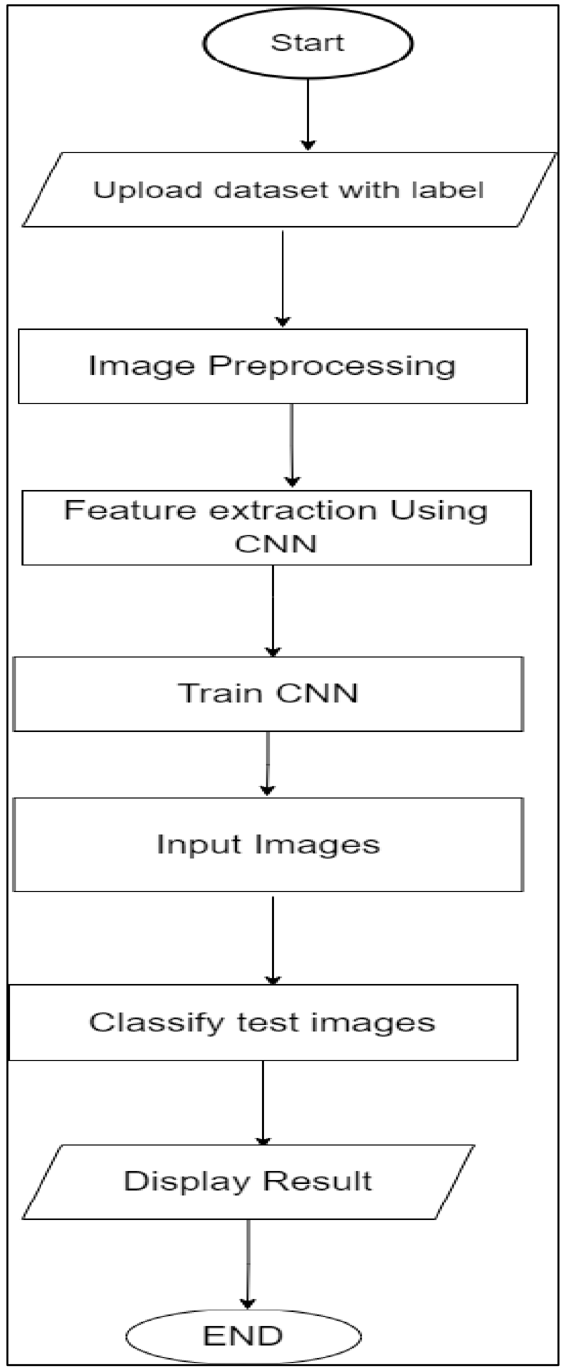

Figure 7:Flow Chart

4.4 Commercilization

Due to their hectic schedules, people find it challenging to frequently visit hospitals to determine whether they are affected with specific disorders. The world has seen the introduction of many kinds of digital applications over time, but none of them have ever been acknowledged as being accurate and comprehensive. We therefore developed a "Medi safe" solution. Not just for the COVID pandemic, it is very helpful in all pandemic situations. This application does two things to save us from having to visit hospitals: it recognizes different types of lung disease and determines whether a person is affected or not. This application will be commercialized in two stages, one of which is unpaid and the other of which is. The premium edition now has extra features as a result. That is seen in the table below.

Limited options for the free edition include

- There was one lung disease found
- The free version of the app can only detect human body temperature

All the features of our product were available in the premium version. Once we have finished the project, we want to post our output on social media. so that we can develop a “Medisafe" page on Instagram and Facebook anyone interested in using this product can do so using their preferred paid or free version.

5. Testing And Implementation

5.1 Testing

The Software Development Life Cycle includes testing as a critical step. In order to make sure that all requirements are met, these processes must be carried out thoroughly. Consequently, the system was put to the test in the following four steps.

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing
- Testing Environment

5.1.1 Unit Testing

You, the developer, are responsible for making sure that each component complies with the test cases. The major benefit of doing unit testing is that we can identify mistakes early on and merge the code without defects to GitLab, allowing us to presume that there won't be any errors in the project.

5.1.2 Integration Testing

Given that our project consists of four main parts, we must combine those modules into a single project. Additionally, our project consists of a number of software modules that were created by different programmers utilizing the python and react languages. Finding flaws in how these software modules interact when combined is the aim of this level of testing.

5.1.3 System Testing

The entire software product's flow, from beginning to end, as well as the desired output based on the inputs, as well as the user experience of our product, must be examined after system testing. Because a small mistake may have a big influence on the entire system, this testing technique was done with extreme caution.

5.1.4 User Acceptance Testing

Before moving to the production environment, user acceptance testing is the last stage of testing. We perform it with our SLIIT friends and then identify several user difficulties.

5.1.5 Testing Environment

This system is tested using the laptops listed below, and the web application loading process is impacted by the laptop's speed. A testing environment is indeed a server that enables you to execute the defined test cases. Setting up a server to conduct tests on is only one aspect of the test environment. In addition, hardware and network settings are involved.

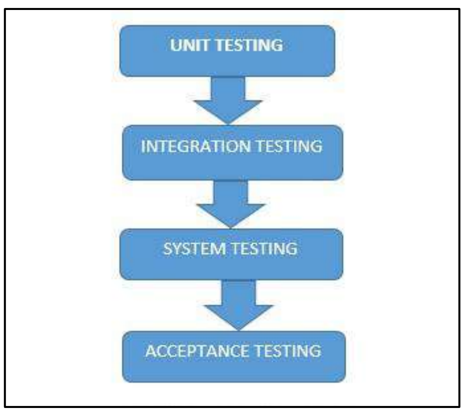

Figure 8:System Testing Approach

5.2 Implementation

5.2.1 Technology selection

When selecting the language, I was thinking of two major aspects. those are front end developments and backend developments. In the backend developments, I selected Python and ReactJS were selected for the frontend developments.

Python is a straightforward, multipurpose, object-oriented, and high-level programming language. Python is also an interpreted programming language. The inventor of the Python programming language is known as Guido Van Rossum. Python’s fast edit-test-debug cycle and lack of a compilation step make for quick development and debugging.

After the select python language for main backend developments. The next difficult task is to find an appropriate IDE for the development process. The IDEs Visual Studio Code is a desktop software program that is small and has a strong source code editor. Microsoft created a free source code editor for Windows, Mac OS, and Linux. It is a software editor with extensive extensions for many other languages, including runtime language extensions for.NET and Unity as well as C++, C+, C, Java, Python, PHP, and Go. It is simple to edit, build, highlight syntax, use snippets, refactor code, and debug. We can modify the application's background theme, make customized keyboard shortcuts, add an extension, and add new features in visual studio code.

ReactJS is a declarative, effective, and adaptable JavaScript library for creating reusable UI components. It is an open-source front end library that just handles the application's view layer. It was initially created and managed by Facebook and then utilized in its products like WhatsApp & Instagram. Building User Interfaces (UI) that increase app speed is ReactJS' primary goal. The application performs better because to the use of virtual DOM (JavaScript object). Compared to the standard DOM, the JavaScript virtual DOM is quicker. ReactJS can be used with different frameworks and on the client- and server-side. It makes use of data and component patterns that make larger programs easier to manage and more readable.

Convolutional neural networks (CNNs) are a special type of artificial neural network architecture that Yann LeCun first proposed in 1988. Since then, CNNs have been widely used to solve a wide range of pattern recognition issues, including computer vision, speech recognition, image classification, image recognition, language translation, medical diagnosis, and others. To obtain meaningful information, a CNN is made up of layers that filter (convolve) the inputs. To automatically change these filters to extract the most important information for the job at hand without feature selection, these convolutional layers have parameters (or kernels) that are learned. To work with images, CNN is preferable. With regard to image categorization issues, conventional neural networks do not fit well. It is crucial to evaluate the classification accuracy since doing so enables one to forecast how accurate future data will be classified. Comparing several classifiers is aided by accuracy. We used the Hold-out approach to evaluate each CNN classifier component of the Multi-CNNs Model.

In a matter of lines of code, Colab enables you to import an image dataset, train an image classifier on it, and then assess the model. Colab notebooks run code on Google's cloud servers, allowing you to take advantage of Google technology, such as GPUs and TPUs, regardless of the processing capability of your computer. Just a browser will do. The machine learning community makes widespread use of Colab in applications like:
Getting started with TensorFlow.

- TensorFlow: How to Get Started
- Neural network development and training
- TPU research and development
- AI research dissemination
- The making of tutorials

Table 2:Technologies and Tools

Language	Libraries	Algorithm	IDE
Python	TensorFlow	CNN	Visual Studio
ReactJS			Google Colab

5.2.2 Implementation Process

This model was trained with Google Colab. Jupiter-based Colab notebooks are cloud-based and have close Google Drive integration. Colab supports a wide range of well-known deep learning libraries, which you can rapidly load into your notebook.

After which you imported the following helpful libraries:

- NumPy
- Pandas
- TensorFlow

In this study, the patient's chest x-ray is used to classify the patient's disease. The paper explains each step in this process: starting with the dataset, pre-processing the data to ensure relevance, developing the model, training the model using the dataset, and ending with the model's prediction of the disease. Following model testing for the best outcome. The document follows each step-in details.

```
[ ] batch_size = 32
    img_height = 180
    img_width = 180
```

```
[ ] train_ds = tf.keras.preprocessing.image_dataset_from_directory(
      data_dir,
      validation_split=0.2,
      subset="training",
      seed=123,
      image_size=(img_height, img_width),
      batch_size=batch_size)

    Found 28 files belonging to 4 classes.
    Using 23 files for training.
```

```
[ ] val_ds = tf.keras.preprocessing.image_dataset_from_directory(
      data_dir,
      validation_split=0.2,
      subset="validation",
      seed=123,
      image_size=(img_height, img_width),
      batch_size=batch_size)

    Found 28 files belonging to 4 classes.
    Using 5 files for validation.
```

Figure 9:Image Training

At least 32 images are trained at least once for each image size, width, and height.

Convolutional neural networks are an improvement on the artificial neural networks' method. There is input, hidden, and output layers in convolutional neural networks as well. A picture with equal-sized rows and columns serves as CNN's input layer. Convolution, pooling, linked, and activation layers are all included in the term "hidden layer," which is used to refer to all layers that are concealed. The final layer, which identifies the type or category of the case research, is the output layer.

```
[ ] num_classes = 4

    model = Sequential([
      layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
      layers.Conv2D(16, 3, padding='same', activation='relu'),
      layers.MaxPooling2D(),
      layers.Conv2D(32, 3, padding='same', activation='relu'),
      layers.MaxPooling2D(),
      layers.Conv2D(64, 3, padding='same', activation='relu'),
      layers.MaxPooling2D(),
      layers.Flatten(),
      layers.Dense(128, activation='relu'),
      layers.Dense(num_classes)
    ])
```

Figure 10:Convolutional layers

The dataset obtained was not exactly as desired for the training of convolutional neural network. Thus, to make the dataset suitable for further processing and training, some pre-processing was applied to them. Processes like tokenization of class label, one hot encoding of class label, resizing of image etc. finally made dataset suitable for the further development. The images were initially of size 0.255(1/255) pixels by next convert to the 0.1pixels. The photos were resized to 180 pixels by 180 pixels to reduce computation power needed for training of model. The choice of image dimension of 180 pixels by 180 pixels is inspired by some of the previous works. Similarly, dataset was divided into training set, validation set and test set for training, validating, and testing of model respectively. Randomly selected 100% data is used for training, 80% for validation, 20%.

A feature map is down sampled (pooled) by using the Max Pooling operation, which determines the maximum value for patches of the feature map. It is frequently applied following a convolutional layer. It adds a tiny bit of translation invariance, which means that changing the image's size slightly has little impact on the values of most pooled outputs.

Get some tangent values and check them randomly. Normally, Adam function finds an optimum point, and that point is an accuracy point. Check the loss function of the system. A loss function analyses how effectively the neural network models the training data by comparing the target and predicted output values. We try to reduce this difference in output between the predicted and the target during training.

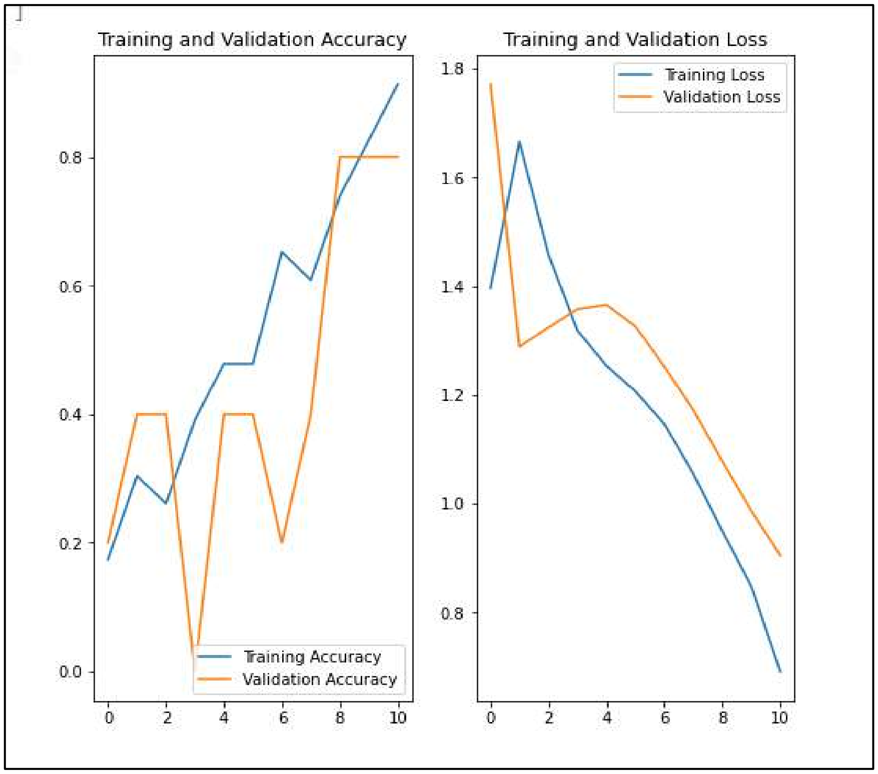

Figure 11:Traning and Validation accuracy and Loss

Beginning with the dataset, preparing the data to ensure relevance, creating the model, utilizing the dataset to train the model, and concluding with the model's disease prediction. Using model testing after for the best result.

6. Result And Discussion

Uploading normal chest x ray disease by the user, the chest x rays will be analysed in which lung disease affected. And providing details of lung disease. Finally, after generated suggestions for the patient.

As a result, we suggest a model in this research based on convolutional neural networks dubbed multi-CNNs. We also suggest a technique in this study, which we refer to as Fusion rules, for combining the outcomes of the model's individual parts. The experimental outcomes (96%in our dataset of x-ray images) demonstrated the viability of the suggested multi-CNNs approach.

The models proposed in this research for the detection of pneumonia, Lung Opacity and COVID -19 on the frontal chest x-ray images using CNN and Transfer learning approaches have significant results. Neural Networks were used to develop models for efficient extraction of features from an x-ray image and predict the presence or absence of pneumonia, covid-19, and lung opacity.

Additionally, data augmentation techniques were used in this study to expand the testing data, which allowed the models to be tested on more photos than in some earlier studies. With the use of technologies like CNN, developments in computer applications have made it possible to detect lung opacity, pneumonia, and COVID-19 using chest X-rays. In this study, techniques that can predict lung diseases with 80% accuracy were created.

7. Conclusion

In this busy lifestyle, it is rare for people to visit hospitals frequently for check-ups. So, this kind of X-ray can be useful to patients to identify which lung disease they have and get confidence in the diagnosis. On the other hand, there is no extra cost and it’s not time wasting like we visit the hospital This research main target was identified what kind of diseases you have been designed with several diseases (Lung Opacity, COVID -19, Pneumonia) and generated suggestions for each patient's disease. Because some people don’t know how to protect their body parts. This system uses technologies like image processing. An improved depth wise convolution network is designed that incorporates spectral analysis. The convolution and max pooling layers are reformulated as a generalized case of filtering and down sampling.

8. References

[1] P. N. Kieu, H. S. Tran, T. H. Le, T. Le and T. T. Nguyen, "Applying Multi-CNNs model for detecting abnormal problem on chest x-ray images," 2018 10th International Conference on Knowledge and Systems Engineering (KSE), 2018, pp. 300-305, doi: 10.1109/KSE.2018.8573404.

[2])A. Sharma, D. Raju and S. Ranjan, "Detection of pneumonia clouds in chest X-ray using image processing approach," 2017 Nirma University International Conference on Engineering (NUiCONE), 2017, pp. 1-4, doi: 10.1109/NUICONE.2017.8325607

[3] S. Chaudhary, S. Sadbhawna, V. Jakhetiya, B. N. Subudhi, U. Baid and S. C. Guntuku, "Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images With Deep Learning," ICASSP 2021 - 2021 IEEE International Conference on Acoustics,Speech and Signal Processing (ICASSP),2021, pp. 8583-8587, doi: 10.1109/ICASSP39728.2021.9414007.

[4] M. S. Pattichis, H. Muralldharan, C. S. Pattichis and P. Soliz, "New image processing models for opacity image analysis in chest radiographs," Proceedings Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, 2002, pp. 260-264, doi: 10.1109/IAI.2002.999929.

[5] N. Darapaneni et al., "COVID 19 Severity of Pneumonia Analysis Using Chest X Rays," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020, pp. 381-386, doi: 10.1109/ICIIS51140.2020.9342702.

[6] A. W. Setiawan, "Effect of Chest X-Ray Contrast Image Enhancement on Pneumonia Detection using Convolutional Neural Networks," 2021 IEEE

International Biomedical Instrumentation and Technology Conference (IBITeC), 2021, pp. 142-147, doi: 10.1109/IBITeC53045.2021.9649319.

[7] M. M. Hasan, M. Md. Jahangir Kabir, M. R. Haque and M. Ahmed, "A Combined Approach Using Image Processing and Deep Learning to Detect Pneumonia from Chest X-Ray Image," 2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), 2019, pp. 89-92, doi: 10.1109/ICECTE48615.2019.9303543.

[8] R. Tomar and A. Sharma, "Analysis & Detection of COVID-19 on Chest X- Ray Images based on Support Vector Machines," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022, pp. 1583-1590, doi: 10.1109/ICSCDS53736.2022.9760748.

[9] H. Sharma, J. S. Jain, P. Bansal and S. Gupta, "Feature Extraction and Classification of Chest X-Ray Images Using CNN to Detect Pneumonia," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 227-231, doi: 10.1109/Confluence47617.2020.9057809.

[10] T. R. Pant, R. K. Aryal, T. Panthi, M. Maharjan and B. Joshi, "Disease Classification of Chest X-Ray using CNN," 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), 2021, pp. 467-471, doi: 10.1109/ICCCA52192.2021.9666246.

[11] N. Nafi’iyah and E. Setyati, "Lung X-Ray Image Enhancement to Identify Pneumonia with CNN," 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), 2021, pp. 421-426, doi: 10.1109/EIConCIT50028.2021.9431856.

[12] J. An, Q. Cai, Z. Qu and Z. Gao, "COVID-19 Screening in Chest X-Ray Images Using Lung Region Priors," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 11, pp. 4119-4127, Nov. 2021, doi: 10.1109/JBHI.2021.3104629.

[13] K. K. Singh and A. Singh, "Diagnosis of COVID-19 from chest X-ray images using wavelets-based depthwise convolution network," in Big Data Mining and Analytics, vol. 4, no. 2, pp. 84-93, June 2021, doi: 10.26599/BDMA.2020.9020012.

[14] G. Labhane, R. Pansare, S. Maheshwari, R. Tiwari and A. Shukla, "Detection of Pediatric Pneumonia from Chest X-Ray Images using CNN and Transfer Learning," 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), 2020, pp. 85-92, doi: 10.1109/ICETCE48199.2020.9091755.

9. Appendices

```
[ ] image_count = len(list(data_dir.glob('*/*.jpg')))
    print(image_count)

    2
```

Figure 12:Check JPEG, PNG, or JPG

```
[ ] import matplotlib.pyplot as plt

    plt.figure(figsize=(10, 10))
    for images, labels in train_ds.take(1):
      for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
```

Figure 13:Plot

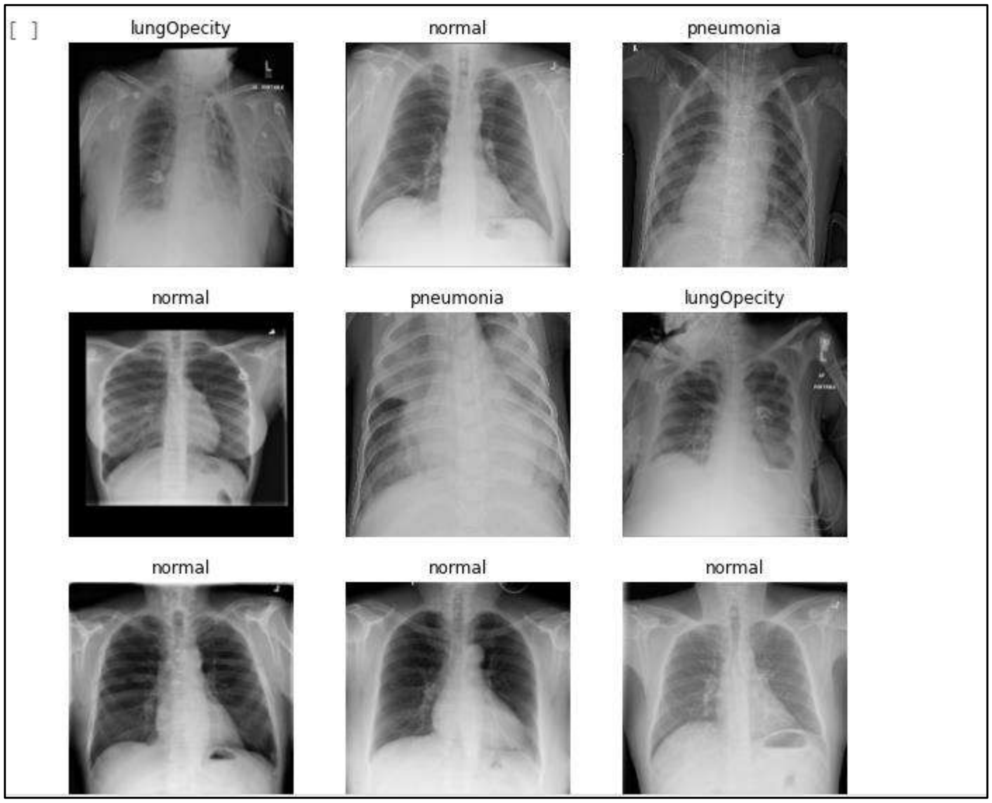

Figure 14:Check Plot Images

```
epochs=11
history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs,
  callbacks=[cp_callback]
)
```

```
Epoch 1/11
1/1 [==============================] - ETA: 0s - loss: 1.3959 - accuracy: 0.1739
Epoch 1: saving model to training_1/cp.ckpt
1/1 [==============================] - 2s 2s/step - loss: 1.3959 - accuracy: 0.1739 - val_loss: 1.7708 - val_accuracy: 0.2000
Epoch 2/11
1/1 [==============================] - ETA: 0s - loss: 1.6651 - accuracy: 0.3043
Epoch 2: saving model to training_1/cp.ckpt
1/1 [==============================] - 1s 888ms/step - loss: 1.6651 - accuracy: 0.3043 - val_loss: 1.2884 - val_accuracy: 0.4000
Epoch 3/11
1/1 [==============================] - ETA: 0s - loss: 1.4579 - accuracy: 0.2609
Epoch 3: saving model to training_1/cp.ckpt
1/1 [==============================] - 1s 872ms/step - loss: 1.4579 - accuracy: 0.2609 - val_loss: 1.3244 - val_accuracy: 0.4000
Epoch 4/11
1/1 [==============================] - ETA: 0s - loss: 1.3184 - accuracy: 0.3913
Epoch 4: saving model to training_1/cp.ckpt
1/1 [==============================] - 1s 868ms/step - loss: 1.3184 - accuracy: 0.3913 - val_loss: 1.3577 - val_accuracy: 0.0000e+00
Epoch 5/11
1/1 [==============================] - ETA: 0s - loss: 1.2538 - accuracy: 0.4783
Epoch 5: saving model to training_1/cp.ckpt
1/1 [==============================] - 1s 854ms/step - loss: 1.2538 - accuracy: 0.4783 - val_loss: 1.3653 - val_accuracy: 0.4000
Epoch 6/11
1/1 [==============================] - ETA: 0s - loss: 1.2068 - accuracy: 0.4783
Epoch 6: saving model to training_1/cp.ckpt
```

Figure 15:Traning

```
[ ] acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']

    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs_range = range(epochs)

    plt.figure(figsize=(8, 8))
    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.title('Training and Validation Accuracy')

    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Training Loss')
    plt.plot(epochs_range, val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title('Training and Validation Loss')
    plt.show()
```

Figure 16:Accuracy Loss check

```
[ ] normalization_layer = layers.experimental.preprocessing.Rescaling(1./255)
```

```
[ ] normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
    image_batch, labels_batch = next(iter(normalized_ds))
    first_image = image_batch[0]
    # Notice the pixels values are now in `[0,1]`.
    print(np.min(first_image), np.max(first_image))

    0.0 0.93016136
```

Figure 17:Normalized Data

```
[ ] model.load_weights(checkpoint_path)

    # Re-evaluate the model
    loss, acc = model.evaluate(val_ds, verbose=2)
    print("Restored model, accuracy: {:5.2f}%".format(100 * acc))

    1/1 - 0s - loss: 0.9053 - accuracy: 0.8000 - 67ms/epoch - 67ms/step
    Restored model, accuracy: 80.00%
```

Figure 18:Get model weight and save

```python
#load model
class TFModel:
    def __init__(self, model_dir) -> None:
        # make sure our exported SavedModel folder exists
        self.model_dir = model_dir
        with open(os.path.join(model_dir, "signature.json"), "r") as f:
            self.signature = json.load(f)
        self.model_file = "../" + self.signature.get("filename")
        if not os.path.isfile(self.model_file):
            raise FileNotFoundError(f"Model file does not exist")
        self.inputs = self.signature.get("inputs")
        self.outputs = self.signature.get("outputs")
        self.lock = Lock()
```

Figure 19:Load Model

```python
#create dictonary
def predict(self, image: Image.Image) -> dict:
    # pre-processing the image before passing to model
    image = self.process_image(image, self.inputs.get("Image").get("shape"))

    with self.lock:
        # create the feed dictionary that is the input to the model
        feed_dict = {}
        # first, add our image to the dictionary (comes from our signature.json file)
        feed_dict[list(self.inputs.keys())[0]] = tf.convert_to_tensor(image)
        # run the model!
        outputs = self.predict_fn(**feed_dict)
        # return the processed output
        return self.process_output(outputs)
```

Figure 20:Image convert to Tense

```python
def process_image(self, image, input_shape) -> np.ndarray:
    """
    Given a PIL Image, center square crop and resize to fit the expected model input, and convert from [0,255] to [0,1] values.
    """
    width, height = image.size
    # ensure image type is compatible with model and convert if not
    if image.mode != "RGB":
        image = image.convert("RGB")
```

Figure 21:Convert RGB

```python
# center crop image (you can substitute any other method to make a square image, such as just resizing or padding edges with 0)
if width != height:
    square_size = min(width, height)
    left = (width - square_size) / 2
    top = (height - square_size) / 2
    right = (width + square_size) / 2
    bottom = (height + square_size) / 2
    # Crop the center of the image
    image = image.crop((left, top, right, bottom))
# now the image is square, resize it to be the right shape for the model input
input_width, input_height = input_shape[1:3]
if image.width != input_width or image.height != input_height:
    image = image.resize((input_width, input_height))
```

Figure 22:Image Resize

```python
for item in outputs['predictions']:
    list_Of_cf.append(item['confidence'])

for item in outputs['predictions']:
    if item['confidence'] == max(list_Of_cf):
        print(' prediction: ',item['label'],' | confidence: ', max(list_Of_cf) * 100)
        prediction_inst.append(item['label'])
```

Figure 23:Prediction

```python
    # temp_val = prediction_inst[0]
    # print(f"Predicted: {temp_val}")
else:
    print(f"Couldn't find image file {args.image}")
```

Figure 24:After confidence is Equal print

Figure 25:Covid Result

Figure 26:Lung Result

Figure 27:Pnuemonia Result