EFFICIENT PNEUMONIA DETECTION IN CHEST XRAY IMAGES USING DEEP TRANSFER LEARNING

*Thesis submitted in partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications** of the **APJ Abdul Kalam Technological University***

submitted by

NOWRIN
(TCR19MCA020)

DEPARTMENT OF COMPUTER APPLICATIONS

GOVERNMENT ENGINEERING COLLEGE
THRISSUR - 680009

MAY 2022

DEPARTMENT OF COMPUTER APPLICATIONS
GOVERNMENT ENGINEERING COLLEGE, THRISSUR
THRISSUR, KERALA STATE, PIN 680009

CERTIFICATE

This is to certify that the main project titled **”EFFICIENT PNEUMONIA DETECTION IN CHEST XRAY IMAGES USING DEEP TRANSFER LEARNING”** *is a bonafide work done by* **NOWRIN (TCR19MCA020)** *under my supervision and guidance, and is submitted in May 2022 in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications from APJ Abdul Kalam Technological University(KTU).*

Hredya Sundaran	Soumia Chandran M	Dr. Sminesh C N
Project Guide	**Project Coordinator**	**Head of Department**

Place : THRISSUR
Date : 13-05-2022

DECLARATION

I hereby declare that the main project named, **Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published another person nor material which has been accepted for the award of any other degree or course of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : THRISSUR	Signature
Date : 13/05/2022	**NOWRIN(TCR19MCA020)**

ACKNOWLEDGEMENT

I would like to thank Department of Computer Application, GEC Thrissur, for giving me this opportunity to pursue this project and successfully complete it.

I am highly indebted to my guide **Prof. Hredya Sundaran** and project coordinator **Prof. Soumia Chandran M** for their guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project.

I also express special thanks to the Head of the Computer Applications Department, **Dr. Sminesh C N** for his keen support and consistent encouragement in our academic activities

I sincerely thank all other faculties of MCA department for guiding through the processes involved in the project .

ABSTRACT

Pneumonia is one of the largest infectious diseases that cause death in children and elderly people across the globe. It has been ranked eighth in the list of the top 10 causes of death in the United States. Due to pneumonia, every year, nearly 3.7 lakh children die in India, which constitutes a total of fifty percent of the pneumonia deaths that occur in India. Chest X-rays are primarily used for the diagnosis of this disease. However, even for a trained radiologist, it is a challenging task to examine chest X-rays. There is a need to improve the diagnosis accuracy. In this work, an efficient model for the detection of pneumonia, trained on digital chest X-ray is proposed, which could aid the radiologists in their decision-making process. The detection model is developed using Transfer learning technique which is used to fine-tune the deep learning models. Several deep learning models such as DenseNet121, MobileNetV3, ResNet152V2 InceptionV3, Sequential, VGG16 are trained for detection, among which the model with highest accuracy is being chosen to integrate with UI. The final proposed weighted classifier model will be able to achieve a test accuracy of 85%+ on the unseen data from pneumonia dataset. Hence, the proposed model can be used for a quick diagnosis of pneumonia and can aid the radiologists in the diagnosis process.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

Pneumonia causes the death of around 700,000 children every year and affects 7% of the global population. According to the WHO, “Every year, it kills an estimated 1.4 million children under the age of five years, accounting for 18% of all deaths of children under five years old worldwide. Chest X-rays are primarily used for the diagnosis of this disease. However, even for a trained radiologist, it is a challenging task to examine chest X-rays. There is a need to improve the diagnosis accuracy. In this work, an efficient model for the detection of pneumonia trained on digital chest X-ray images is proposed, which could aid the radiologists in their decision-making process.

1.2 Motivation

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

1.3 Objective

In this project, an efficient model for the detection of pneumonia, trained on digital chest X-ray is proposed, which could aid the radiologists in their decision-making process. The detection model is developed using Transfer learning technique which is used to fine-tune the deep learning models. It is a supervised learning approach in which the network predicts the result based on the quality of the data set used. Several deep learning models such as DenseNet121, MobileNetV3, ResNet152V2 InceptionV3, Sequential, VGG16 are trained for detection, among which the model with highest accuracy is being chosen to integrate with UI .The final proposed weighted classifier model will be able to achieve a test accuracy of 85%+ on the unseen data from pneumonia data set. Hence, the proposed model can be used for a quick diagnosis of pneumonia and can aid the radiologists in the diagnosis process.

The initial stage of the project deals with proper Data Collection and Data Augmentation. Following this , the development of customized transfer learning models which have been chosen has to be done. After the development of these customized models , they should be trained and then finally evaluated with proper evaluation metrics such as accuracy to find the best model. The model is then being integrated with the android mobile application UI to predict the results.

1.4 Contribution

The major contributions in this project are:

- Designed and developed a new system for detecting pneumonia from chest Xray.
- How to minimise human intervention in detecting pneumonia.
- How to effectively identify pneumonia in a timely manner.

CHAPTER 2
LITERATURE REVIEW

In the United States, pneumonia is listed eighth among the top ten causes of mortality. Every year, 3.7 lakh children die in India as a result of pneumonia, accounting for half of all pneumonia deaths in the country. Simple actions can be used to prevent it, and low-cost, low-tech medication and care can be used to treat it. As a result, there is an urgent need to conduct research and development on computer-aided diagnosis in order to reduce pneumonia-related mortality, particularly in youngsters.

A recent paper [1] has been published in which an efficient model for the detection of pneumonia trained on digital chest X-ray images is proposed, which could aid the radiologists in their decision making process. The weighted predictions from state-of-the-art deep learning models such as ResNet18, Xception, InceptionV3, DenseNet121, and MobileNetV3 are combined in an optimal fashion using a novel approach based on a weighted classifier. This is a supervised learning technique in which the network predicts the outcome based on the dataset's quality. To improve training and validation accuracy, transfer learning is utilised to fine-tune deep learning models. To extend the training dataset in a balanced fashion, partial data augmentation approaches are used. All of the separate models are outperformed by the suggested weighted classifier.

Another paper [2] suggests implementing a the deep-learning (DL) scheme to detect the pneumonia. A binary classification achieved with the SoftMax classifier unit confirms the disease detection performance of the DL scheme. During this evaluation, 2000 photos (1000 healthy and 1000 pneumonia) are analysed, and the appropriate performance metrics are generated to confirm the results. AlexNet's experimental results on the considered image database showed a better 98%percent accuracy.

Another paper [3] suggests various deep convolutional neural network (CNN) architectures to extract features from chest X-ray pictures and classify the images to detect the existence of pneumonia in a person with greater accuracy. The real-world dataset provided by the National Library of Medicine for research purposes was used in this study. The goal of the research is to demonstrate how deep learning can assist us in recognising disease in its early stages, making it ideal for treatment, which was previously impossible.

Another paper [4] proposed an aim to detect COVID-19 patients and Pneumonia patients from X-Rays which is one of the medical imaging modes to analyse the health of patient's lung inflammation. For the detected dataset, the best Convolutional Neural Network Model is chosen. On a real-world dataset of lung X-Ray pictures, the model detects COVID-19 patients and Pneumonia patients. Normal, COVID-19, and Pneumonia are among the classifications for which images are pre-processed and trained. Following pre-processing, disease identification is accomplished by picking appropriate features from each dataset's photos. The result shows the COVID against Normal and COVID vs Pneumonia detection accuracy. COVID vs Normal has a higher level of accuracy than COVID vs Pneumonia.

CHAPTER 3
ENVIRONMENTAL STUDY

3.1 System Configuration

System configuration describe the hardware and software requirement of the system for development

3.1.1 Hardware Requirements

- Memory : 4 GB of RAM
- Processor : Intel Core i3 or equivalent CPU
- Speed : 2.4 GHz
- Proper Internet Connection

3.1.2 Software Requirements

- Operating system : Windows 8 or above
- Front End : Android , Java
- Back End : python
- IDE Used : Google Colab , Android studio
- Libraries : Tensorflow , Numpy , Keras

3.2 Software Specification

3.2.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Guido van Rossum designed it between 1985 and 1990. This open-source language is friendly and simple to use, with a focus on code readability. It's noted for its minimal code length and clear syntax. Python is a platform-independent language, which means that Python-based applications can run on a wide range of operating systems without the need for an interpreter. It's also dependable, accessible, and extremely productive.

Python is also the foundational language for Machine Learning Engineers and Data Scientists. It is indisputable that machine learning is transforming the mobile app market. Python is an appropriate backend language in this field of technology because of its emphasis on code readability and its huge libraries and frameworks. Because the language is simple and consistent, developers may design dependable systems while focusing on machine learning concerns. They won't have to worry about the difficulties that come with complex programming languages. Furthermore, Python contains a large number of machine learning libraries, such as Keras, TensorFlow, and others.

3.2.2 Android Studio

Built on JetBrains' IntelliJ IDEA software and designed exclusively for Android development, Android Studio is the official integrated development environment (IDE) for Google's Android operating system. It can be downloaded for Windows, Mac OS X, and Linux operating systems. It takes the position of the Eclipse Android Development Tools (ADT) as the primary IDE for developing native Android apps. The current stable version includes the following features: Support for Gradle-based builds, Android-specific refactoring, and bug fixes Performance, usability, version compatibility, and other issues are caught with lint tools. ProGuard's app-signing and integration features, Create standard Android designs and components using template-based wizards. Users may drag-and-drop UI components in a powerful layout editor, and layouts can be previewed on numerous screen configurations. The ability to create Android Wear apps is supported. Google Cloud Platform support is built-in, allowing for connection with Firebase Cloud Messaging (formerly 'Google Cloud Messaging'), Google App Engine, and the Android Virtual Device (Emulator) to run and debug apps in the Android studio.

3.2.3 Java

Java is one of the powerful general-purpose programming languages, created in 1995 by Sun Microsystems (now owned by Oracle). Java is an object-oriented programming language. It is not regarded true object-oriented, though, because it supports basic data types (like int, char, etc). The syntax of Java is similar to that of C/C++. However, Java does not provide low-level programming features such as pointers. In addition, Java code is usually written as classes and objects. Android significantly relies on the Java programming language, and all of the SDKs needed to create Android apps rely on Java's standard libraries. Java is simple to learn if you come from a traditional programming background like C or C++.

3.2.4 Google Colab

Google Research's Colaboratory, or "Colab" for short, is a product. Colab is a web-based Python editor that allows anyone to write and run arbitrary Python code. It's notably useful for machine learning, data analysis, and education. Colab is a hosted Jupyter notebook service that doesn't require any setup and offers free access to computational resources, including GPUs. The distinction between Jupyter and Colab is that Jupyter is an open source project, whereas Colab is a commercial product. Without having to download, install, or run anything, Colab allows you to utilise and share Jupyter notebooks with others. As a programmer, you may use Google Colab to do the following: Python allows you to write and run code. Make a list of all the code you've written that supports mathematical equations. Notebooks can be created, uploaded, and shared. Google Drive notebooks can be imported and exported. GitHub notebooks can be imported and published. External datasets, such as those from Kaggle, can be imported. Using a free GPU, integrate PyTorch, TensorFlow, Keras, OpenCV, and the Free Cloud service.

3.2.5 TensorFlow

TensorFlow is a Google-developed open-source framework for using machine learning, deep learning, and other analytical and statistical work features. It was created, like previous platforms, to make the process of building and deploying advanced analytic algorithms for users like data scientists easier. And the usual forecasts and coders. TensorFlow works with data sets presented as graphical interfaces, with the edges linking the nodes to the graph representing vectors or multidimensional matrices, resulting in tensors.

Keras was imported from Tensorflow. Tensorflow includes libraries for categorization, data splitting, and other tasks (validation and training). To convert an array to HDF5 binary data format, the h5py package was utilised. Using the h5py package, we built validation and training files in the h5 format.

3.2.6 Numpy

NumPy is a Python library. It includes support for big matrices and multidimensional arrays, as well as a vast set of high-level mathematical functions for manipulating them. Because they are both interpreted, NumPy in Python provides capability comparable to MATLAB. They enable users to develop quick applications as long as the majority of operations are performed on arrays or matrices rather than scalars.

3.2.7 Keras

Keras is a tiny neural network library written largely in Python that can be used with TensorFlow or Theano. It was created with the goal of allowing for quick testing. It puts the user experience front and centre. Keras employs the most effective methods for reducing the burden of understanding: it provides consistent and straightforward APIs, decreases the amount of actions required by the user for common use cases, and delivers clear and effective feedback on user error. The neural layers, cost functions, optimizers, implementation schemes, implementation functions, and generic programmes, in particular, are all stand-alone modules that can be combined to form new models.

3.2.8 Tensorflow Lite

TensorFlow Lite is a free deep learning framework that allows you to run TensorFlow models on your mobile device. TensorFlow Lite is a collection of tools for running TensorFlow models on mobile, embedded, and IoT devices. TensorFlow Lite is a model interpreter that executes specially optimised models on a variety of hardware. TensorFlow Lite is a lightweight version of TensorFlow Mobile that is the next step up from TensorFlow Mobile. Tensorflow light is aimed towards mobile and embedded device developers, allowing them to create next-generation programmes on platforms like as Android, iOS, Raspberry Pi, and others. It allows for low latency and tiny binary size on-device machine learning inference. The machine learning is created using Tensorflow lite. For the prediction of chest X-rays, a convolutional neural network model was used. TensorFlow includes stable Python (for version 3.7 across all platforms) and C APIs, as well as C++, Go, Java, JavaScript, and Swift (which has been archived and development has stopped). C, Haskell, Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal all have third-party packages.

3.3 Functional Requirements

The system should be designed to upload the chest X-ray images so that it can predict the category or class to which the image belongs to.

3.4 Performance Requirements

The system would need least 4 GB of RAM. Less RAM will result in the poor performance of the system.Proper internet connection is a must. Python should be pre-installed.

CHAPTER 4
SYSTEM ANALYSIS

System analysis by definition means a deliberate examination process in order to collect information, decipher the truth, diagnose the problem and use it to either develop a new framework or prescribe improvements in the current framework. A good system study involves looking to better strategies and systems to improve a business situation. At the heart of the investigation phase are the needs of the framework and the problems the customer seeks to explain, regardless of how the requirements are implemented.

4.1 Requirements Analysis

Requirements analysis, likewise called requirements engineering, is the way toward deciding client desires for another or changed item. These highlights, called necessities, must be quantifiable, significant and itemized. In programming designing, such prerequisites are frequently called useful determinations. The point of the undertaking is to recognize, see and make profitable grievance management.

This project aims to develop an efficient model for the detection of pneumonia, trained on digital chest X-ray is proposed, which could aid the radiologists in their decision-making process. The detection model is developed using Transfer learning technique which is used to fine-tune the deep learning models. Several deep learning models such as DenseNet121, MobileNetV3, ResNet152V2 InceptionV3, Sequential, VGG16 are trained for detection, among which the model with highest accuracy is being chosen to integrate with UI .

4.2 Existing System

One of the following tests can be done for pneumonia diagnosis :

- chest X-rays
- CT of the lungs
- Ultrasound of the chest
- Needle biopsy of the lung and
- MRI of the chest.

4.3 Limitations of existing system

- Currently, chest X-rays are one of the best methods for the detection of pneumonia, but even for a trained radiologist, it is a challenging task to examine chest X-rays.
- X-ray imaging is preferred over CT imaging because CT imaging typically takes considerably more time than X-ray imaging
- Sufficient high-quality CT scanners may not be available in many underdeveloped regions
- Lack of technologies
- Time consuming
- Slow decision making

4.4 Proposed System

This project aims to develop an efficient model for the detection of pneumonia, trained on digital chest X-ray is proposed, which could aid the radiologists in their decision-making process. The detection model is developed using Transfer learning technique which is used to fine-tune the deep learning models. Several deep learning models such as DenseNet121, MobileNetV3, ResNet152V2 InceptionV3, Sequential, VGG16 are trained for detection, among which the model with highest accuracy is being chosen to integrate with UI .

4.5 Advantages of proposed System

- Design and develop a new system for detecting pneumonia from chest Xray that could help doctors to make better decision.
- Less human intervention
- Less time consuming

4.6 Feasibility Study

A feasibility study is a significant level container form of the whole System examination and Design Process. The examination starts by grouping the problem definition.Feasibility is to decide whether the task is worth attempted. When an acknowledgment issue definition have been created, the expert builds up a legitimate model of the framework. A quest for choices is dissected carefully.There are 3 sections in feasibility study

4.6.1 Technical Feasibility

Technical feasibility is an investigation of function performance and requirements that may influence the capacity to accomplish an adequate framework. It is every now and again the most troublesome area to evaluate at this phase of framework advancement process. During specialized examination, the investigator assesses the specialized benefits of the system idea, while simultaneously gathering extra data about performance,reliability, viability and reducability, the principle specialized issues generally raised during attainability phase of examination incorporate.Understand the different progressions related with the proposed framework before starting the task the framework must be clear about what are the advances that are to be required for the advancement of the new framework.Current technical resources are sufficient for the new system.By considering these facts this project is technically feasible.

4.6.2 Operational Feasibility

The project is profitable only if it can be converted into information systems that will satisfy the needs of the organization. In short, this crash test check whether the system will work if it is build and installed.Here are some points to evaluate the project's performance:

- If the existing system is so popular and used to the extent that people are blind to the reasons for change, there can be opposition.
- If the existing business techniques is not admissible to the user.Then users can accept changes that will make more efficient systems.
- Timely involvement decreases the likelihood of a program failure and generally improves the chances of a successful system. As the plan proposed was helpful to decrease the difficulty

4.6.3 Economical Feasibility

Cost-benefit-analysis is among the most significant data contained in a feasibility study, which is an appraisal of economic justification or a PC based framework project cost-benefit-analysis investigation portrays costs for projects advancement and loads them against tangible and intangible of system. To create system actually for the utilization it need money related advantages or on the other hand surpasses the expenses or made equivalent. The inquiries raised with the end goal of evaluating are

- Cost to conduct a full system investigation
- Cost of hardware and software for the class of application being considered
- Benefits in the form of reduced costs or few costly errors

CHAPTER 5
SYSTEM DESIGN

5.1 Applications Architecture

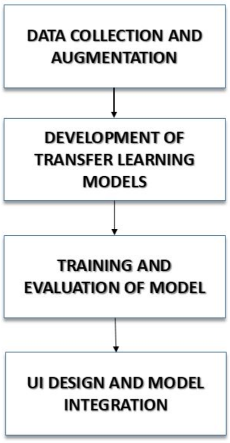

DATA COLLECTION AND AUGMENTATION
DEVELOPMENT OF TRANSFER LEARNING MODELS
TRAINING AND EVALUATION OF MODEL
UI DESIGN AND MODEL INTEGRATION

Fig. 5.1: Architecture Diagram

Fig. 4.1 is the architecture of the proposed model for detection of pneumonia from chest X-ray images. In the developed system,the users are requested to upload a clear chest X-ray image using the upload button in the user interface. After uploading , the predict button predicts the result to the user about the respective class to which the X-ray belong to , i.e , whether the patient is pneumonic or not.

The results are predicted by the trained deep learning models. The data collected are splitted into two folders for training and testing in a ratio of 8:2 . Data augmentation is done to significantly increase the diversity of data available for training models. These data are then used to train the customised deep learning models developed . Model for training is built using keras. Several Deep learning models are developed and trained to obtain the best suitable model for detection. In order to customize the model, Add preprocessing layer to the front of the model. These layers are then set to be trainable for the type of data we want. The final output layers flattened and dense using ReLu activation function.The model thus created is complied using optimiser Adam with binary-crossentropy as loss and accuracy as the metrics.Finally the model is trained with the required number of epochs and is saved. The model thus created is converted to tflite model for integrating to android UI. Inorder to predict the result , the user need to upload a chest X-ray image to test the model. When the user clicks the predict button, it will return the corresponding class to which the image belongs to.

5.2 Workflow

Implementation is supposed to have the following steps :

- Data Collection
- Data Augmentation
- Development of transfer learning models (customization)
- Train models
- Evaluation
- Choose the best model
- UI Design and model Integration
- Generate predicted results

5.3 List of Modules

Pneumonia Detection consists of mainly four modules. They are:

- Data collection and Augmentation
- Development of Transfer learning models
- Training and Evaluation of models
- UI Design and Model integration

5.3.1 Data collection and Augmentation

The Data set is taken from kaggle repository for training. Some Data are collected from hospital for testing purpose. Split collected images as 80% for training and 20% for validation. Images are grouped into 2 classes - normal and pneumonia. The training set contains 5889 labeled images.The test set contains 1749 images. All the images are resized to [224,224] Since neural networks receive inputs of the same size, all images need to be resized to a fixed size before inputting them to the CNN. Data augmentation is done to significantly increase the diversity ofdata available for training models. ImageDataGenerator class is used for Data Augmentation.

5.3.2 Development of Transfer learning models

Model for training is built using keras. After selecting the model , Add preprocessing layer to the front of the model. Here we will be using imagenet weights which is large visual database designed for use in visual object recognition. The layers in the model are set ”trainable = false” , thus not to train the existing weights of the model. The last layer in the model is flattened and then dense using ReLU Activation function. Rectified Linear Unit activation function is a linear function that will output the input directly if it is positive, otherwise, it will output zero.

Several Deep learning models are customised and developed in order to choose the best model for prediction. The Models are as follows :

5.3.2.1 DenseNet121

DenseNet is a convolutional neural network in which each layer is connected to all other layers deeper in the network; for example, the first layer is connected to the second, third, fourth, and so on, while the second layer is connected to the third, fourth, fifth, and so on. The DenseNet Architecture is made up of DenseBlocks. In the four dense blocks of the DenseNet-121, for example, there are [6,12,24,16] layers.

5.3.2.2 MobileNetV3

MobileNetV3 is a convolutional neural network that is optimised for mobile phone CPUs using a mix of hardware-aware network architecture search (NAS) and the NetAdapt algorithm, and then further improved using innovative architecture innovations.

5.3.2.3 InceptionV3

Inceptionv3 is a convolutional neural network architecture from the Inception family that uses Label Smoothing, Factorized 7 x 7 convolutions, and an auxiliary classifer to transport label information lower down the network (along with the use of batch normalisation for layers in the sidehead). The accuracy curves produced by GPU workloads with similar configurations are matched by Inception v3 TPU training runs. On v2-8, v2-128, and v2-512 settings, the model has been successfully trained. The model has attained greater than 78.1% accuracy in about 170 epochs on each of these.

5.3.2.4 ResNet152V2

A CNN architecture with hundreds or thousands of convolutional layers is known as a Residual Network (ResNet). Additional layers' efficacy was reduced by previous CNN configurations. ResNet has a large number of layers and is extremely fast. The main difference between ResNetV2 and the original (V1) is that V2 applies batch normalisation to each weight layer before applying it. ResNet has great performance in image recognition and localization tasks, demonstrating the importance of numerous visual recognition tasks. Instead of the VGG19-CNN model, ResNet152V2 is employed as a feature extraction model.

5.3.2.5 Sequential

The Sequential model is made up of a series of layers that are stacked in a linear fashion. A sequential design is the most typical ConvNet architecture. Some architectures, however, are not linear stacks. Siamese networks, for example, are two parallel neural networks with some shared layers. Here are some more instances.

5.3.2.6 VGG16

In their publication ″Very Deep Convolutional Networks for Large-Scale Image Recognition,″ K. Simonyan and A. Zisserman from the University of Oxford proposed the VGG16 convolutional neural network model. In ImageNet, a dataset of over 14 million images belonging to 1000 classes, the model obtains 92.7 percent top-5 test accuracy. It was a well-known model that was submitted to the ILSVRC-2014. It outperforms AlexNet by sequentially replacing big kernel-size filters (11 and 5 in the first and second convolutional layers, respectively) with numerous 33 kernel-size filters. VGG16 had been training for weeks on NVIDIA Titan Black GPUs.

5.3.3 Training and Evaluation of models

Model thus created is compiled. Adam is used as the optimizer with binary-crossentropy as loss and accuracy as the metrics. optimizer is a function or an algorithm that modifies each epoch's weights and minimize the loss function. Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. Finally the model is trained with the required number of epochs and is saved.

5.3.4 UI Design and Model integration

The model thus created is converted to tflite_model for integrating to android UI. TFLiteConverter in Tensorflow Library is used for conversion. Upload a chest X-ray image to test the model. When the user clicks the predict button, it will return the corresponding class to which the image belongs to.

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 System Implementation

Implementation is supposed to have the following steps :

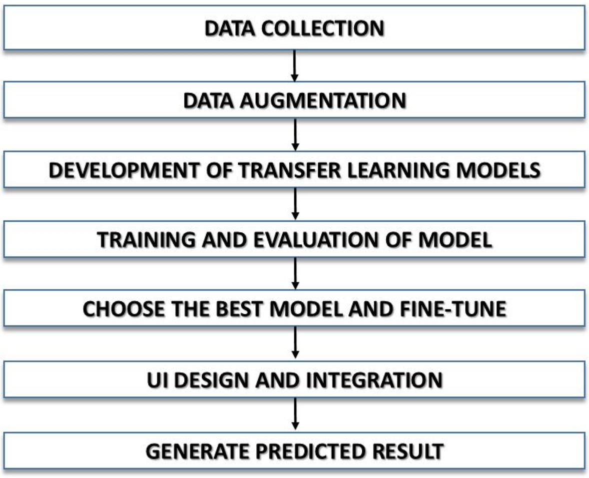

DATA COLLECTION
DATA AUGMENTATION
DEVELOPMENT OF TRANSFER LEARNING MODELS
TRAINING AND EVALUATION OF MODEL
CHOOSE THE BEST MODEL AND FINE-TUNE
UI DESIGN AND INTEGRATION
GENERATE PREDICTED RESULT

Fig. 6.1: Implementation levels

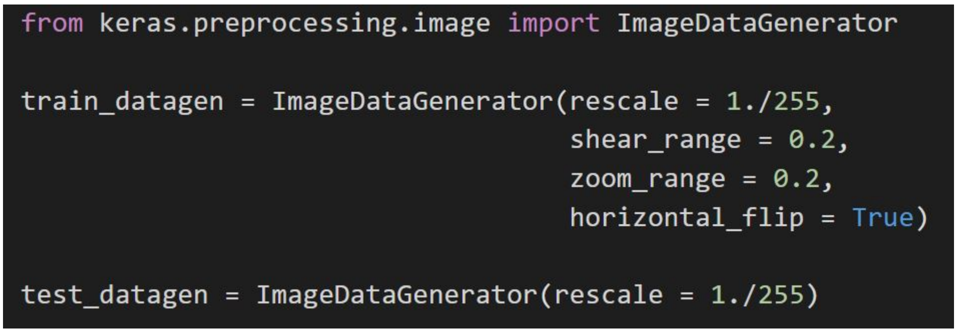
6.1.1 Data Collection

- The Data set is taken from kaggle repository for training.
- Some Data are collected from hospital for testing purpose.
- Split collected images as 80% for training and 20% for validation.
- Images are grouped into 2 classes - normal and pneumonia.
- The training set contains 5889 labeled images.
- The test set contains 1749 images.

- All the images are resized to [224,224] .
- Since neural networks receive inputs of the same size, all images need to be resized to a fixed size before inputting them to the CNN.

6.1.2 *Data Augmentation*

- Data augmentation is done to significantly increase the diversity of data available for training models.
- ImageDataGenerator class is used for Data Augmentation.



```
from keras.preprocessing.image import ImageDataGenerator

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

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

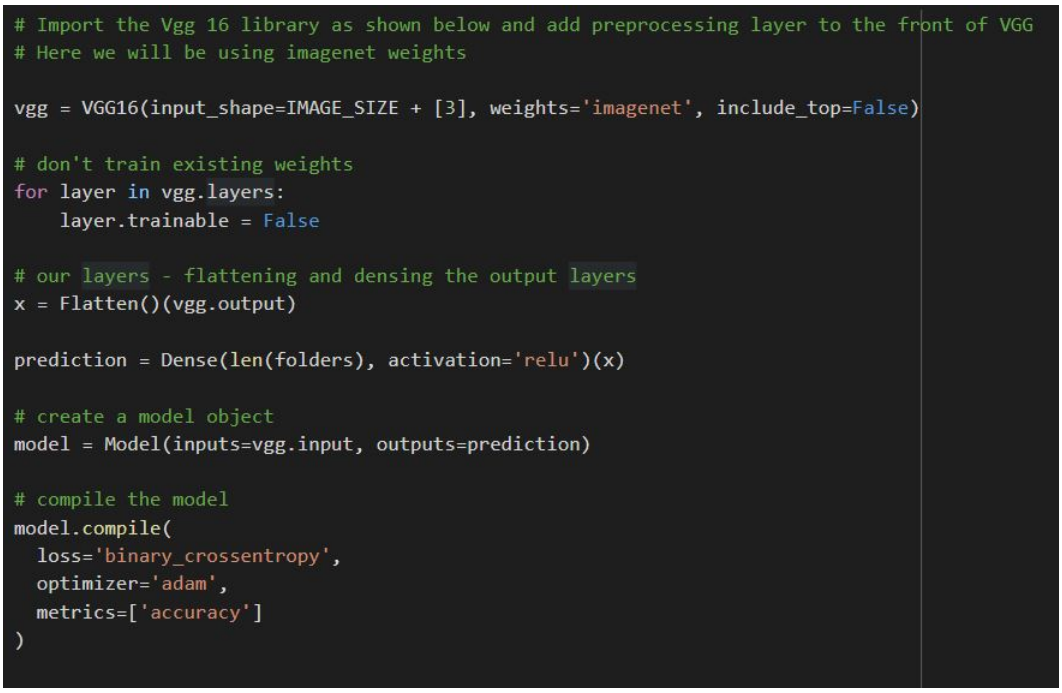
Fig. 6.2: Data Augmentation

- Rescale is a value by which we will multiply the data before any other processing.
- Our original images consist in RGB coefficients in the 0-255, but such values would be too high for our model to process(given a typical learning rate).
- so we target values between 0 and 1 instead by scaling with a 1/255. factor.

6.1.3 *Development of transfer learning models*

- Model for training is built using keras.
- After selecting the model , Add preprocessing layer to the front of the model.

- Here we will be using imagenet weights which is large visual database designed for use in visual object recognition.
- The layers in the model are set ”trainable = false” , thus not to train the existing weights of the model.
- The last layer in the model is flattened and then dense using ReLU Activation function.
- Rectified Linear Unit activation function is a linear function that will output the input directly if it is positive, otherwise, it will output zero.



```
# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG
# Here we will be using imagenet weights

vgg = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)

# don't train existing weights
for layer in vgg.layers:
    layer.trainable = False

# our layers - flattening and densing the output layers
x = Flatten()(vgg.output)

prediction = Dense(len(folders), activation='relu')(x)

# create a model object
model = Model(inputs=vgg.input, outputs=prediction)

# compile the model
model.compile(
  loss='binary_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

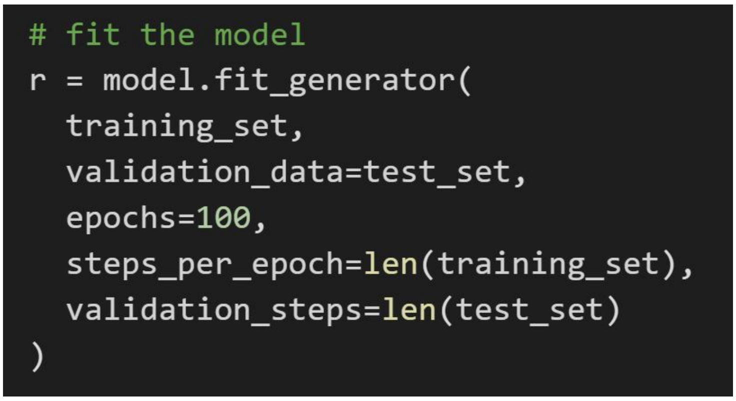
Fig. 6.3: Model Creation

6.1.4 *Training and Evaluation of models*

- Model thus created is compiled.
- Adam is used as the optimizer with binary-crossentropy as loss and accuracy as the metrics.
- optimizer is a function or an algorithm that modifies each epoch’s weights and minimize the loss function.

- Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1.

- Finally the model is trained with the required number of epochs and is saved.



```
# fit the model
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=100,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

Fig. 6.4: Model Training

6.1.5 *Choose the best model*

Accuracy of various trained models are as follows :

- DenseNet121 - 50%

- MobileNetV3 - 49%

- InceptionV3 - 50%

- ResNet152V2 - 85%

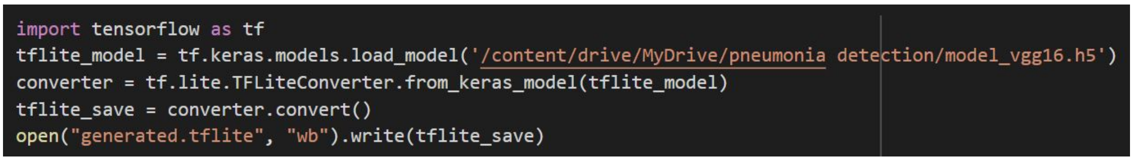
- Sequential - 88%

- VGG16 - 90%

Among these, VGG16 shows the highest accuracy . So it has been chosen to integrate with the android UI.

6.1.6 *UI Design and Integration*

- UI is designed in Android with the help of java programming language.
- The model thus created is converted to tflite_model for integrating to android UI.
- TFLiteConverter in Tensorflow Library is used for conversion



```
import tensorflow as tf
tflite_model = tf.keras.models.load_model('/content/drive/MyDrive/pneumonia detection/model_vgg16.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(tflite_model)
tflite_save = converter.convert()
open("generated.tflite", "wb").write(tflite_save)
```

Fig. 6.5: Conversion of ”model.h5” to ”tflite” model

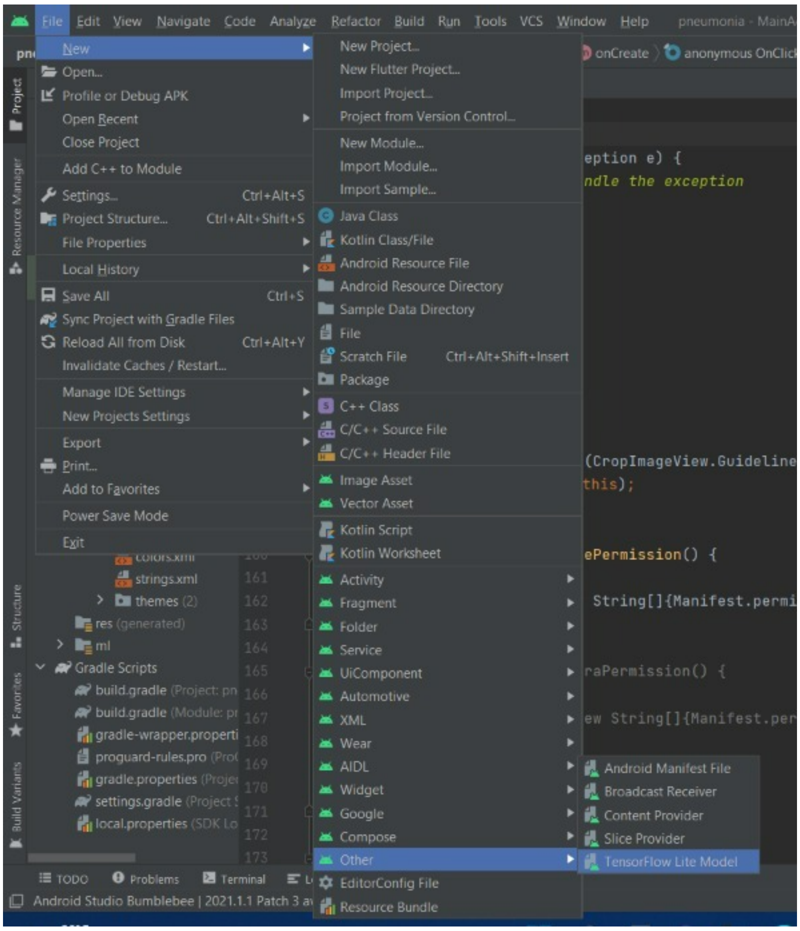


Fig. 6.6: Adding tflite model to Android

6.1.7 Predict Results

- Upload a chest X-ray image using Upload button in Android UI.
- When the user clicks the predict button, it will return the corresponding class to which the image belongs to.

6.2 Testing

Software testing is the process of comparing the application to the user′s preferences or requirements and determining whether or not the requirements are met. It is carried out in tandem with the stages of development, as each change introduced is tested multiple times to ensure that the desired output is produced. Software is tested at several levels:

6.2.1 Unit testing

Unit testing refers to the process of testing each individual unit of a module. It is used to ensure that the application runs smoothly and without errors after it has been deployed. The current system has successfully passed unit testing.

6.2.2 Integration testing

Integration testing ensures that each of the individual units tested during unit testing functions correctly when combined with other modules. It examines the combined performance result and evaluates based on expected value. If the requirement is not met, the system is adjusted. Successful Integration testing has been implemented in the current system.

6.2.3 Functionality testing

The purpose of functional testing is to determine how efficient the software is. It examines the performance in terms of time taken in order to assess the accuracy of the desired output. This is accomplished naturally through Load Testing and Stress Testing, which primarily account for the time required to complete specific tasks. The current system has passed functionality testing successfully.

6.2.4 Validation

Validation is the procedure of inspecting whether the product fulfils the client prerequisites. It is completed toward the end of the Software development life cycle. If the software matches the prerequisites, it is validated. Validation of user inputted responses is done. It checks whether an image is given in correct format. If no input is given, then it is warned by a message to add a response.

6.2.5 Verification

Verification is the process of confirming whether or not the software meets the requirements and was developed in accordance with the proper specifications and methodologies. The authentication mechanism was tested using all possible test cases.

CHAPTER 7
RESULTS AND DISCUSSION

Fig. 7.1: App Splash Screen

This is the splash screen of the app that splashes for a while when the app is opened.

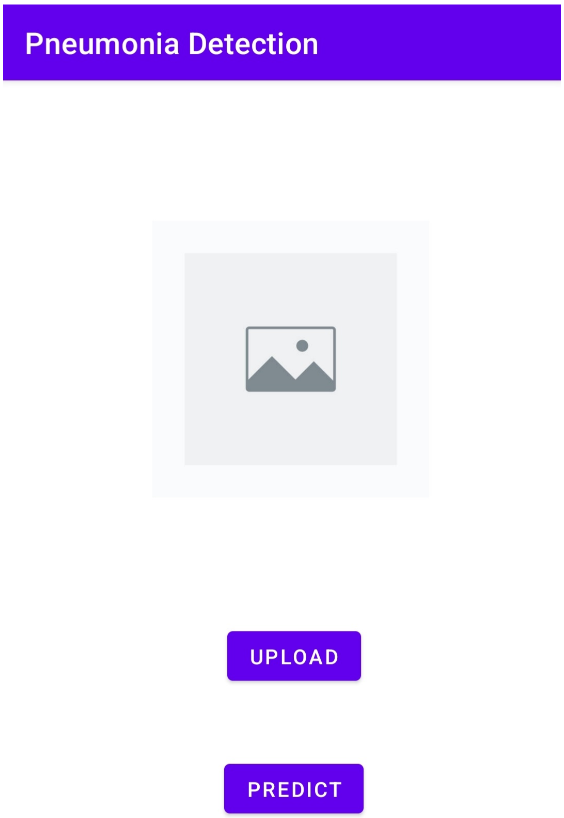


Fig. 7.2: Application Home Page

The Home page contains two buttons , namely , select and predict to choose the chest X-ray image and predict the result respectively. There is also an area in the homepage to preview the selected image.

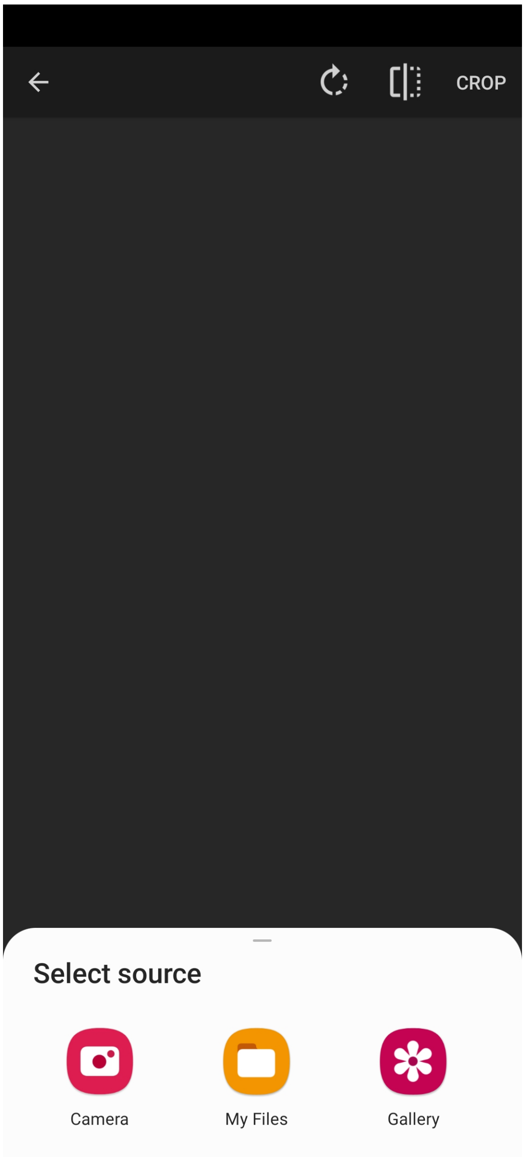
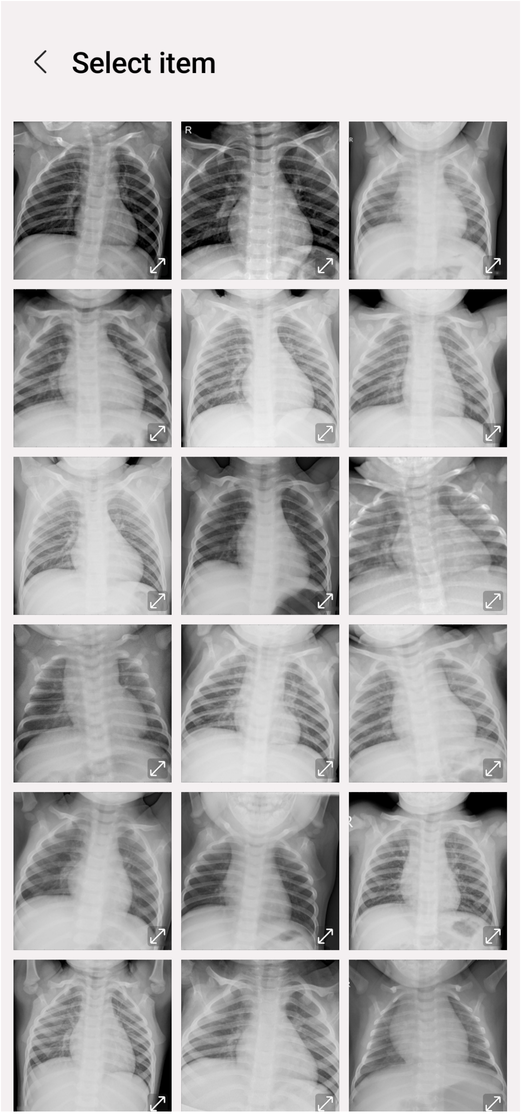


Fig. 7.3: Select sources

When the select option in the home page is clicked , different select sources will be displayed to select X-ray image.



Select item

Fig. 7.4: Select item

After choosing the select option , it will be redirected to the various images available in the selected option.

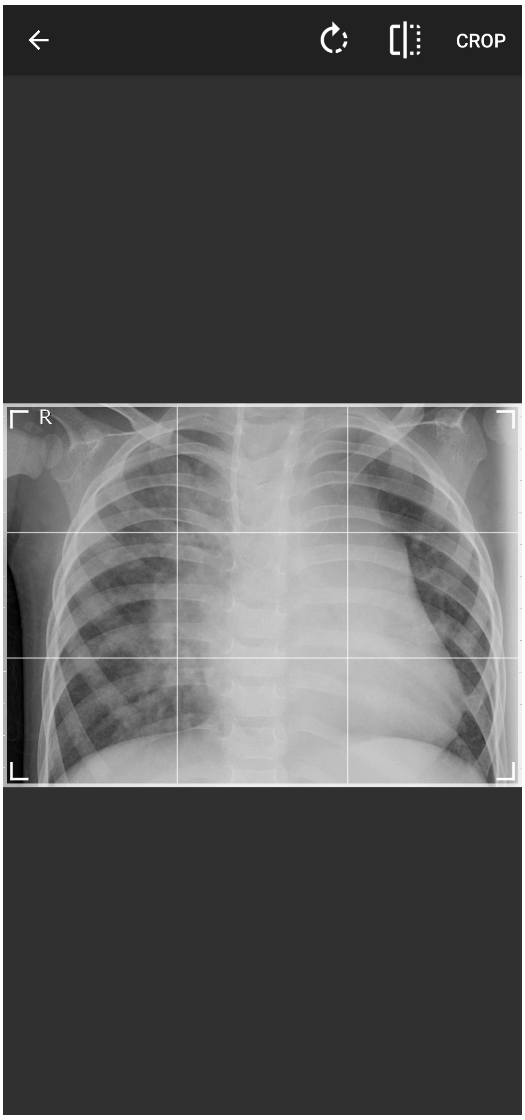
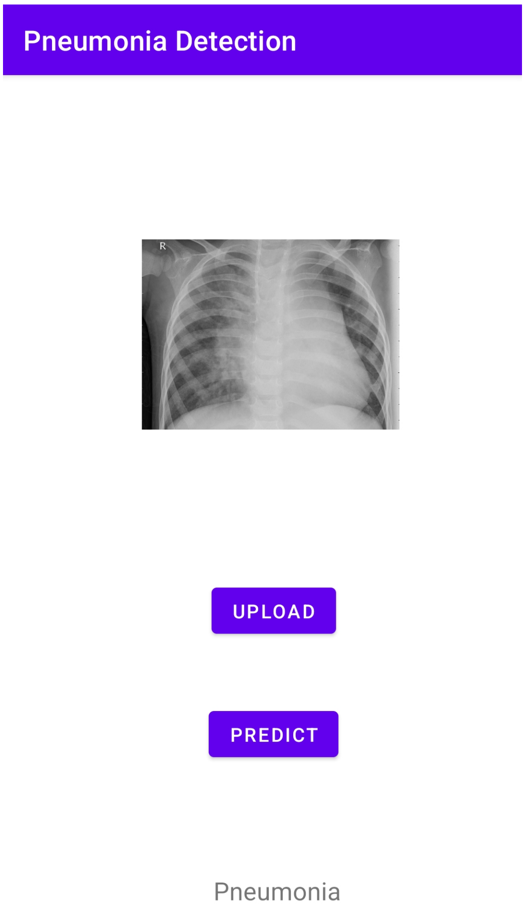


Fig. 7.5: Crop image

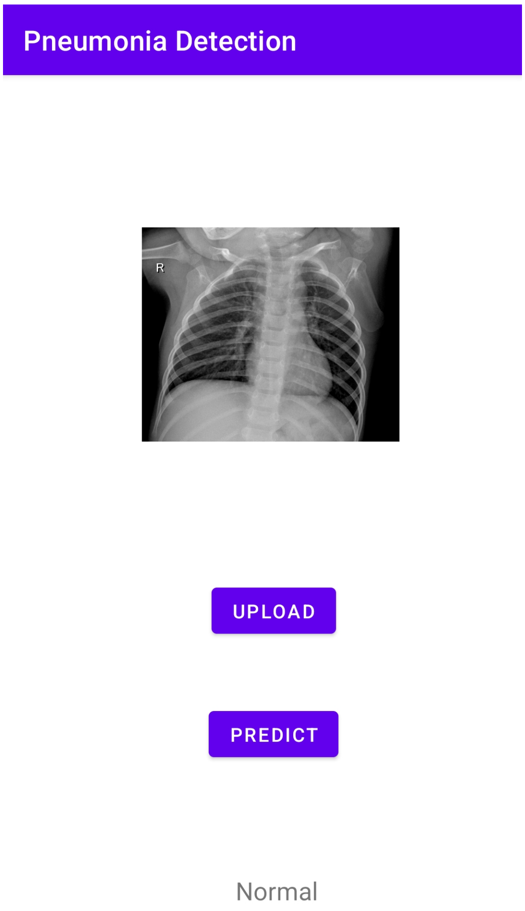
The image chosen can be cropped to the required area for prediction.



Pneumonia

Fig. 7.6: Result of Pneumonic X-ray selected

After choosing the image using upload button , the imaged will be previewed. Then the predict button displays the result class to which the X-ray belongs to. Here the image chosen is of a pneumonic patient and the displayed result is also pneumonia.



Normal

Fig. 7.7: Result of Normal X-ray selected

After choosing the image using upload button , the imaged will be previewed. Then the predict button displays the result class to which the X-ray belongs to. Here the image chosen is of a normal patient and the displayed result is also normal.

7.1 Evaluation Measures

Evaluation measures helps to understand how well the model is working . It helps to understand the efficiency of our project. The evaluation measures that I use to check the efficiency of my project is accuracy. Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset.

7.1.1 Accuracy

Accuracy refers to the closeness of the measured value to a standard or true value. It refers to the level of agreement between the actual measurement and the absolute measurement. It is the ability of a measurement to match the actual value of the quantity being measured. It represents how closely the results agree with the standard value. The following equation shows how to obtain the value of accuracy

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

where correct predictions represent the predictions or outputs that are correct. An output is more accurate if it is close to the desired output. That means, as the value of accuracy increases, the output become more reliable. In this project ,the model that predicts pneumonia has an accuracy 90%

CHAPTER 8
CONCLUSION

An android application using a deep transfer learning model is built that can detect pneumonia from digital chest X-ray images with an accuracy above 90. With this model, it is possible for early diagnosis of Pneumonia, that helps to save the patients from being severe. It helps to Minimise human intervention in detecting pneumonia. The result shows that the proposed method has good performance that helps the trained radiologist for early detection of pneumonia.

In the developed system,the users are requested to upload a clear chest X-ray image using the upload button in the user interface. After uploading, the predict button predicts the result to the user about the respective class to which the X-ray belong to , i.e , whether the patient is pneumonic or not. Model for training is built using keras. Several Deep learning models are developed and trained to obtain the best suitable model for detection. In order to customize the model.Finally the model is trained with the required number of epochs and is saved. VGG16 model is chosed for detection after training several models. The model thus created is converted to tflite model for integrating to android UI.

CHAPTER 9

SCOPE FOR THE FUTURE ENHANCEMENT

In the future, this work could be extended to detect and classify X-ray images consisting of lung cancer and pneumonia. This work could also be extended to explore more accurate classification architectures to diagnose two types of pneumonia, viruses, and bacteria. The future works can also involve developing an algorithm which can localize the parts of the lung affected by pneumonia.

In the future, it is possible to investigate techniques such as contrast enhancement of the images or other pre-processing steps to improve the image quality. It can also be considered using segmentation of the lung image before classification to enable the models to achieve improved feature extraction,

Thus in the future, it would be interesting to see approaches in which the weights corresponding to different models can be estimated more efficiently and a model that takes into account the patient's history while making predictions.

BIBLIOGRAPHY

[1] Hashmi, Mohammad Farukh et al. *"Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning."* Diagnostics (Basel, Switzerland)vol. 10,6 417. 19 Jun. 2020, doi:10.3390/diagnostics10060417

[2] S. Arunmozhi, V. Rajinikanth and M. P. Rajakumar, *"Deep-Learning based Automated Detection of Pneumonia in Chest Radiographs,"* 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021, pp. 1-4, doi: 10.1109/ICSCAN53069.2021.9526482.

[3] K. More, P. Jawale, S. Bhattad and J. Upadhyay, *"Pneumonia Detection using Deep Learning,"* 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021, pp. 1-5, doi: 10.1109/SMARTGENCON51891.2021.9645844.

[4] K. S. and D. Radha, *"Analysis of COVID-19 and Pneumonia Detection in Chest X-Ray Images using Deep Learning,"* 2021 International Conference on Communication, Control and Information Sciences (ICCISc), 2021, pp. 1-6, doi: 10.1109/ICCISc52257.2021.9484888.

[5] https://www.geeksforgeeks.org/python-image-classification-using-keras/

[6] https://stackabuse.com/image-recognition-in-python-with-tensorflow-and-keras/

[7] https://keras.io/api/applications/

[8] https://www.w3schools.com/

[9] https://www.kaggle.com/