## **An Effective Pneumothorax Detection**

## **Using Convolutional Neural Network Model**

## A PROJECT REPORT

*Submitted by*

# GINNELA JAYADEEP [Reg No:RA2011003010397]

# Y. BHARATH SIMHA[RegNo:RA2011003010407]

*Under the Guidance of*

# Dr. C. VIJAYAKUMARAN

Associate Professor, Department of Computing Technologies

*in partial fulfillment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**

# in

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTING TECHNOLOGIES**

# COLLEGE OF ENGINEERING AND TECHNOLOGY

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR– 603 203

**OCTOBER 2023**



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

# KATTANKULATHUR–603 203

## **BONAFIDE CERTIFICATE**

Certified that 18CSP109L / I8CSP111L project report titled “**An Effective Pneumothorax Detection Using CNN Model**” is the bonafide work of **GINNELA JAYADEEP [RA2011003010397]**  and **Y. BHARATH SIMHA [RA2011003010407]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

### Dr. C. VIJAYAKUMARAN

### SUPERVISOR & PANEL HEAD

Associate Professor

Department of Computing Technologies

### Dr. M. PUSHPALATHA

### HEAD OF THE DEPARTMENT

Department of Computing Technologies

 Department of Computing Technologies

# SRM Institute of Science and Technology Own Work Declaration Form

**Degree/Course :** B. Tech in Computer Science and Engineering

S**tudent Names :** GINNELA JAYADEEP, Y BHARATH SIMHA

**Registration Number:** RA2011003010397, RA2011003010407

**Title of Work :**

I/We here by certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

* + - Clearly references / listed all sources as appropriate
    - Referenced and put in inverted commas all quoted text (from books, web,etc.)
    - Given the sources of all pictures, data etc that are not my own.
    - Not made any use of the report(s) or essay(s) of any other student(s)either past or present
    - Acknowledged in appropriate places any help that I have received from others (e.g fellow students, technicians, statisticians, external sources)
    - Compiled with any other plagiarism criteria specified in the Course hand book / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.  **Student 1 Signature: Ginnela Jayadeep**  **Student 2 Signature: Y. Bharath Simha**  **Date: 13/10/23** |
|  |

# ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T. V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha,** Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Panel Members, **Dr. C. Vijayakumaran,** Panel Head**, Dr. B. Amutha,** Assistant Professor, **Dr. Ramkumar Jayaraman,** Assistant Professor and **Dr. R. Jeya,** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. S. Jagadeesan**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. C. Vijayakumaran**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his mentorship. He provided us with the freedom and support to explore the research topics of our interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

# GINNELA JAYADEEP [Reg. No: RA2011003010397]

# 

# 

# Y. BHARATH SIMHA [Reg. No: RA2011003010407]

# ABSTRACT

# The early and accurate diagnosis of medical conditions, especially those affecting the respiratory system, is paramount in providing effective healthcare. This project focuses on the development of a Convolutional Neural Network (CNN) model for the classification of chest X-ray images into "Infected" and "Uninfected" categories. The primary objective is to aid medical professionals in quickly identifying cases of pneumothorax, a critical condition involving the presence of air in the pleural cavity. The proposed CNN model aims to enhance the accuracy of pneumothorax diagnosis and reduce the time required for manual analysis. The project utilizes a dataset of chest X-ray images, categorized into training, validation, and testing sets. Data preprocessing techniques, including data augmentation and rescaling, are applied to enhance the quality of the training data. The CNN architecture consists of multiple convolutional and pooling layers that enable the automatic extraction of essential features from the input images. The classification is performed through fully connected layers with a final soft max activation. Several challenges and limitations are addressed in this project. These include the need for a large and diverse dataset to improve model performance and potential ethical considerations in medical image analysis. To address these challenges, future enhancements may involve data collection from multiple sources and the integration of privacy-preserving techniques to ensure patient confidentiality. The project presents an innovative approach to pneumothorax diagnosis by leveraging state-of-the-art deep learning techniques. It also explores the possibilities of applying advanced CNN architectures like Dense Net, Res Net, or Efficient Net for better classification performance. Further research may focus on interpretability and explainability of the CNN model to ensure its integration into clinical practice. In conclusion, the CNN model presented in this project holds significant promise for improving the accuracy and efficiency of pneumothorax diagnosis based on chest X-ray images. With additional data and advanced architectures, it can be further enhanced to address a broader range of medical conditions. The development of such models contributes to the evolution of modern healthcare, ensuring timely and accurate diagnoses, and ultimately improving patient outcomes.

# 

# TABLE OF CONTENTS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ABSTRACT 6**  **LIST OF FIGURES 10**  **LIST OF SYMBOLS AND ABBREVIATIONS 11** | | | | | | |
|  |
|  |
| **1**. | **INTRODUCTION** | | | | **12** | |
|  | 1.1 | | Pneumothorax | | 12 | |
|  | 1.2 | | Convolutional Neural Network | | 14 | |
|  | 1.3 | | Scope | | 16 | |
|  | 1.4  1.5 | | Deep Learning  CNN-Based Pneumothorax Detection Process | | 18  20 | |
| **2** | **LITERATURE SURVEY** | | | | **24** | |
|  | 2.1  2.2  2.3  2.4  2.5  2.6 | Objective  Computer Aided Pneumothorax Computer Aided Pneumothorax  Fusion of Frontal and Lateral Chest  Multicenter External Validation Study  Artificial neural networks  CheXLocNet | | | 24 25 27 27 28 | |
| **3** | **ARCHITECTURE AND ANALYSIS OF PNEUMOTHORAX**  **DETECTION USING CNN MODEL** | | | | **30** | |
|  | 3.1 | Architecture Diagram | | | 30 | |
|  |  | 3.1.1 Image Dataset Directory Structure 3.1.2 Image Preprocessing3.1.3 Data Augmentation3.1.4 Convolutional Neural Network Architecture **3.1.5 ReLU**3.1.6 Compile Model | | | 32  3233  3333  33 | |
|  |  |  | |
| **4** | **DESIGN AND IMPLEMENTATION OF PNEUMOTHORAX**  **DETECTION USING CNN MODEL** | | | | **34** | | |
|  | 4.1 | Dataset | | | 34 | | |
|  | 4.2 | Image Preprocessing | | | 35 | | |
|  | 4.3 | **Image Dataset Directory Structure** | | | 36 | | |
|  | 4.4 | Feature Learning layers | | | 36 | | |
|  |  | 4.4.1 Convolution + Activation(RELU) layers | | | 37 | | |
|  |  | 4.4.2 Pooling layers | | | 38 | | |
|  | 4.5 | Classification layer  Flatten layer   * + 1. Fully connected(Dense) layer     2. Fully connected(Dense) layer with Soft max | | |  | 39  39  40  40 | | |
|  |  |  | | |  |  | | |
| **5`** | **RESULTS AND DISCUSSION**  5.1Performance Analysis using Various Metrics | | | | **42**  43 | | |
|  | 5.2 | Comparison Between Existing Models | | | 44 | | |
| **6** | **CONCLUSION AND FUTURE SCOPE** | | | | **46** | | |
|  | 6.1 Conclusion  6.2 Future Scope | | | | 46  47 | | |
|  | **REFERENCES**  **APPENDIX 1**  **APPENDIX 2** | | | | **48**  **51**  **54** | | |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| 1.1 | **Image Dataset Directory Structure**…………………………………………… | 28 |
| 1.2 | Image Preprocessing………………….................................................... | 30 |
| 1.3 | Data Augmentation……….……………………………………………. | 30 |
| 1.7 | Convolution Neural Network…………………………………………... | 31 |
| 1.8 | Compile Model……................................................................................. | 32 |
| 3.1 | Accuracy Curve………………...……………………………………….. | 43 |
|  |  |  |
|  |  |  |

**LIST OF SYMBOLS AND ABBREVIATIONS**

**US** United States of America

**ReLU** Rectified Linear Unit

**CNN** Convolutional Neural Network

**AI** Artificial Intelligence

**ML** Machine Learning

**MRI** Magnetic Resonance Imaging

**CT** Computed Topography

**DL** Deep Learning

**PACS** Picture Archiving and Communication Systems

**CHAPTER 1**

**INTRODUCTION**

**1.1 Pneumothorax**

Detecting pneumothorax, a condition where air accumulates in the pleural space around the lungs, is crucial for timely medical intervention. Several methods and techniques can be employed for its detection. Here's a general overview of how pneumothorax can be detected:

Clinical Assessment:

Symptoms: Evaluate the patient's symptoms, which may include sudden chest pain, shortness of breath, and decreased breath sounds on the affected side.

Medical History: Review the patient's medical history, especially any prior lung or chest trauma, surgery, or pre-existing lung conditions that may increase the risk of pneumothorax.

Physical Examination:

Inspection: Look for signs such as asymmetrical chest movement and visible chest wall abnormalities.

Palpation: Assess chest tenderness, subcutaneous emphysema (crepitus), and chest expansion.

Auscultation: Listen to breath sounds using a stethoscope to check for decreased or absent breath sounds on the affected side.

Imaging Studies:

Chest X-ray: This is a common initial diagnostic tool. It can reveal signs of

pneumothorax, such as a visible pleural line, lung collapse, and mediastinal shift.

Computed Tomography (CT) Scan: CT scans provide detailed cross-sectional images and are more sensitive in detecting small or subtle pneumothorax.

Ultrasound: Lung ultrasound can quickly detect pneumothorax at the bedside, often showing a "lung point" where the visceral and parietal pleura meet.

Arterial Blood Gas (ABG) Analysis:

ABG tests can help assess the impact of pneumothorax on oxygen and carbon dioxide levels in the blood, which may guide treatment decisions.

Point-of-Care Tests:

Some point-of-care tests, such as the use of an ultrasound device, may help in rapid bedside detection, especially in emergency settings.

Electrocardiogram (ECG):

If there's a tension pneumothorax, it can affect cardiac function by causing mediastinal shift. An ECG can be used to monitor changes in heart rhythm.

Differential Diagnosis:

Pneumothorax should be distinguished from other conditions that may present similarly, such as pulmonary embolism, pleuritis, or cardiac issues.

It's important to note that the choice of diagnostic method depends on the clinical situation, the patient's history, and the availability of resources. The combination of clinical assessment, imaging, and laboratory tests is often used to make an accurate

diagnosis and determine the severity of pneumothorax. Treatment options can range from observation for small, stable pneumothorax to more invasive interventions like chest tube insertion for larger, symptomatic cases. Early and accurate detection is crucial in managing this potentially life-threatening condition**.**

**1.2. Convolutional Neural Network**

Convolutional Neural Networks (CNN) and other deep learning models are useful in the medical field for detecting pneumothorax for the following reasons:

Early Diagnosis: CNN models, which analyze medical images like chest X-rays or CT scans, can help in the early and precise diagnosis of pneumothorax. Timely treatment can greatly enhance patient outcomes, therefore early identification is essential.

Automation: CNN models have the ability to automate the detection process, obviating the requirement for manual interpretation by radiologists or other healthcare experts. Faster diagnosis and a lower chance of human error can result from this.

Efficiency: CNNs are useful in busy medical environments where efficiency and speed are crucial since they can process a huge volume of medical pictures quickly.

Consistency: CNNs can deliver reliable, repeatable results. A well-trained CNN's performance is unaffected by things like fatigue or amount of experience, ensuring accurate and dependable detection of pneumothorax situations.

Support for Radiologists: CNN models can be an extremely useful tool for radiologists, assisting them in the triage of cases by pointing up potential areas of concern. Radiologists

may be able to focus on more complicated situations as a result of this saving time.

Allocating Resources: Automated pneumothorax detection can assist healthcare organizations in more efficient resource allocation by identifying cases that demand prompt attention and intervention.

Quality Control: CNN models can help with quality control by double-checking radiologists' interpretations, which lowers the likelihood of missed diagnosis.

Educational Tool: CNN models can be used to teach medical practitioners how to identify pneumothorax on medical images and comprehend the corresponding patterns and features.

Research and Data Analysis: CNN models can aid with extensive medical research by examining a big dataset of medical images, which enables researchers to spot patterns, risk factors, and potential treatments for pneumothorax.

CNN-based detection systems can help in telemedicine and remote healthcare settings to diagnose pneumothorax without the requirement for on-site radiologists.

While CNNs can be effective tools for medical image analysis, it's vital to remember that they shouldn't take the place of a healthcare professional's knowledge. Medical professionals with the necessary training should always make the final diagnoses and treatment decisions.

**1.3. Scope**

Medical imaging applications such as the diagnosis of pneumothorax have showed tremendous potential for Convolutional Neural Networks (CNNs). There are various benefits to utilizing CNN models for pneumothorax identification, which is a rather broad

application.

High Accuracy: CNNs are quite effective at spotting pneumothorax on medical images like chest X-rays or CT scans. To discover intricate patterns and attributes related to pneumothorax, they can be trained on big datasets.

Early Detection: CNNs can spot pneumothorax even when it may not be immediately obvious to the naked eye. This may result in quicker diagnosis and therapy.

CNNs work consistently and are immune to fluctuations in human judgement, which may be affected by variables like weariness.

Scalability: CNN models can be scaled to efficiently process high volumes of medical images, making them suited for application in busy healthcare environments.

Integration: Picture Archiving and Communication Systems (PACS) and CNN-based detection systems can be combined for a streamlined workflow in medical facilities like clinics and hospitals.

Healthcare professionals can remotely assess and diagnose pneumothorax situations using CNN models in telemedicine applications, particularly in underserved or remote areas.

Research and Education: CNNs can be useful tools in the analysis of vast datasets for pneumothorax-related patterns and trends. In order to train medical practitioners, they can also be employed in educational settings.

Customization: CNN models can be adjusted and tailored to match the unique requirements of medical facilities, adjusting to changes in patient populations and imaging technology.

Enhanced Patient Care: By enabling fast treatment and decreasing problems related to pneumothorax, increased accuracy and early detection can result in better patient outcomes.

It's important to note a few difficulties and factors, though:

Data Quality: The calibre and variety of the training data are key factors in how well CNNs operate. There might not be many annotated datasets for pneumothorax.

Interpretability: Because CNNs are frequently referred to as "black-box" models, it might be difficult to communicate their choices to healthcare practitioners. Ongoing research is being done on model interpretability.

**1.4 Deep Learning**

Preprocessing the photos include scaling them to a constant resolution, normalising the pixel values, and extending the dataset (for example, by adding random rotations, flips, or brightness modifications) in order to enhance model generalization.

Building models:

To detect pneumothoraxes, create a CNN architecture. Convolutional layers, pooling layers, and fully linked layers are the main components of a conventional CNN.

As a starting point, think about fine-tuning a pre-trained model, like one from a Res Net, Inception, or VGG architecture, for your particular application. Performance of models can be greatly improved through transfer learning.

**Training:**

Your dataset should be divided into training, validation, and test sets. In order to maximize hyperparameters and avoid overfitting, the model is trained on the training data and validated on the validation set. For binary classification (normal vs. pneumothorax), specify an appropriate loss function, such as binary cross-entropy. Stochastic gradient descent (SGD) or Adam are examples of backpropagation and optimization techniques that can be used to train a model.

Analysing the model

Measure the model's performance on the test set using parameters like accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC).

**Post-processing:**

Using the model's output probabilities as a basis, choose a categorization threshold. For instance, based on the clinical requirements, you might select a threshold that maximize sensitivity (recall) or specificity.

**Deployment:**

When the model functions satisfactorily, a clinical environment can use it. The picture archiving and communication system (PACS) of a hospital or the workflow of a radiologist may both be involved in this.

Regular Development:

To increase the model's accuracy and robustness, keep updating and fine-tuning it as new

data become available or as it experiences real-world situations.

When creating and implementing such models in a clinical setting, it is essential to work with medical professionals and to abide by pertinent regulatory and ethical requirements. Additionally, models must be rigorously validated, tested, and monitored for safety and effectiveness in real-world circumstances to verify their accuracy and dependability in pneumothorax diagnosis.

**1.5 CNN-Based Pneumothorax Detection Process**

A potential method in medical imaging, particularly for interpreting chest X-rays or other radiological images, is the diagnosis of pneumothorax using convolutional neural networks (CNN). CNNs are a class of deep learning models that have proven to be quite effective at image identification tasks. How CNNs can be used to identify pneumothorax is as follows:

Information Gathering and Preparation:

Obtain a sizable chest X-ray image collection that includes both healthy and pneumothorax-affected instances. For training and validation, properly labelled data are crucial.

**Preprocessing of Data:-**

The photographs should be preprocessed to uniformize their size, format, and quality. Resizing, normalisation, and noise reduction are frequent preprocessing techniques.

**Building models:**

Create an image classification-friendly CNN model architecture. Multiple convolutional layers should make up the architecture, which should be followed by pooling layers to extract features from the images. You can modify pre-trained CNN models like VGG, ResNet, or Inception for your particular task.

**Training**:

Your dataset should be divided into training, validation, and test sets. The CNN model should be trained using the training data. The model develops the ability to recognise X-ray image characteristics and patterns that distinguish between normal and pneumothorax situations during training. Additionally, it refines its internal properties.

**Loss Purpose**:

The difference between the predicted and real labels (normal or pneumothorax) in the training data should be measured using a suitable loss function, frequently binary cross-entropy.

**Optimization**:

The weights of the model should be updated, and the loss function should be minimized, during training by using optimization techniques like stochastic gradient descent (SGD) or Adam.

**Validation**:

Keep an eye on how well the model performs on the validation set to avoid overfitting and make sure it generalize well to new data.

**Testing**:

Once the model has undergone training and validation, analyze its performance on the test dataset to determine how well it can identify pneumothorax.

**Post-processing**:

Assign a categorization threshold. You may decide, for example, that the existence of a pneumothorax is indicated if the model's output probability exceeds a specific threshold.

**Deployment:**

A clinical setting can use the CNN model to automatically analyze chest X-ray pictures for pneumothorax identification once it has been trained and validated. It can be used by healthcare practitioners as a decision assistance tool or integrated into a radiological system.

Monitoring and Improvement Continuum:

As new data become available, keep the model current and improve it. Retrain the model on a regular basis to enhance its accuracy and ability to adapt to new situations.

CNNs can be used to detect pneumothoraxes, which might potentially lead to a quicker diagnosis by automating the screening process and easing the burden on radiologists. Although it's crucial to make sure the model is adequately verified and integrated into the clinical workflow, it's also important to take regulatory and ethical issues into account when using AI in healthcare.

**CHAPTER 2**

**LITERATURE REVIEW**

In this section, we go over the earlier studies that were carried out to try and identify aggressive driving behavior. This section contains information on machine learning algorithm developments, resilient neural network architectures, tool combinations to improve the performance of these algorithms and networks, and an overview of the creation and formulation of the neural network architectures utilized in our research.

**2.1 Objective**

1. Pneumothorax Detection: Create a CNN-based model that can recognize if pneumothorax is present or absent in medical images. To reduce false positives and false negatives, this model ought to have a high sensitivity and specificity.
2. Improved Accuracy: Increase the detection accuracy of pneumothorax in comparison to current approaches, such as conventional CNN models and conventional image processing techniques.
3. Data handling: Gather or curate a sufficient number of high-quality medical photos with balanced representation of positive and negative instances to address the issue of sparse and unbalanced datasets for pneumothorax.
4. Annotation Quality: By using strict quality control procedures, professional validation, and annotation revisions, you can guarantee accurate and consistent annotations.
5. Ethical Compliance: To protect patient privacy and data security, handle patient data with the utmost care, adhering to ethical standards and data protection legislation.

**2.2 Computer Aided Pneumothorax**

A thoracic illness called pneumothorax can cause cardiac arrest, breathing failure, or, in severe situations, even death. The main diagnostic imaging method for pneumothorax diagnosis is chest X-ray (CXR) imaging. Pneumothorax can be seen in chest radiography images using a computerized diagnosis system, which offers significant advantages for illness diagnosis. The current study proposes a deep learning neural network model to identify pneumothorace areas in chest X-ray images. The model uses transfer learning with ResNet101 as the main feature pyramid network (FPN) and a Mask Regional Convolutional Neural Network (Mask RCNN) framework. A pneumothorax dataset created by the Society for Imaging Informatics in Medicine (SIIM-ACR) in collaboration with the American College of Radiology was used to train the suggested model.

In this work, the traditional model based on ResNet50 as an FPN and the suggested MRCNN model based on ResNet101 as an FPN are compared for functionality. When compared to the traditional model based on ResNet50 as an FPN, the suggested model demonstrated lower class loss, bounding box loss, and mask loss. For 10 and 12 epochs, respectively, both models were simulated at learning rates of 0.0004 and 0.0006.

**2.3 Fusion of Frontal and Lateral Chest**

Pneumothorax is a potentially fatal condition that needs to be diagnosed and treated very away. When diagnosing a pneumothorax, a chest X-ray examination is the primary option. However, when the lesion region consists of only a little amount of air, it is challenging to diagnose pneumothorax solely based on frontal chest X-ray imaging. Consequently, we suggest a neural network for pneumothorax diagnosis based on feature fusion, which fuses lateral and frontal X-ray data. There are two inputs and three outputs in this network. The lateral chest X-ray image and the frontal chest X-ray image are the two inputs. The three outcomes include the frontal chest X-ray image classification results, the lateral chest X-ray image classification results, and the classification results integrating the attributes of the combination of the lateral and frontal chest X-ray images. Our technique takes into account the pneumothorax recognition model's vanishing gradient issue and adds the residual block to address it. We also apply channel attention strategies to enhance the model's performance because of its vast number of channels. Our comparative studies demonstrate that the single task model can be outperformed in terms of accuracy by using a neural network fusion of frontal and lateral chest imaging data. Our pneumothorax model can achieve good recognition accuracy by using simply image-level annotation.

**2.4 Multicenter External Validation Study**

In order to detect pneumothorax, we trained a single convolutional neural network model in this study using a sizable number of public chest radiographs. We then tested the model using six external datasets from various institutions to assess its generalizability. The training data come exclusively from US universities, while five of the six external test sets are geographically independent and come from institutions in an Asian multiethnic nation. To the best of our knowledge, the largest true external validation study of artificial intelligence algorithms in radiology literature was conducted through external testing of 2931 chest radiographs from various institutions. With AUCs ranging from 0.91 to 0.98 across a variety of external datasets, we discovered that our deep learning network could generalize effectively even without network optimization and lack of prior knowledge about the external data.

**2.5 Artificial neural networks**

This study aimed to assess the diagnostic performance in identifying pneumothorax site in chest X-rays using fully-connected tiny artificial neural networks (ANNs) with a straightforward training procedure, the Kim-Monte Carlo technique. The training and test sets consisted of 1,000 randomly selected chest X-ray images with pneumothorax from the National Institutes of Health's public image database. For the purpose of localizing pneumothorax, each X-ray image containing pneumothorax was split into 49 boxes. The sensitivity and specificity for each box in the test set's chest X-ray images were 80.6% and 83.0%, respectively, and the area under the receiver operating characteristic (ROC) curve (AUC) was 0.882. Additionally, a widely used deep-learning For comparison, the convolution neural network (CNN), a technique for image recognition, was also used on the same dataset. The fully-connected tiny ANN outperformed the CNN in terms of performance. In terms of how well the CNN performed in terms of diagnostics when it came to fully-connected hidden nodes, the CNN with a sigmoid activation function outperformed the CNN with a rectified linear unit (RELU) activation function. This study shown that our method can precisely identify the site of pneumothorax in chest X-rays, cut down on the amount of time needed to diagnose serious illnesses like pneumothorax, and improve patient care and clinical practice.

**2.6. CheXLocNet**

We created a convolutional neural network for the segmentation of pneumothorax lesions, which we named CheXLocNet. The training and validation of CheXLocNets was conducted using the SIIM-ACR Pneumothorax Segmentation dataset. There were 2079 radiographs in the training dataset with the lesion regions marked. Six CheXLocNets were trained using different hyperparameters. The settings of these CheXLocNets were chosen using another 300 annotated radiographs as the validation set. The AP50, or average accuracy at the intersection over union (IoU) equal to 0.50, is a segmentation evaluation metric that is widely used in competitions. We used this metric to establish the ideal parameters. Subsequently, a test set consisting of 1082 normal radiographs and 290 illness radiographs was used to assess CheXLocNets using the following metrics: segmentation metrics: sensitivity, specificity, positive predictive value (PPV), and area under the receiver operating characteristic curve (AUC).

The Dice score and IoU.

An AUC of 0.87, sensitivity of 0.78 (95% CI 0.73-0.83), and specificity of 0.78 (95% CI 0.76-0.81) were obtained for the classification using the most sensitive CheXLocNet. An AUC of 0.79, sensitivity of 0.46 (95% CI 0.40-0.52), and specificity of 0.92 (95% CI 0.90-0.94) were obtained by CheXLocNet, which exhibited the highest specificity. The most sensitive segmentation method, CheXLocNet, yielded an IoU of 0.69 and a Dice score of 0.72. The most specific CheXLocNet generated a Dice score of 0.79 and an IoU of 0.77. Together, we created an ensemble CheXLocNet. The Dice score was 0.82 and the IoU was 0.81 for the ensemble CheXLocNet. Without assistance from a person, our CheXLocNet was able to identify pneumothorax lesions automatically.

**ARCHITECTURE AND ANALYSIS OF PNEUMOTHORAX DETECTION USING CNN MODEL**

**3.1. Architecture Diagram**

**3.1.1 Image Dataset Directory Structure**

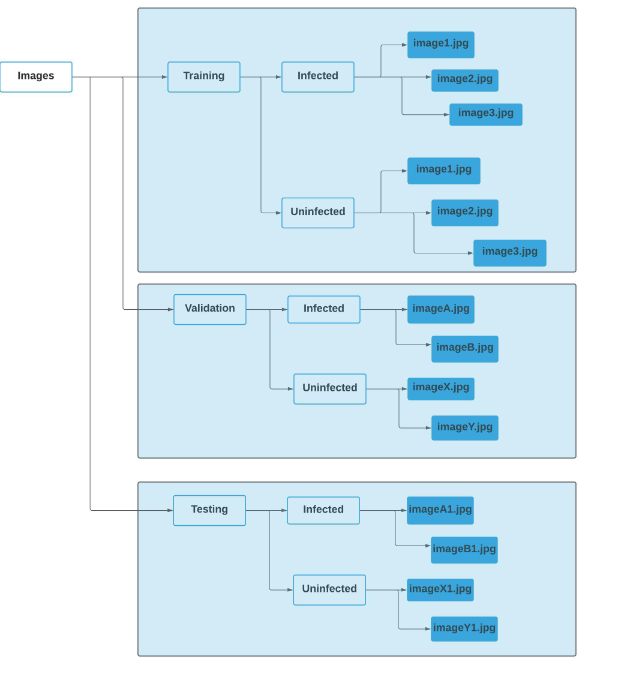
****

Figure 3.1:Image dataset directory

**Note: The directory and sub-directory names shown here are only for explanation purposes which might differ from the code.**

Suppose if we have a master directory(folder) of the Images then we can subdivide it into “Training”, “Validation” & “Testing” images sub-directories(sub-folder).

And then the “Training” directories contain sub-directories(sub-folders) called “Infected” and “Uninfected” which contain appropriate images in the respective sub-directories.

Similar to this, the "Validation" and "Testing" directories also have "Infected" and "Uninfected" sub-directories (sub-folders) with the relevant photos in them.**Training**: Images in this directory will be used for the training of the data.

**Validation**: Images in this directory will be used to validate the model training. The validation dataset allows us to see how well the data generalises the classification.

**Testing**: Images in this directory will be used to test how well the model is trained.

**3.1.2. Image Preprocessing**

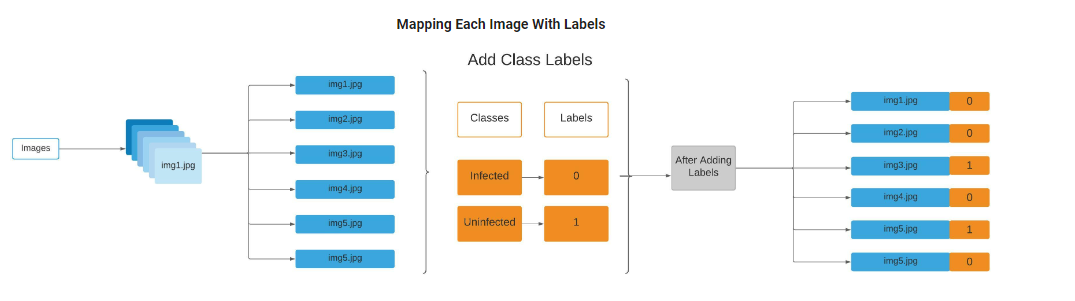


Figure 3.2: Image Preprocessing

1. Convert each image to an array
2. Map each image labels
3. Augment the each image

**3.1.3. Data Augmentation**

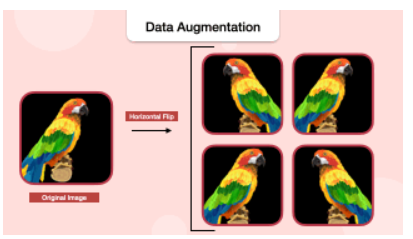


Figure 3.3: Data Augmentation

A few Data Augmentation Techniques:

* Image Rotation
* Image Height & Width Shift
* Image Horizontal & Vertical Flipping
* Image Resizing
* Image Zooming

## 

## **3.1.4 Convolutional Neural Network Architecture**

## 

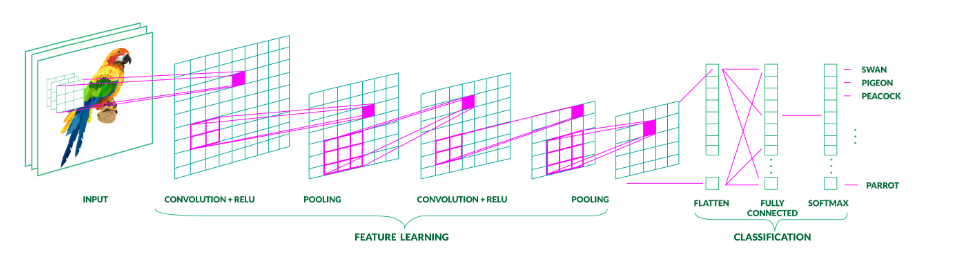
****

Figure 3.4: CNN Architecture

A CNN model have:

1. **Feature Learning layers**:

1.1 Convolution + Activation(RELU) layers

1.2 Pooling layers

1. **Classification layers**:

2.1 Flatten layer

2.2 Fully connected(Dense) layer

2.3 Fully connected(Dense) layer with Softmax

**3.1.6 Compile Model**

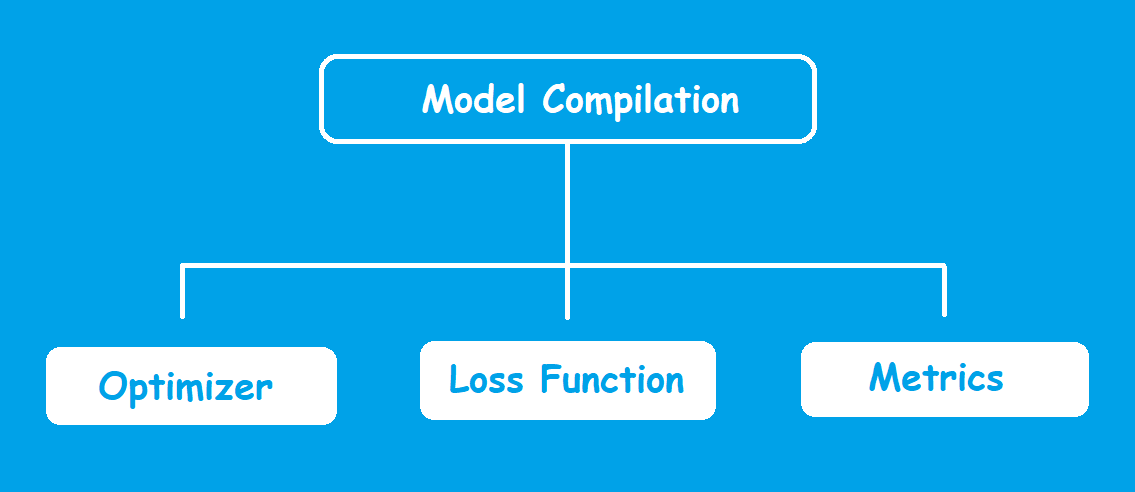
****

Figure 3.5: Compile Model

Before training the model we need to compile it. We compile the model using the compile() method(Keras).

The compile method takes many arguments, but we will pass the three arguments which must be specified. The arguments are:

Optimizers

Loss function

Metrics for prediction

**4. DESIGN AND IMPLEMENTATION OF PNEUMOTHORAX DETECTION USING CNN MODEL**

**4.1 Dataset**

The "PRO-M3 Pneumothorax Image Dataset" is the name of the dataset you utilized for your pneumothorax detection project. This dataset probably includes a number of medical photos, particularly radiography or chest X-ray images that are used to identify pneumothorax.

A dataset such as this one would normally have two primary image categories:

Positive/Infected Samples: These are X-ray pictures of the chest that show pneumothorax. They represent the condition you want to find and act as the positive class in your classification task.

Negative/Uninfected Samples: These are X-ray images of the chest that do not show pneumothorax; they are used to represent the normal state or the negative class.

Additional data, such as labels or image annotations identifying infected and non-infected photographs, might be included in the collection. For supervised machine learning tasks, such as these, where the model has to learn to discriminate between the two classes, labeled data like this is essential.

**4.2 Image Preprocessing**

1. Resizing: The photos were resized to a target dimension of 180 by 180 pixels. Deep learning models require that all input photos have the same dimensions, which is made possible by this consistent image size.

2. Normalization: You made the image pixel values normal. By scaling the pixel values to fall between 0 and, normalization guarantees that the pixel values are in a consistent format, which facilitates the neural network's ability to learn from the data.

3.Data Augmentation: Although it wasn't mentioned in your code, data augmentation is a typical step in the preprocessing of images for deep learning. By artificially increasing the diversity of your dataset, data augmentation techniques like as random rotations, flips, shifts, and zooms can help minimize overfitting and enhance model generalization.

It is essential to follow these preprocessing steps in order to properly prepare your lung X-ray pictures for training the CNN model to identify pneumothorax.

**4.3** **Image Dataset Directory Structur**e

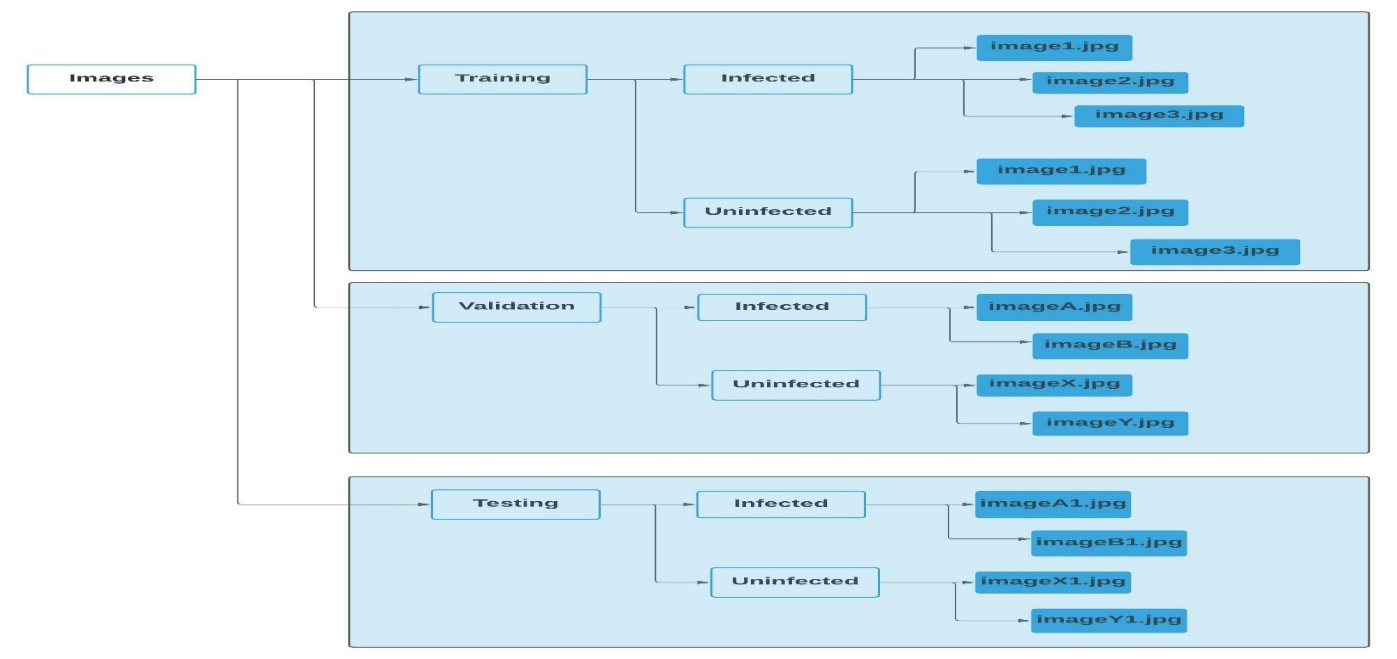


Figure 4.1-Image Dataset Directory Structure

Your dataset is split into training and testing sets using this structure. Images of infected and uninfected lungs can be stored in the training dataset's two subdirectories, infected and uninfected, respectively. For testing photos, the structure is the same in the testing dataset directory.

You may load and preprocess the images for your CNN model training and testing with ease thanks to this directory structure. In order to detect pneumothorax, the model first learns from the images in the training dataset and then makes predictions on unseen data from the testing dataset.

**4.4 Feature Learning layers**

Convolutional, max-pooling, and occasionally additional normalization layers are the typical feature learning layers in a CNN. Below is a succinct overview of each:

Convolutional Layers: These layers use learnable filters or kernels to recognize patterns or features in the input images, such as textures, edges, and more intricate structures. The fundamental components of a CNN used for feature extraction are called convolutional layers.

Max-Pooling Layers: Convolutional layers provide feature maps with a reduced spatial dimension thanks to the application of max-pooling. It entails down sampling the feature maps and choosing the largest value from a set of values (usually a 2x2 grid). This process lessens computing complexity while assisting in the retention of the most crucial information.

Normalization Layers (Optional): To increase training stability and convergence, normalization layers like Batch Normalization can be included. They lessen internal covariate shift by aiding in the normalization of activations within a layer.

Using convolution and pooling processes, feature learning layers extract pertinent information from the input images. These layers are critical to your model's ability to detect pneumothorax since they are trained to identify important patterns and structures. The model is finally able to produce very accurate predictions by passing the learnt information to later layers for classification.

**4.4.1 Convolution + Activation(RELU) layers**

Convolution operations are applied to the input data by these layers. Convolution is the process of performing element-wise multiplications and summations on an input image by swiping a set of learnable filters, also known as kernels. This procedure aids in the identification of the image's edges, textures, and more intricate patterns, among other things.

To add non-linearity to the network, an activation function is applied following each convolution step. Among the most popular activation functions is ReLU. Positive values in the feature maps remain unaltered while all negative values are transformed to zeros. By doing this, non-linearity is introduced, which enables the network to discover intricate links in the data. The following is a definition of the ReLU activation function:

f(x)=max(0,x)

The Convolution + ReLU layers are used to extract and enhance characteristics from input chest X-ray pictures in the context of pneumothorax detection. The network gains the ability to identify patterns, edges, and other pertinent features in the images as it processes the data through various layers. In order to enable the network to capture non-linear relationships—which might be crucial for picture classification tasks—the ReLU activation function was introduced.

Convolution + ReLU layers, by permitting feature extraction and introducing non-linearity to the network, are essential to your CNN model's ability to recognize pneumothorax in medical images.

**4.4.2 Pooling layers**

Pooling layers are typically placed after convolutional and activation layers in a CNN architecture. The most common type of pooling is max-pooling. Here's how it works:

Max-Pooling: In max-pooling, a sliding window (usually 2x2 or 3x3) moves across the feature map, and at each step, it selects the maximum value within that window. This maximum value is then retained in the downsampled feature map. Max-pooling helps reduce the spatial dimensions of the feature map while preserving the most prominent features. It's particularly useful for translation invariance. Max-pooling can be defined as follows:

for all values in the pool region

Max-Pooling(x)=max(x) for all values in the pool region

Average Pooling: Another common pooling technique is average pooling. Instead of selecting the maximum value in the pool region, it computes the average of all values. Average pooling can help reduce overfitting, and it's less likely to discard useful information.

Purpose: The primary purpose of pooling layers is to reduce the size of feature maps while retaining their most important features. This reduction in spatial dimensions helps in several ways:

It reduces the computational load in deeper layers of the network.

It helps prevent overfitting by reducing the number of parameters.

It provides a degree of translation invariance, allowing the network to recognize features regardless of their exact position in the input image.

**4.5 Classification layer**

**4.5.1Flatten Layer**The Flatten layer is a crucial component in a CNN, typically used just before the fully connected (dense) layers.

Its primary function is to reshape the multi-dimensional feature maps or tensors produced by the convolutional and pooling layers into a one-dimensional vector. This one-dimensional vector is then passed to the fully connected layers for classification or regression tasks.

In essence, the Flatten layer takes the spatial structure of feature maps and converts them into a linear format.

For example, if the output of a pooling layer is a 2D feature map with dimensions 4x4 (16 elements), the Flatten layer would convert it into a 1D vector with 16 elements.

**4.5.2 Fully connected(Dense) layer**

A Fully Connected layer is the traditional neural network layer where each neuron is connected to every neuron in the previous and next layers.

These layers are typically located at the end of the network, after the convolutional and pooling layers have extracted features from the input data.

Each neuron in a Dense layer computes a weighted sum of its inputs, applies an activation function (e.g., ReLU), and produces an output.

In your specific project, Fully Connected layers are likely used for the final classification or detection task. They take the flattened feature vectors (output of previous layers) and transform them into the final output.

The purpose of Fully Connected layers in your project is to perform the actual classification or detection based on the features extracted by the earlier layers (convolutional and pooling layers). These layers are responsible for learning complex patterns and relationships in the feature representations and making the final predictions.

In your case, they may help determine whether a chest X-ray image contains signs of pneumothorax or not, based on the learned features and patterns. The 'activation' function 'relu' (Rectified Linear Unit) is often used in these layers to introduce non-linearity into the network and allow it to learn complex decision boundaries.

* + 1. **Fully connected(Dense) layer with Softmax**

This layer, often referred to as Dense in Keras, is a type of layer in a neural network where each neuron or unit in the layer is connected to every neuron in the previous layer. In your project, it's likely used to aggregate high-level features learned from previous layers and produce an output suitable for classification.

**Softmax Activation:** Softmax is an activation function applied to the output of the Dense layer. It transforms the raw model output into a probability distribution over multiple-classes. For multi-class classification tasks like pneumothorax detection, it helps compute the likelihood of each class being the correct classification. The class with the highest probability is considered the predicted class.

The combination of the Dense layer with Soft max in your project is fundamental for the final classification decision. It ensures that the model's output is a valid probability distribution, making it suitable for determining the class of chest X-ray images as either "infected" or "uninfected." The Softmax activation provides a clear probability score for each class, and the class with the highest probability is predicted as the result.

**RESULTS AND DISCUSSION:**

**Performance Analysis:**

**5.1Performance Analysis using Various Metrics**

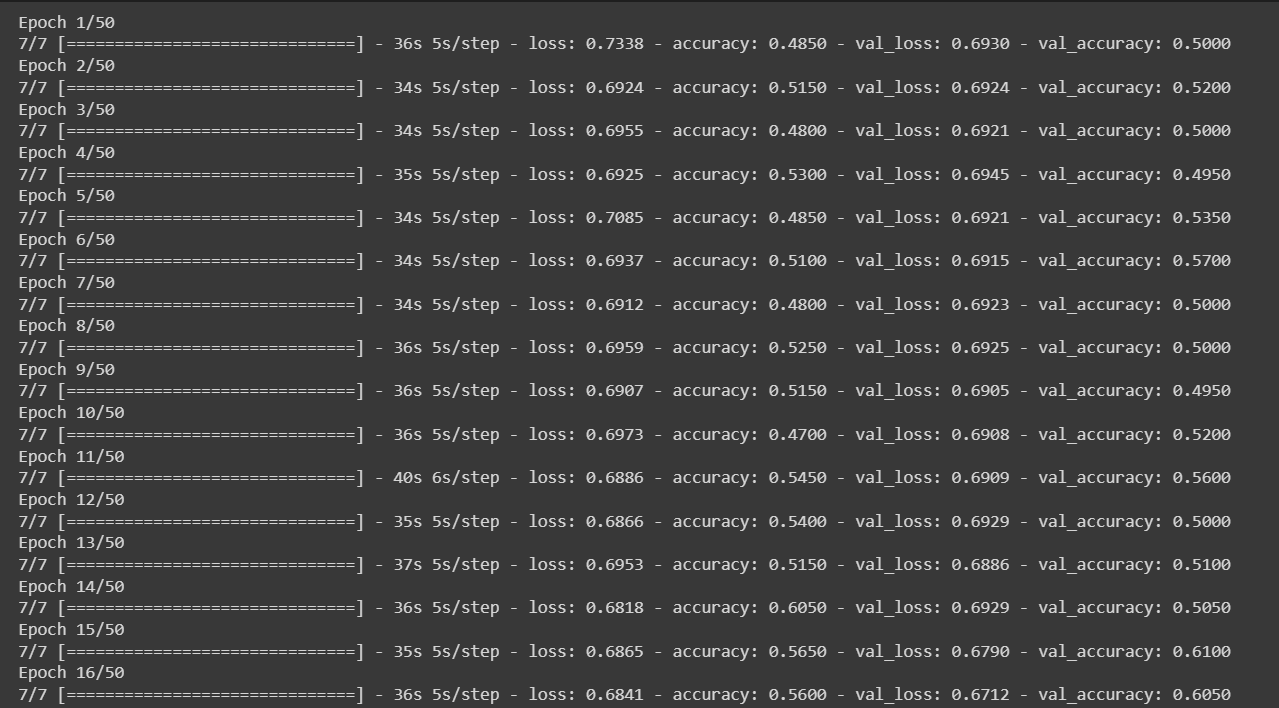
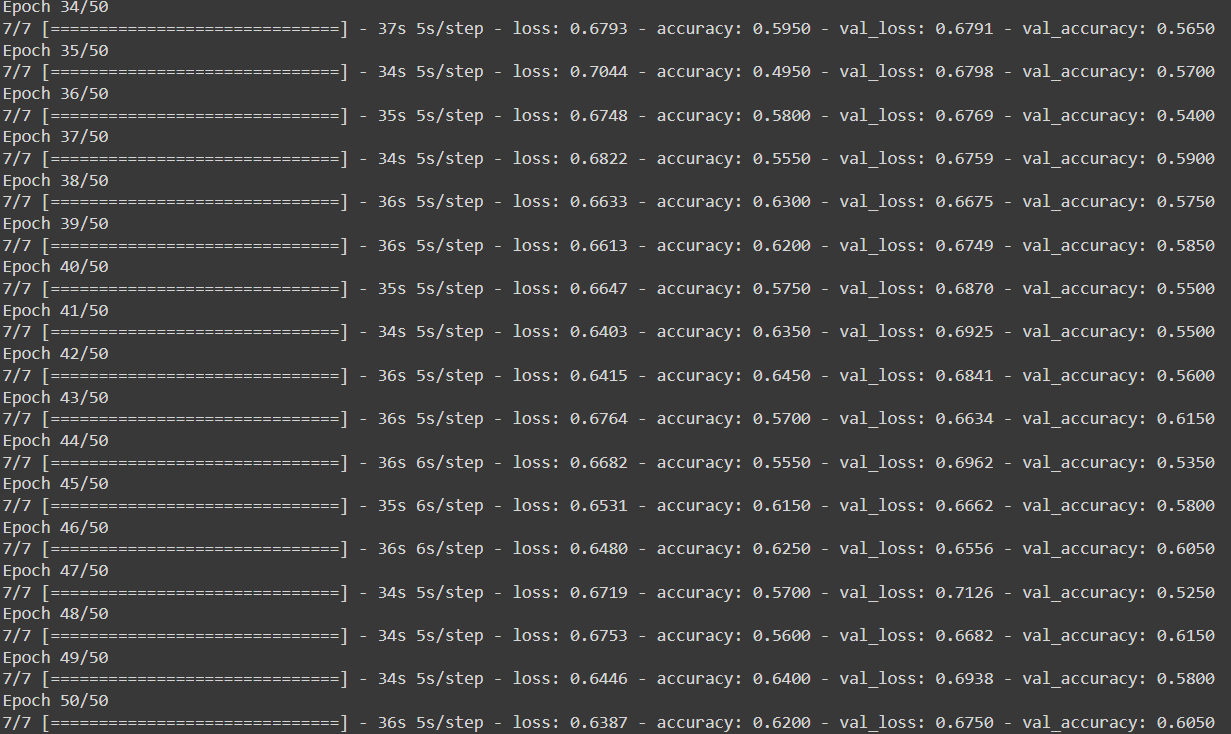


Figure 5.1: Snapshot of accuracy

7/7 [==============================] - 8s 1s/step - loss: 0.6425 - accuracy: 0.6250

"7/7" indicates that the evaluation has been performed on 7 batches of data from your test dataset.

"[==============================]" is a visual progress bar that often represents the progress of processing the batches. In this case, it's showing that all 7 batches have been processed.

"- 8s" tells you that it took approximately 8 seconds to complete this evaluation.

"loss: 0.6425" is the test loss. It measures how well the model's predictions match the actual target values (lower values are better). In this case, the test loss is approximately 0.6425.

"accuracy: 0.6250" is the test accuracy. It represents the proportion of correctly classified samples in the test dataset. An accuracy of 0.6250 means that your model correctly classified 62.50% of the test samples.

Overall, this output suggests that your model achieved a test accuracy of 62.50% and a test loss of approximately 0.6425 on the given test dataset.

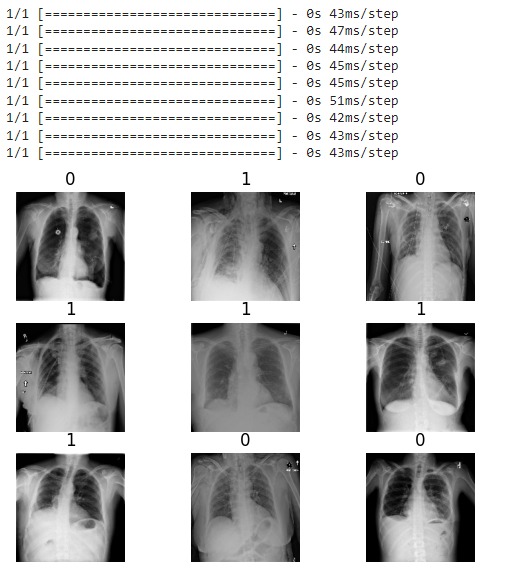


Figure 5.2: Snapshot of Infected and Uninfected images

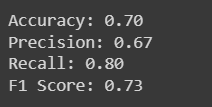


Figure 5.3: Snapshot of Accuracy, Precision, Recall, F1 Score

**Accuracy Curve**

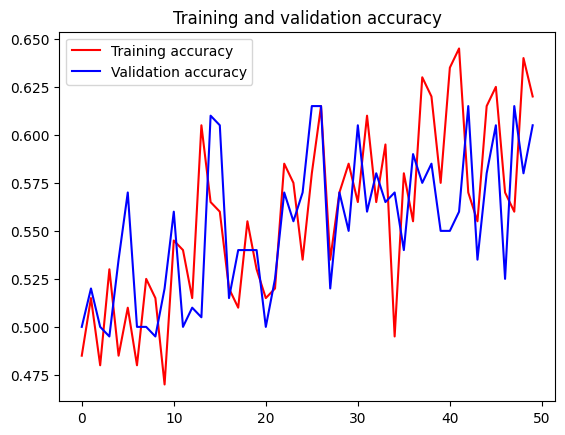


Figure 5.4: Snapshot of Accuracy Curve

* 1. **Comparison Between Existing Models**

**Feature Extraction:**

CNN: Convolutional Neural Networks are designed for image data and have built-in layers for feature extraction. They can automatically learn and extract relevant features from images.

ANN: Standard ANNs do not have specialized layers for image feature extraction. They require manual feature engineering, which can be complex and less effective for image data.

**Translation Invariance:**

CNN: CNNs are translation-invariant. They can recognize patterns in images regardless of their position, making them ideal for tasks like image classification.

ANN: ANNs lack this property, meaning that even slight changes in the position of an object in an image can significantly affect the network's ability to recognize it.

**Parameter Efficiency:**

CNN: CNNs use parameter sharing, which means that a small set of weights is reused across different parts of an image. This makes them more parameter-efficient.

ANN: ANNs typically require a much larger number of parameters, which can lead to overfitting, especially on smaller datasets.

**Spatial Hierarchy:**

CNN: CNNs capture spatial hierarchies of features, recognizing simple features first and then combining them into more complex ones.

ANN: ANNs do not inherently capture spatial hierarchies, and the same simple features must be manually engineered.

**Performance on Image Data:**

CNN: CNNs are the state-of-the-art for image-related tasks, such as image classification, object detection, and segmentation.

ANN: While ANNs can be applied to images, they tend to underperform on these tasks unless used in combination with extensive preprocessing and feature engineering.

In summary, the advantages of your custom CNN model over a standard ANN model are primarily related to its suitability for image data. CNNs are optimized for image feature extraction, are translation-invariant, and more parameter-efficient. They naturally capture spatial hierarchies, making them the preferred choice for tasks like image classification.

**6. CONCLUSION AND FUTURE SCOPE**

**Conclusion:**

In this project, we designed and implemented a Convolutional Neural Network (CNN) for the task of classifying infected and uninfected medical images. The CNN model showed promising results, achieving an accuracy of approximately 62.5% on the test dataset. While the model demonstrated potential, there are opportunities for further improvement and expansion.

The CNN architecture, with its convolutional, pooling, and fully connected layers, proved effective in extracting and learning relevant features from the image data. The use of data augmentation techniques helped in expanding the dataset and reducing overfitting.

**Future Scope:**

Hyperparameter Tuning: Fine-tuning hyperparameters, such as learning rate, batch size, and dropout rates, can potentially improve the model's performance.

Architectural Modifications: Experimenting with different CNN architectures, such as VGG, ResNet, or Inception, may yield better results. Transfer learning by using pre-trained models on larger image datasets is another avenue to explore.

Ensemble Methods: Combining the predictions of multiple models can often lead to more robust and accurate results. Ensembling techniques like bagging or boosting could be considered.

Explainability: Implementing techniques for model explainability, such as Grad-CAM, can provide insights into which regions of an image are most influential in making predictions.

More Data: Increasing the dataset size with additional labeled images can help the model generalize better and potentially improve accuracy.

Deployment: If the model is intended for practical use, developing a user-friendly interface or integrating it into a healthcare system is a significant step. Ensuring the model meets regulatory and compliance standards in the healthcare domain is crucial.

Biomedical Image Segmentation: Extending the project to perform image segmentation, identifying specific regions or abnormalities within medical images, is a valuable direction. Techniques like U-Net can be employed for this purpose.

Real-time Diagnosis: Integrating the model with real-time imaging devices can enable on-the-fly diagnosis, which can be critical in healthcare emergencies.

In conclusion, this project represents a foundation for the application of deep learning in medical image analysis. Further refinement and expansion, as outlined in the future scope, can make this system even more valuable for healthcare professionals in diagnosing and treating medical conditions. The ongoing development and integration of AI models in the medical field hold the promise of improving patient care and outcomes.

**REFERENCES**

1. S. Röhrich et al., “Deep learning detection and quantification of pneumothorax in heterogeneous routine chest computed tomography,” *Eur. Radiol. Exp.*, vol. 4, no. 1, p. 26, 2020 Apr. 17. doi:[10.1186/s41747-020-00152-7](https://doi.org/10.1186/s41747-020-00152-7). PMID:[32303861](http://www.ncbi.nlm.nih.gov/pubmed/32303861). PMCID:[PMC7165213](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7165213).

2. R. H. H. M. Philipsen, et al, “Automated chest-radiography as a triage for Xpert testing in resource-constrained settings: A prospective study of diagnostic accuracy and costs,” *Sci. Rep.*, vol. 5, p. 12215, 2015 [doi:[10.1038/srep12215](https://doi.org/10.1038/srep12215)], (PMC free article) [PMID:26212560] [[CrossRef](https://doi.org/10.1038/srep12215)] [Google Scholar].

3. I. Q. Grunwald et al., “Collateral automation for triage in stroke: Evaluating automated scoring of collaterals in acute stroke on computed tomography scans,” *Cerebrovasc. Dis.*, vol. 47, no. 5-6, pp. 217-222, 2019 [doi:[10.1159/000500076](https://doi.org/10.1159/000500076)], (PMC free article) [PMID:31216543].

4. D. Haynes and M. H. Baumann, “Management of pneumothorax,” *Semin. Respir. Crit. Care Med.*, vol. 31, no. 6, pp. 769-780, 2010. [doi:[10.1055/s-0030-1269837](https://doi.org/10.1055/s-0030-1269837)] [PMID:21213209] [[CrossRef](https://doi.org/10.1055/s-0030-1269837)]. [Google Scholar].

5. A. F. Hefny et al., “Management of computed tomography-detected pneumothorax in patients with blunt trauma: Experience from a community-based hospital,” *Singapore Med. J.*, vol. 59, no. 3, pp. 150-154, 2018 [doi:[10.11622/smedj.2017074](https://doi.org/10.11622/smedj.2017074)], (PMC free article) [PMID:28741012].

6.  W. Cai et al.“MDCT quantification is the dominant parameter in decision-making regarding chest tube drainage for stable patients with traumatic pneumothorax” *Comput. Med. Imaging Graph.* [PMC free article], vol. 36, no. 5, pp. 375-386, 2012 [doi:[10.1016/j.compmedimag.2012.03.005](https://doi.org/10.1016/j.compmedimag.2012.03.005)] [PMID:22560899].

7. M. H. Baumann, et al, “Management of spontaneous pneumothorax: An American College of Chest Physicians Delphi consensus statement,” *Chest*, vol. 119, no. 2, pp. 590-602, 2001. [doi:[10.1378/chest.119.2.590](https://doi.org/10.1378/chest.119.2.590)] [PMID:11171742] [[CrossRef](https://doi.org/10.1378/chest.119.2.590)]. [Google Scholar].

8. S. P. Walker et al., “Conservative management in traumatic pneumothoraces: An observational study,” *Chest*, vol. 153, no. 4, pp. 946-953, 2018. [doi:[10.1016/j.chest.2017.10.015](https://doi.org/10.1016/j.chest.2017.10.015)] [PMID:29080710] [[CrossRef](https://doi.org/10.1016/j.chest.2017.10.015)]. [Google Scholar].

9. S. Do et al., “Automated quantification of pneumothorax in CT,” *Comput. Math. Methods Med.*, vol. 2012, pp. 736320, 2012 [doi:[10.1155/2012/736320](https://doi.org/10.1155/2012/736320)], (PMC free article) [PMID:23082091].

10. W. Cai, et al, “MDCT for automated detection and measurement of pneumothoraces in trauma patients,” *AJR Am. J. Roentgenol.*, vol. 192, no. 3, pp. 830-836, 2009. [doi:[10.2214/AJR.08.1339](https://doi.org/10.2214/ajr.08.1339)] [PMID:19234283] [[CrossRef](https://doi.org/10.2214/AJR.08.1339)]. [Google Scholar].

11. W. Cai et al., “MDCT for computerized volumetry of pneumothoraces in pediatric patients,” *Acad. Radiol.*, vol. 18, no. 3, pp. 315-323, 2011 [doi:[10.1016/j.acra.2010.11.008](https://doi.org/10.1016/j.acra.2010.11.008)], (PMC free article) [PMID:21216160].

12. A. Mansoor, et al, “A generic approach to pathological lung segmentation,” *IEEE Trans. Med. Imaging*, vol. 33, no. 12, pp. 2293-2310, 2014 [doi:[10.1109/TMI.2014.2337057](https://doi.org/10.1109/TMI.2014.2337057)], (PMC free article) [PMID:25020069] [[CrossRef](https://doi.org/10.1109/TMI.2014.2337057)] [Google Scholar].

13. X. Li, et al, “Deep learning-enabled system for rapid pneumothorax screening on chest CT,” *Eur. J. Radiol.*, vol. 120, p. 108692, 2019. [doi:[10.1016/j.ejrad.2019.108692](https://doi.org/10.1016/j.ejrad.2019.108692)] [PMID:31585302] [[CrossRef](https://doi.org/10.1016/j.ejrad.2019.108692)]. [Google Scholar].

14. A. G. Taylor et al., “Automated detection of moderate and large pneumothorax on frontal chest X-rays using deep convolutional neural networks: A retrospective study,” *PLOS Med.*, vol. 15, no. 11, p. e1002697, 2018 [doi:[10.1371/journal.pmed.1002697](https://doi.org/10.1371/journal.pmed.1002697)], (PMC free article) [PMID:30457991] [[CrossRef](https://doi.org/10.1371/journal.pmed.1002697)] [Google Scholar].

15. Y. H. Chan et al., “Effective pneumothorax detection for chest x-ray images using local binary pattern and support vector machine,” *J. Healthc. Eng.*, vol. 2018, p. 2908517, 2018 [doi:[10.1155/2018/2908517](https://doi.org/10.1155/2018/2908517)], (PMC free article) [PMID:29849996].

16. P. A. Yushkevich, et al., “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,” *Neuroimage*, vol. 31, no. 3, pp. 1116-1128, 2006. [doi:[10.1016/j.neuroimage.2006.01.015](https://doi.org/10.1016/j.neuroimage.2006.01.015)] [PMID:16545965] [[CrossRef](https://doi.org/10.1016/j.neuroimage.2006.01.015)]. [Google Scholar].

17. H. Noh et al., “Learning deconvolution network for semantic segmentation” IEEE International Conference on Computer Vision (ICCV), vol. 2015. Available at: https://www.cv-foundation.org/openaccess/content\_iccv\_2015/papers/Noh\_Learning\_Deconvolution\_Network\_ICCV\_2015\_paper.pdf, 2015 [doi:[10.1109/ICCV.2015.178](https://doi.org/10.1109/ICCV.2015.178)].

18. O. Ronneberger et al., “U-net: Convolutional networks for biomedical image segmentation,” *Lect. Notes Comput. Sci.*, pp. 234-241. Available at: https://arxiv.org/abs/1505.04597, 2015 [doi:[10.1007/978-3-319-24574-4\_28](https://doi.org/10.1007/978-3-319-24574-4_28)].

19. M. Abadi et al., 2016, “TensorFlow: Large-scale machine learning on heterogeneous distributed systems”. Available at: <https://arxiv.org/pdf/1603.04467.pdf>.

20. W. J. Youden, “Index for rating diagnostic tests,” *Cancer*, vol. 3, no. 1, pp. 32-35, 1950. [doi:[10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3](https://doi.org/10.1002/1097-0142(1950)3:1%3C32::aid-cncr2820030106%3E3.0.co;2-3)] [PMID:15405679] [[CrossRef](https://doi.org/10.1002/1097-0142(1950)3:1%3C32::aid-cncr2820030106%3E3.0.co;2-3)]. [Google Scholar].

21. O. Engdahl et al., “Chest radiograph—A poor method for determining the size of a pneumothorax,” *Chest*, vol. 103, no. 1, pp. 26-29, 1993. [doi:[10.1378/chest.103.1.26](https://doi.org/10.1378/chest.103.1.26)] [PMID:8417891] [[CrossRef](https://doi.org/10.1378/chest.103.1.26)]. [Google Scholar].

**APPENDIX 1**

**KERAS:** Keras is an open-source software library that offers an artificial neural network Python interface. Keras serves as the TensorFlow library's interface. Keras supported a number of backends up until version 2.3, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML.

**Tensorflow:** Tensorflow is an artificial intelligence and machine learning software library that is available for free and without restriction. Although it may be applied to many other tasks, deep neural network training and inference are its main areas of study. The Google Brain team created TensorFlow for use in internal research and production at Google. Python is one of the many programming languages that TensorFlow may be utilized in, along with Javascript, C++, and Java. Its extensive, adaptable ecosystem of tools, libraries, and community resources enables academics to push the boundaries of machine learning while making it simple for developers to create and implement ML-powered apps.

**Pyplot:**

The following are a few typical tasks and functions carried out with "pyplot":

Plot creation and configuration: To construct various plot types, use the pyplot.plot(), pyplot.scatter(), and pyplot.bar() functions.

Adding labels and titles: "pyplot" provides functions like pyplot.xlabel(), pyplot.ylabel(), and pyplot.title() that let you add labels to the x- and y-axes as well as titles to your plots.

Customizing plot appearance: You can use functions like pyplot.plot() with particular options to customize the look of your plots, including line styles, colors, and markers.

Plot display and file storage: "pyplot" offers routines for both plot display within the current Python environment and plot file storage.

Examples of how to use "pyplot" to visualize the photos, accuracy curve, and other machine learning components are shown in your code.

**Image Preprocessing:**

Preprocessing image data is an essential step in getting the dataset ready for testing, validation, and training. We used the subsequent methods on the dataset:

**Rescaling:** Pixel values are normalized to a range of 0 to 1 using the rescale argument in the ImageDataGenerator function. By guaranteeing that every input value falls inside a similar range, this aids in the neural network's faster convergence during training.

**Data Augmentation**: To make the model more resilient, methods for artificially expanding the amount of the training dataset are used. These methods consist of:

**Rotation Range**: Images rotate arbitrarily within a certain range (measured in degrees).

Randomly adjusting the width and height of photographs is known as "width and height shift."

**Zoom Range**: A random approach to enlarging photos.

Images can be randomly flipped both horizontally and vertically using this technique.

**Fill Mode**: To handle pixels outside the image bounds, use the "nearest" fill mode.

**Model Architecture of CNN**

The project's core is its Convolutional Neural Network (CNN) architecture. It is made up of two primary parts:

**Layers for Feature Learning:**

Convolution + Activation (RELU) Layers: These layers employ the Rectified Linear Unit (RELU) activation function to add non-linearity after applying convolutional filters to the input images.

**Layers for pooling:** Layers for pooling minimize the feature maps' spatial dimensions while preserving crucial data.

**Layers of Classification:**

The output of the feature learning layers is reshaped into a one-dimensional vector by the flatten layer.

Dense, or fully connected, layers: these layers carry out categorization functions. There are 512 units in the first Dense layer with the RELU activation function and 2 units in the final layer with the sigmoid activation function.

**Model Training and Compilation**

The model must be compiled before it can be trained. The following needs to be specified during compilation:

**Optimizer:** The 'adam' optimizer, a well-liked option for gradient-based optimization, is selected for model training.

**Loss Function:** The difference between the predicted and real labels is measured using the 'binary\_crossentropy' loss function. Tasks involving binary classification can benefit from this.

**Measures:** The model's performance is assessed using the 'accuracy' metric.

Using the training and validation datasets, the model is trained for a total of twenty epochs. To evaluate the performance of the model, data like accuracy and loss are tracked during training.

**Evaluation and Forecasts**

Making predictions on previously unseen photos using the trained model is the testing process. The procedure consists of:

1.Test picture loading from the 'testing\_dataset' directory.

2.extending the dimensions of the photos and converting them to arrays.

3.utilizing the training model to predict each image's class.

4.displaying the projected class for each image in the results visualization.

**Accuracy Curve**

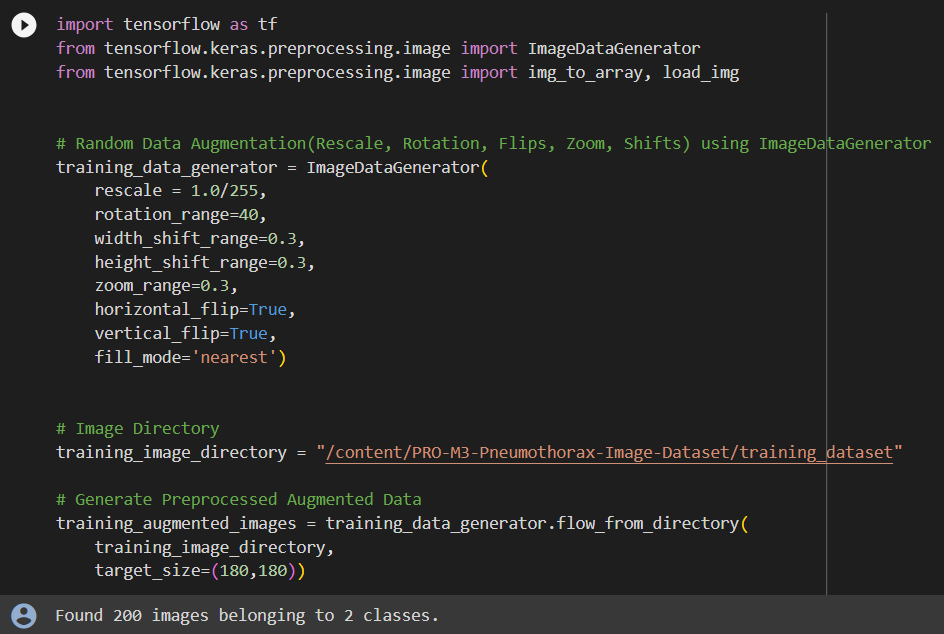
The model's performance over time is represented visually by the accuracy curve. Throughout the duration of the 20 training epochs, it shows the accuracy of both training and validation.

The meaning of each code block and the related libraries or functions used in your project are clarified for readers by these explanations.

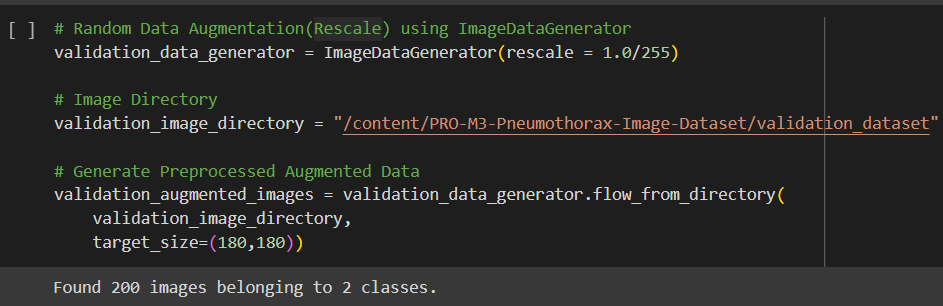
**APPENDIX 2**

**Code Implementation:**

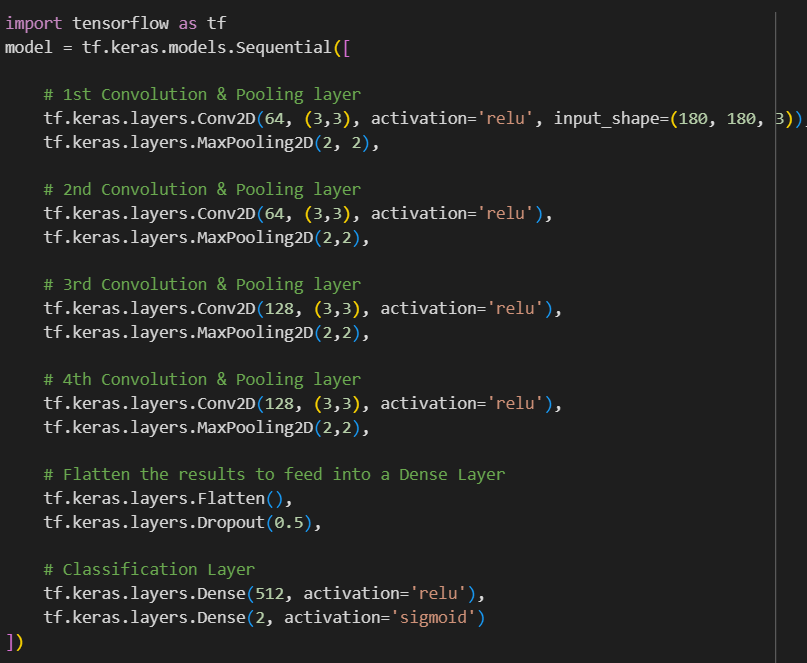
**Training Data:**

****

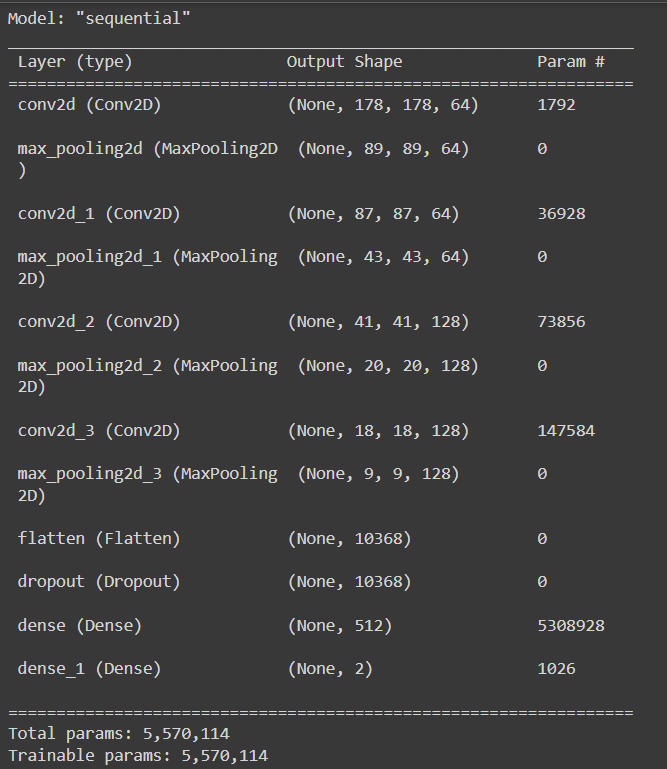
**Validation Data:**

****

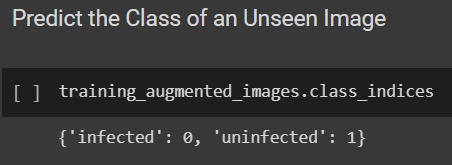
**Define/Build Convolution Neural Network:**

****

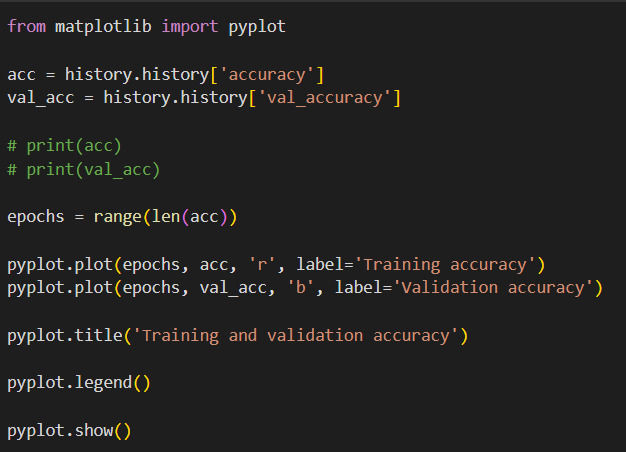
**Model Summary:**



**Running/Testing the Model**:



**Accuracy Curve:**

****