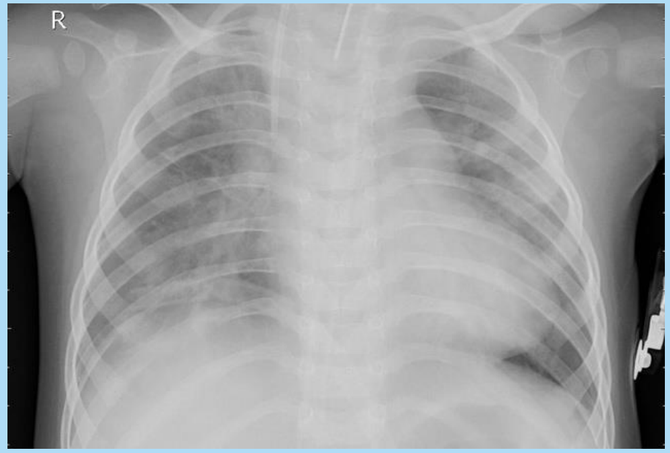
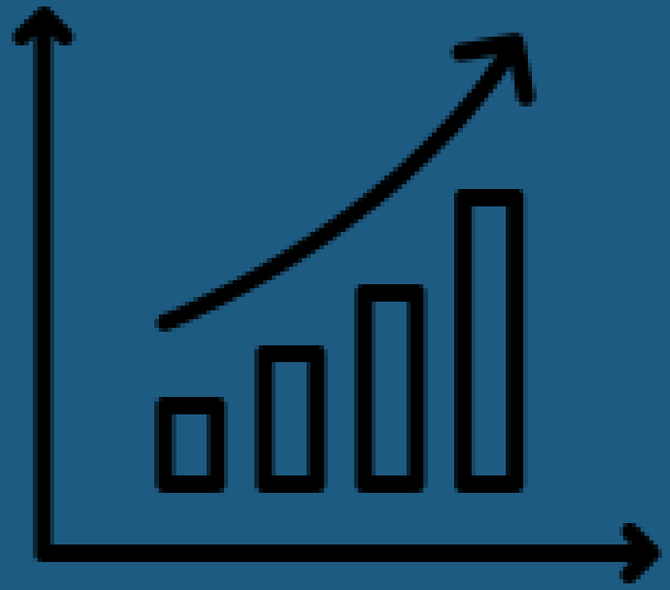
# Detection of Lung Diseases using Deep Learning

## Does a patient have either COVID-19 or Pneumonia?



# IMPACT OF PNEUMONIA AND COVID-19

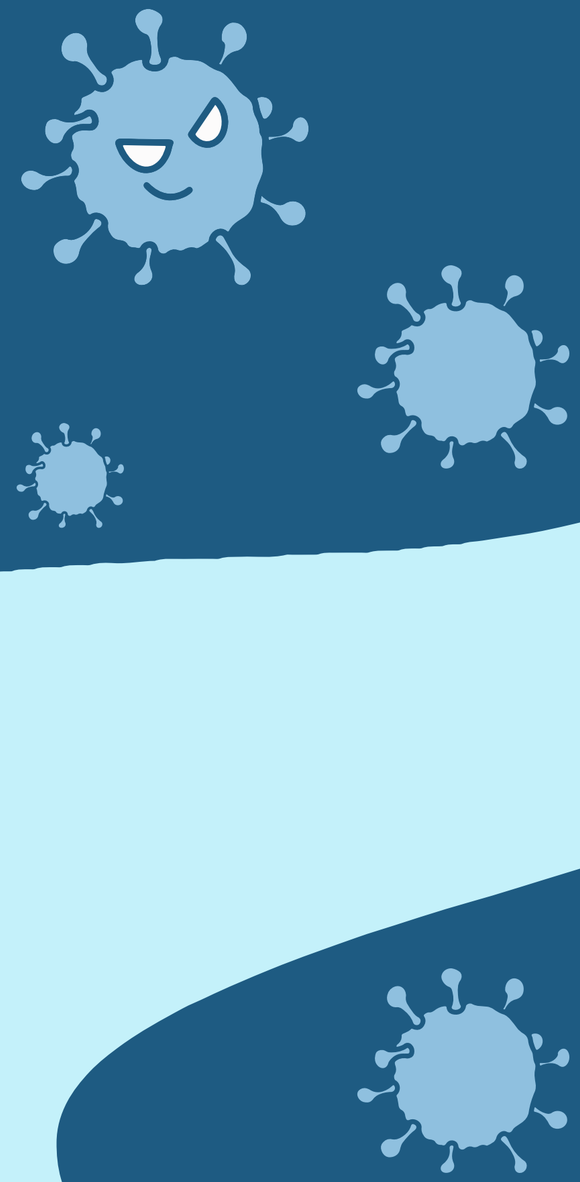


# WHY WE NEED THE HELP OF AUTOMATED DETECTION ?

# Overview

**The aim of the project was:**

- to develop a CNN model that can categorize X-ray images of chest into three categories: ***Pneumonia***, ***Normal*** *and* ***COVID-19***



# Project Structure

01 Data Preprocessing

02 Data Augmentation

03 Model Definition

04 Training and Evaluation

# Data Preprocessing

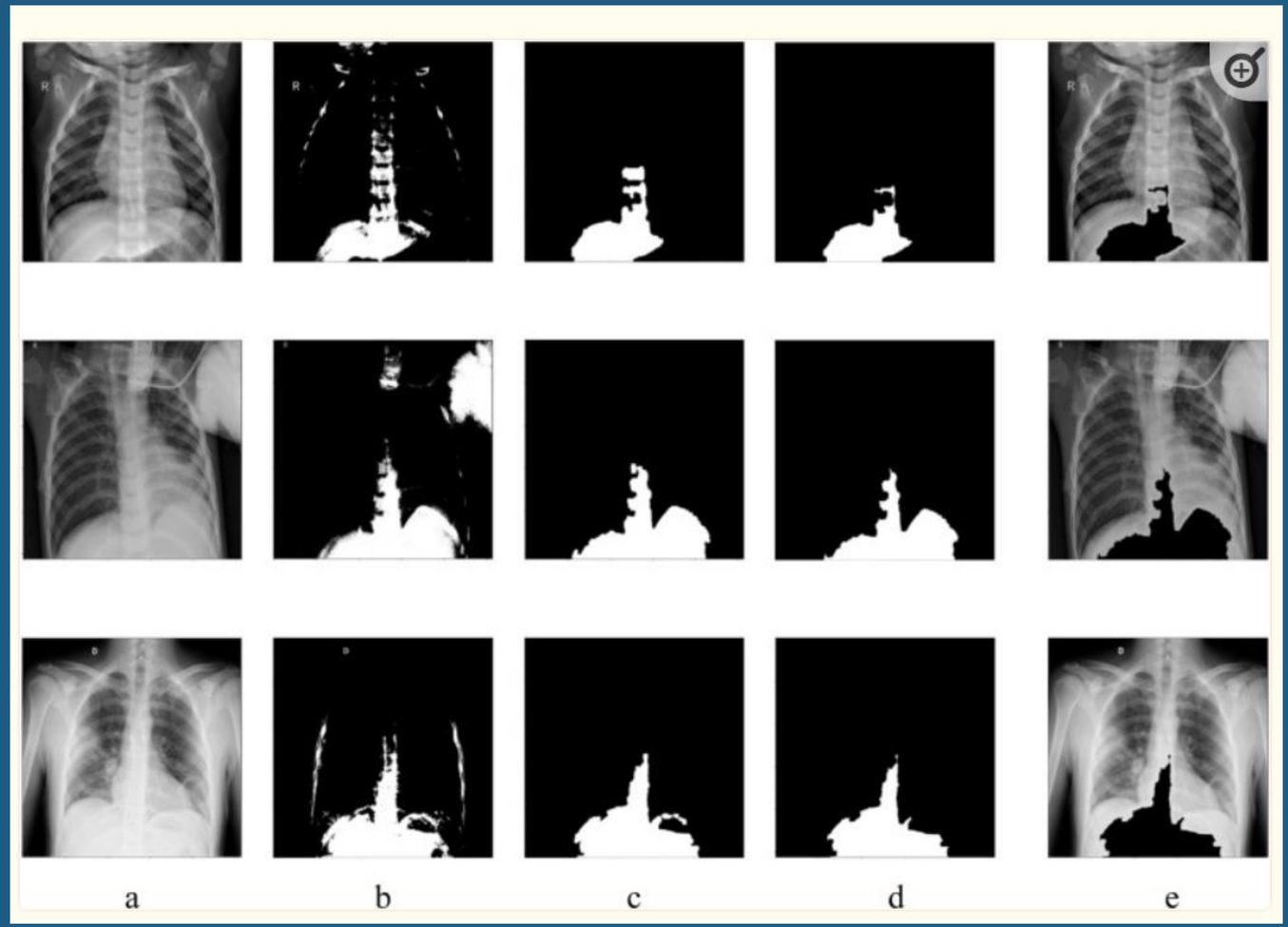
- The code **loads** images from directories of three classes: ‘pneumonia’, ‘normal’, and ‘covid’.
- Images are converted to **grayscale** and **resized** to **150x150** pixels and **224x224** pixels for different models.

Image Preprocessing includes **diaphragm removal algorithm, splitting the image into three different channels, applying bilateral filter and histogram equalization**

normal

covid

pneumonia



# Data Preprocessing

- The code **loads** images from directories of three classes: ‘pneumonia’, ‘normal’, and ‘covid’.
- Images are converted to **grayscale** and **resized** to **150x150** pixels and **224x224** pixels for different models.

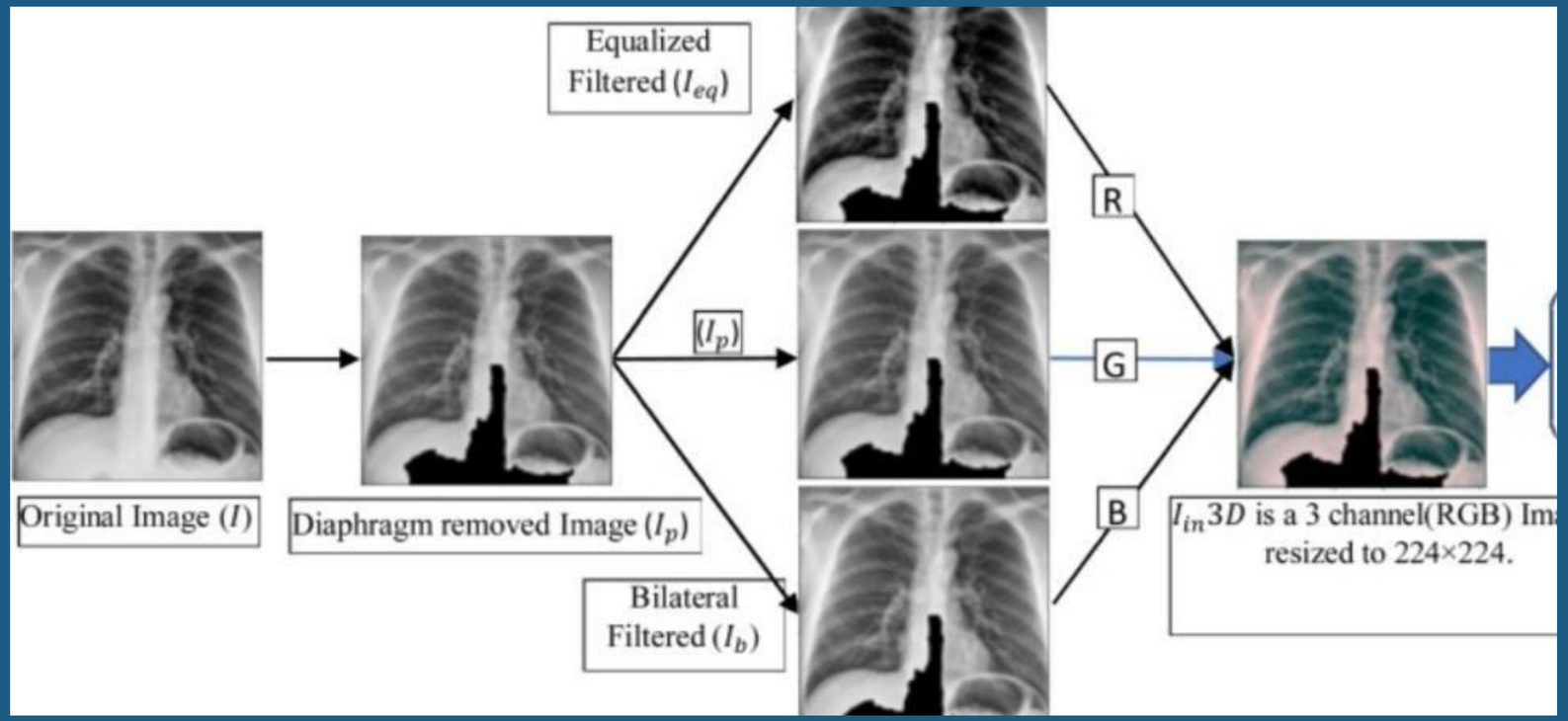
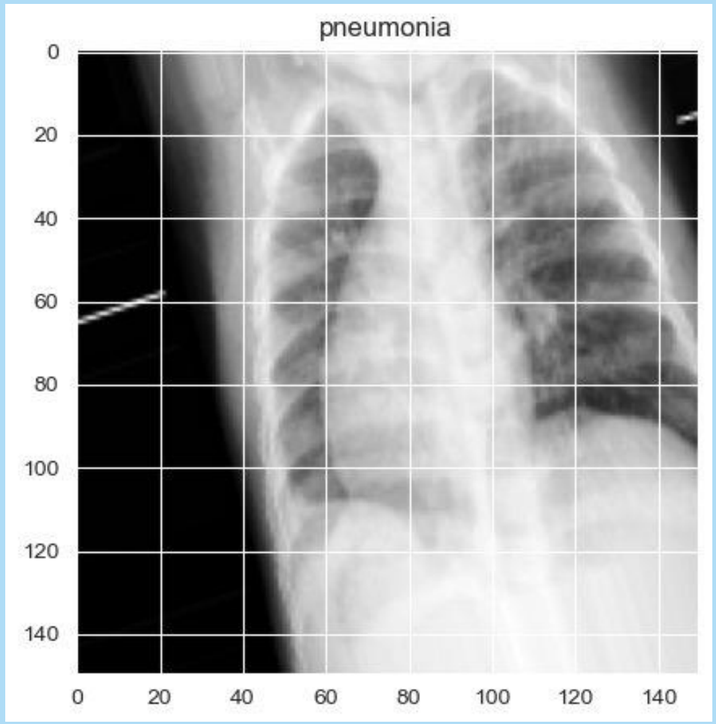
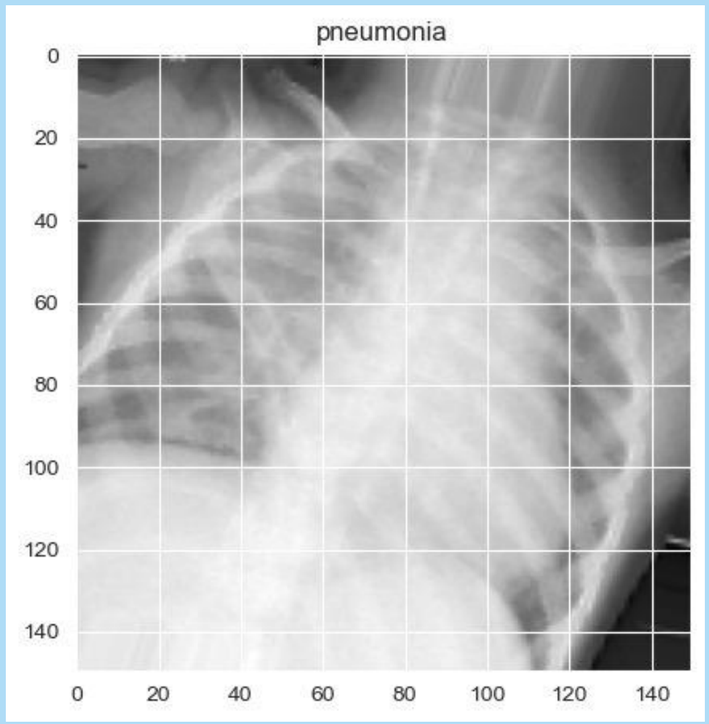



Image Preprocessing includes **diaphragm removal algorithm, splitting the image into three different channels, applying bilateral filter and histogram equalization**

# Data Augmentation

Data augmentation was implemented to increase the size of the dataset by applying various image transformations, such as randomly zooming, flipping, shifting

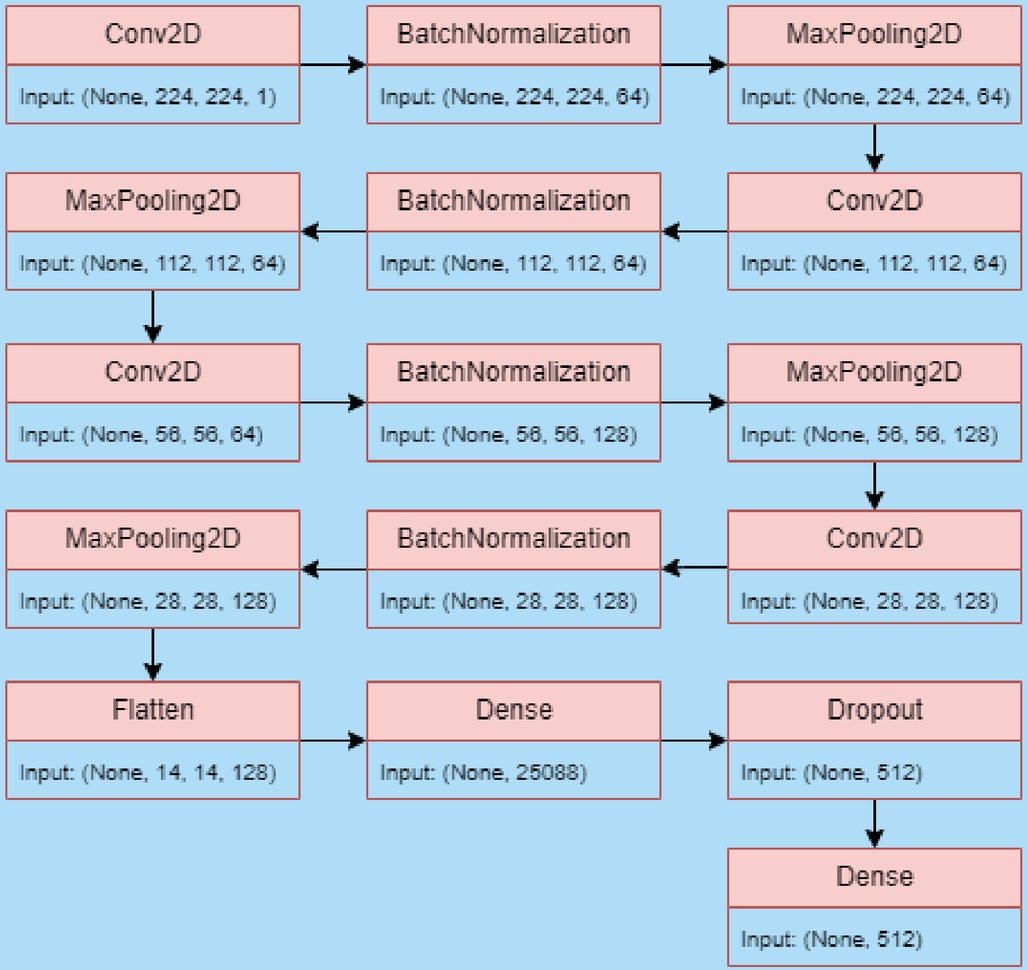


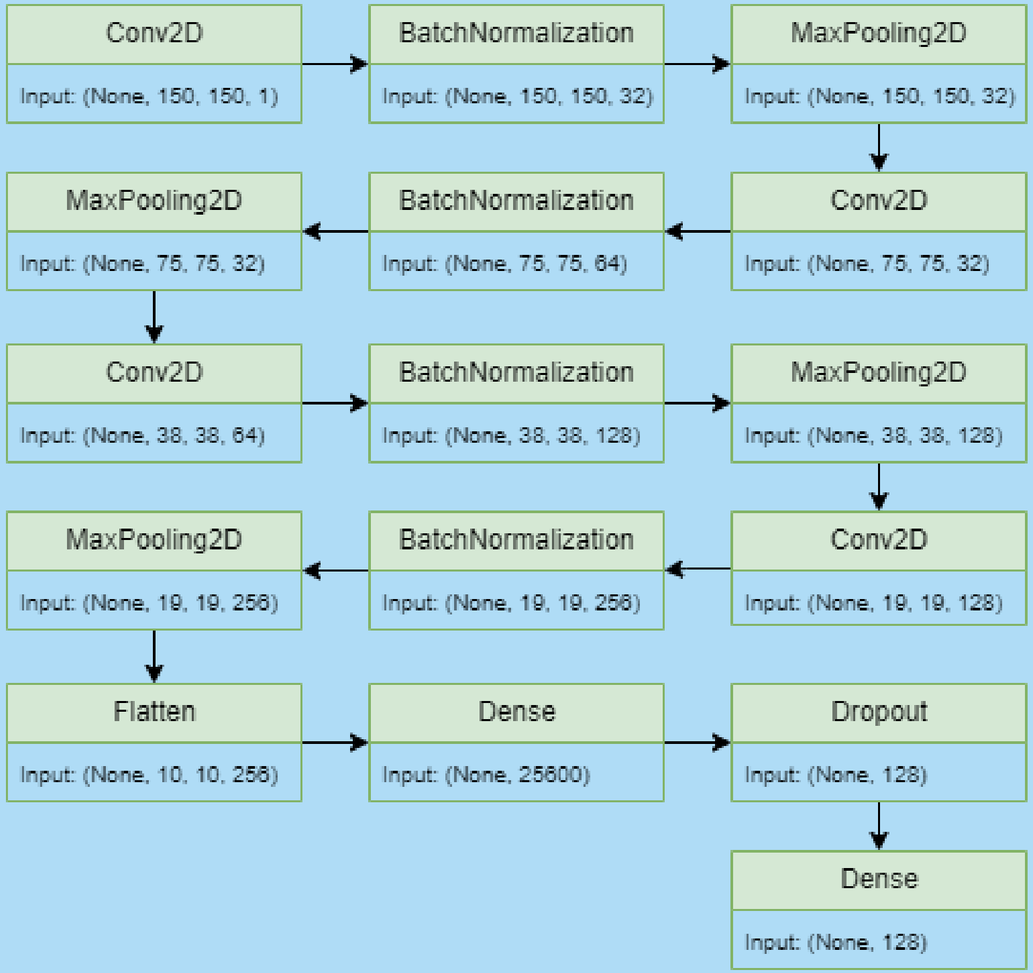

```
rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest'
```

# Model Architecture Visual Representation

## Model 1




## Model 2



# Training and Evaluation

## Model 1

using normal dataset

### Version 1

- 50 epochs
- Batch size 32
- Using Learning Rate Reduction with 0.0010 as the starting learning rate

using preprocessed dataset

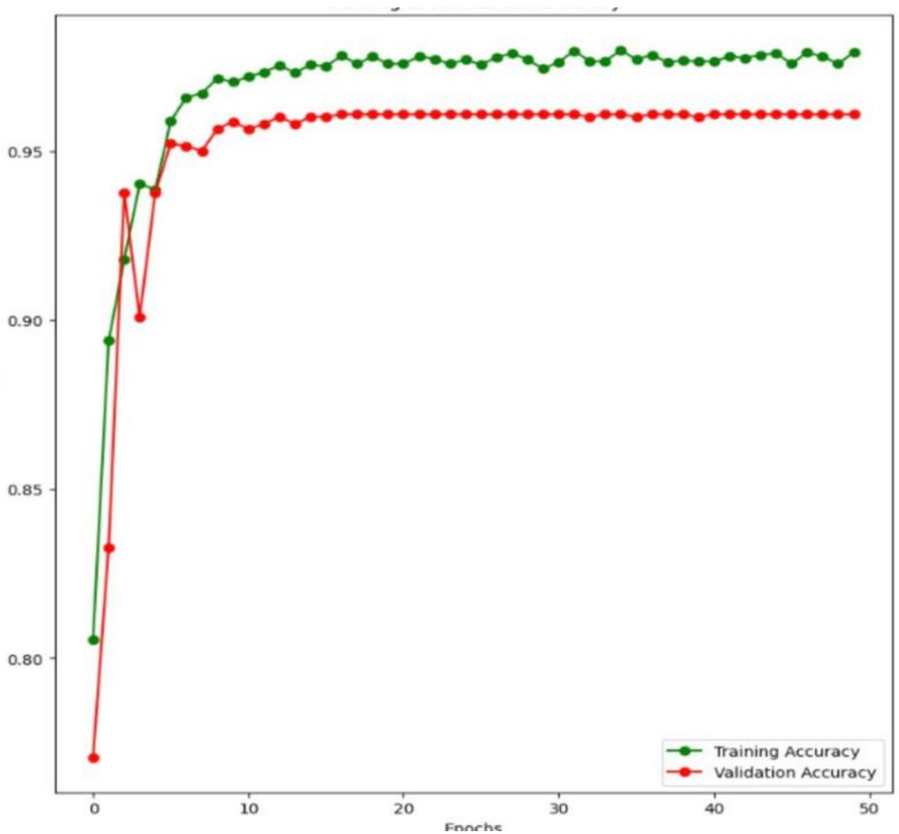
### Version 2

- 50 epochs
- Batch size 32
- Using Learning Rate Reduction with 0.0010 as the starting learning rate
- Early stopping was implemented to prevent overfitting

## Model 2

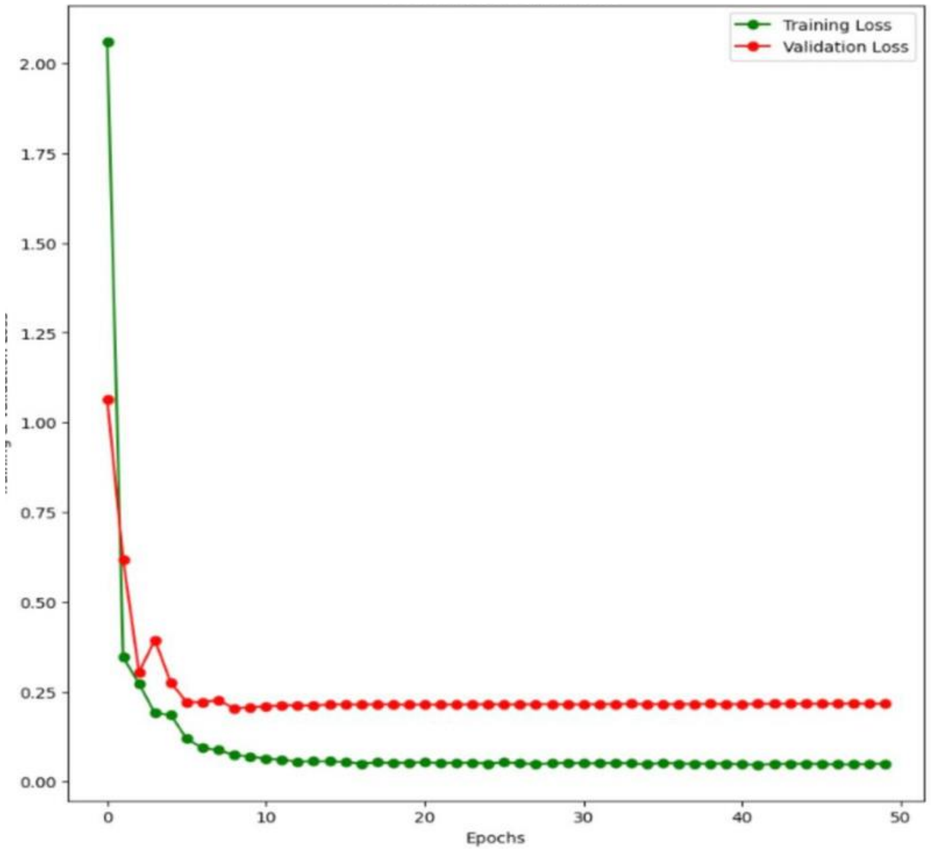
- 50 epochs
- Batch size 32
- Using Learning Rate Reduction with 0.0010 as the starting learning rate
- Early stopping and learning rate reduction were implemented to prevent overfitting

## Accuracy




99.55% on training
99.68% on validation
96.05% on testing

## Loss

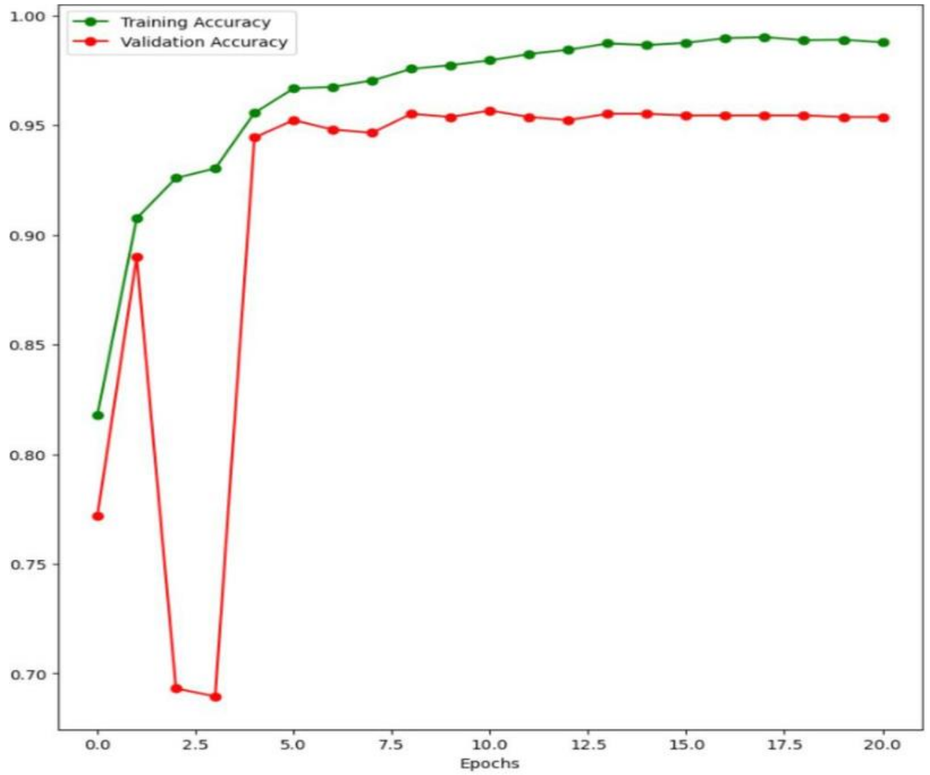



0.0179 on training
0.265 on validation
0.2041 on testing

# Result

**Module 1 version 1**
Accuracy and Loss

## Accuracy

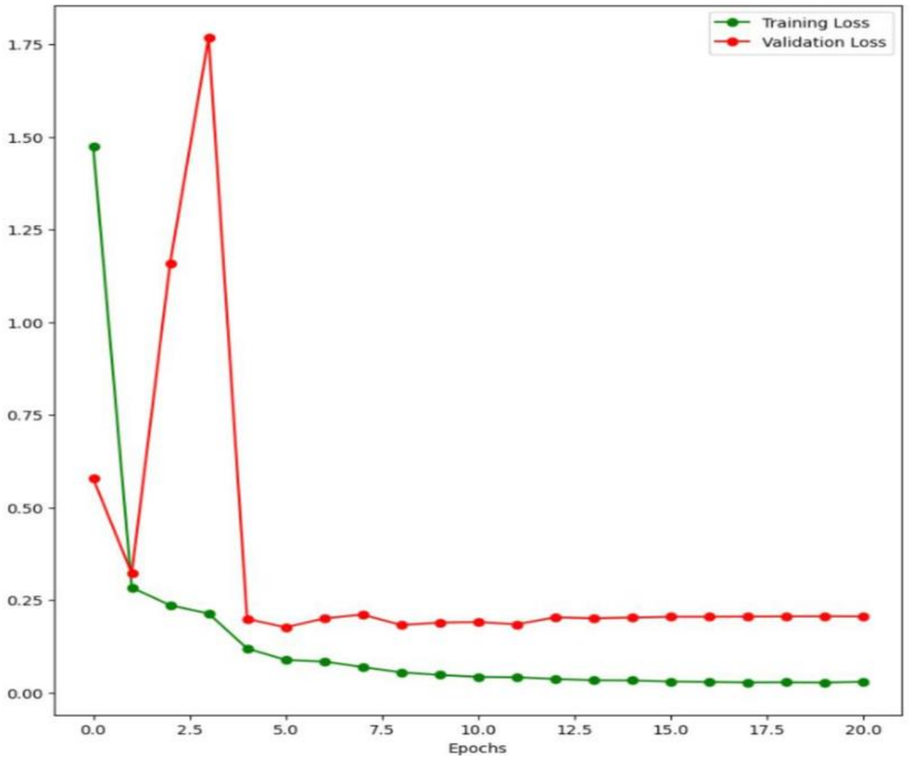



97.65% on training

95.51% on validation

95.29% on testing

## Loss




0.0556 on training

0.1825 on validation

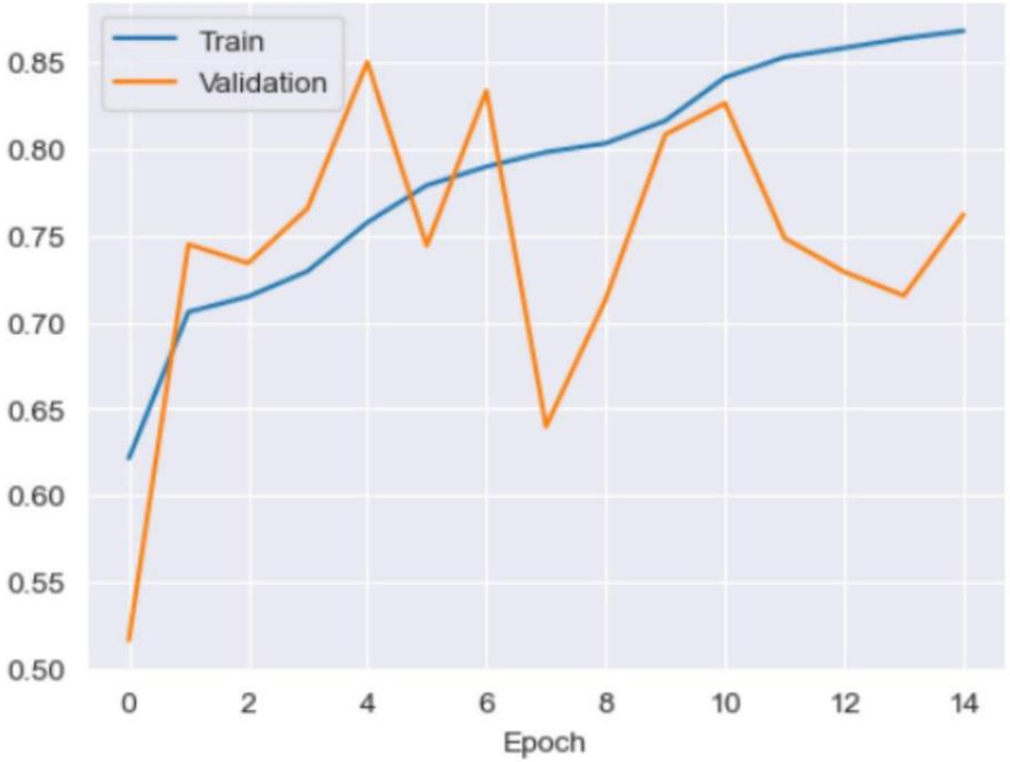
0.2681 on testing

# Result

**Module 1 version 2**

Accuracy and Loss

## Accuracy

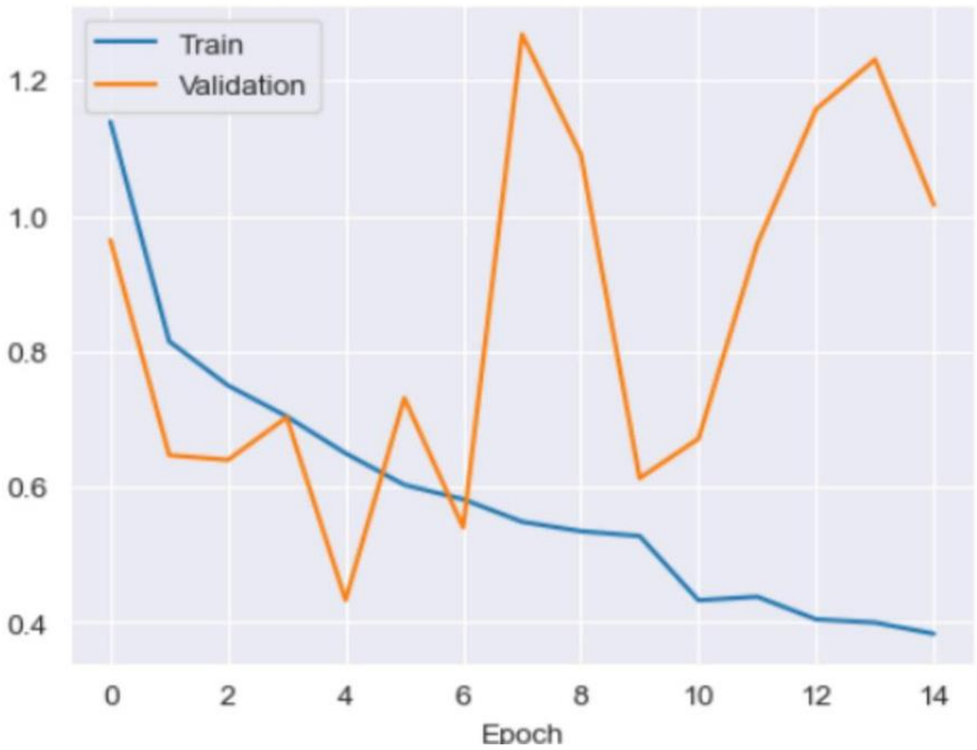



87.60% on training

85.01% on validation

85.69% on testing

## Loss




0.3894 on training

0.4332 on validation

0.4403 on testing

# Result

**Module 2**

Accuracy and Loss

# The Future

We will investigate the reason why preprocessing images having a negative impact on model 1 and looking for more method to improve the accuracy of the model.

# QR CODE

GITHUB

# THANK YOU