ABSTRACT

A chest radiograph, called a chest X-ray (CXR), or chest film, is a projection radiograph of the chest used to diagnose conditions affecting the chest, its contents, and nearby structures. The CXR dataset that is used in this system consists of details about 15 diseases. This project is implemented on the VinDr-CXR dataset and Shenzhen dataset, in which the former comprises of 18,000 X-Ray images with 14 different disease labels which was released in the year 2020 and the latter comprises of 660 X-Ray images for Tuberculosis classification. This system is defined to automatically detect 15 different types of thoracic lung diseases namely Aortic enlargement, Atelectasis, Calcification, Cardiomegaly, Consolidation, ILD(Interstitial Lung Disease), Infiltration, Lung Opacity, Nodule/Mass, Other lesion, Pleural effusion, Pleural thickening, Pneumothorax, Pulmonary fibrosis and Pulmonary Tuberculosis from CXR's by applying traditional deep convolutional neural network models. This system uses MobileNet, a 53- layer convolutional neural network that inputs a chest X-ray image and outputs the probability of various diseases. Here, MobileNet is extended to detect all 15 diseases in VinDr-CXR dataset and Shenzhen dataset, and achieve state of the art results on all 15 diseases. This system also proposes an explainable method aimed to automatically detect the areas of interest in the chest X-ray, symptomatic of these thoracic lung diseases. LIME algorithm is used in this system to explain the model's predictions. This automated system could accurately identify and localize findings on chest radiographs which would relieve the stress of busy doctors while also providing patients with a more accurate diagnosis which can be considered as a valuable second opinion for radiologists.

KEYWORDS:

Chest X-Rays (CXR's), Thoracic diseases, Deep convolutional neural network, MobileNet, LIME algorithm.

CHAPTER 1

INTRODUCTION

1.1 GENERAL

According to a survey taken by World Health Organization (WHO), over 2 million premature deaths occur due to chronic lung diseases such as lung cancer, bronchitis, tuberculosis, pneumonia, cardiomegaly, fibrosis, effusion, emphysema and other lethal clinical symptoms, and approximately 70% of the people died due to their negligence in detecting these diseases in early stages. If these diseases were detected on right time, with help of proper precautions & treatments, it can be stopped from becoming fatal. Therefore, early detection of lung diseases has become more important than ever. Machine learning and deep learning can play a vital role for this purpose. This proposed system can provide doctors and other researchers a direction for detecting lung disease with the help of deep learning methodology. A large number of lung/chest radiographic images are used as a dataset. The system presented herein can also assist to detect diseases more accurately, which can protect numerous vulnerable people and decrease the disease rate.

Although both computed tomography (CT) scans and X-Rays can be used for detecting lung disease, CT scans are not generally preferred due to it's high cost and also access to CT scans are very limited in rural areas when compared to x-rays. Hence, this system uses chest x-rays or chest radiographs to detect these lung diseases.

There are more than 200 types of thorax diseases, because of these diseases the blood circulation in the body is affected and the breathing capability gets reduced. Although it is very difficult to recognize these diseases with normal symptom in preliminary stages, it is also a difficult task to detect these diseases from low feature, smooth and low-resolution images. To resolve this problem new and large datasets called VinDr-CXR dataset and Shenzhen dataset are used.

1.2 OBJECTIVE

This system is defined to automatically detect 15 different types of thoracic lung diseases namely Aortic enlargement, Atelectasis, Calcification, Cardiomegaly, Consolidation, ILD(Interstitial Lung Disease), Infiltration, Lung Opacity, Nodule/Mass, Other lesion, Pleural effusion, Pleural thickening, Pneumothorax, Pulmonary fibrosis and Pulmonary Tuberculosis from chest x-rays by applying traditional deep convolutional neural network models. This system uses MobileNet, a 53- layer convolutional neural network that inputs a chest X-ray image and outputs the probability of various diseases. Here, MobileNet is extended to detect all 15 diseases in VinDr-CXR dataset and Shenzhen dataset, and achieve state of the art results on all 15 diseases.

1.3 MOBILENET

MobileNet-v2 is a convolutional neural network that is 53 layers deep. This model was developed at Google, pre-trained on the ImageNet dataset with 1.4M images. This model can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

CHAPTER 2

LITERATURE SURVEY

Automatic Lung Cancer Prediction from Chest X-ray Images Using the Deep Learning Approach [1]

In this study, the 121-layer convolutional neural network also known as DenseNet121 by G. Huang et. al., along with the transfer learning scheme was explored as a means to classify lung cancer using chest X-ray images. The proposed model also provides a heat map for identifying the location of the lung nodule. These findings are promising for further development of chest x-ray-based lung cancer diagnosis using the deep learning approach.

Advantages:

- The proposed training strategy performed better than the normal transfer learning method, which resulted in higher mean accuracy and mean sensitivity

Disadvantages:

- CAMs illustrated by Model B do not show the accurate position of lung cancer and often show that the model used too large an image area to classify lung cancer.

Deep Convolution al Neural Networks for Chest Diseases Detection [2]

. In this paper, they demonstrate the feasibility of classifying the chest pathologies in chest X-rays using conventional and deep learning approaches. In this paper, convolutional neural networks (CNNs) are presented for the diagnosis of chest diseases. The architecture of CNN and its design principle are presented. For comparative purpose, backpropagation neural networks (BPNNs) with supervised

learning, competitive neural networks (CpNNs) with unsupervised learning are also constructed for diagnosis chest diseases. All the considered networks CNN, BPNN, and CpNN are trained and tested on the same chest X-ray database, and the performance of each network is discussed. Comparative results in terms of accuracy, error rate, and training time between the networks are presented.

Advantages:

- The obtained results have demonstrated the high recognition rates of the proposed CNN.

Disadvantages:

- Computation time and the number of iterations were roughly higher.

Hybrid deep learning for detecting lung diseases from X-ray images [3]

In this paper, they propose a new hybrid deep learning framework by combining VGG, data augmentation and spatial transformer network (STN) with CNN. This new hybrid method is termed here as VGG Data STN with CNN (VDSNet). As implementation tools, Jupyter Notebook, Tensorflow, and Keras are used. The new model is applied to NIH chest X-ray image dataset collected from Kaggle repository.

Advantages:

- Their hybrid approach efficiently detects the inflammatory area in chest X-ray images.

Disadvantages:

- It has low validation Accuracy.

Early Prediction of Lung Diseases [4]

The aim of the paper is to detect and diagnose the lung diseases as early as possible which will help the doctor to save the patient's life. This paper describes how lung diseases was predicted and controlled, using Machine Learning. In this paper, they develop a Machine Learning based decision support system that contributes in diagnosing the diseases.

Advantages:

- CNN algorithm added additional benefits to predict the lung diseases in advance with better results.

Disadvantages:

- Data is not standardized.

Detection of Lung Diseases using Deep Learning [5]

This paper presents an effective way for expert diagnosis of lung diseases using Deep Learning. It focuses on creating a system for assistance of Radiologists in detection of lung diseases. This will especially benefit rural areas where radiologists aren't easily available. Their system connects radiology labs with the radiologists who can diagnose faster and better with their model.

Advantages:

- This model has been deployed as an application in cloud, hence can be used by anyone from anywhere.

Disadvantages:

- The approach for localization of lung diseases in Chest Xray images improvements can be made in algorithms to generate the activation maps.

Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks [6]

In this paper, they have explored DCNN based abnormality detection in frontal chest X Rays. They have also performed localization of features responsible for classification decision.

They have also found ensemble models to improve classification significantly compared to single model when only DCNN models are used. They have combined DCNN models with rule based models which degraded the accuracy.

Advantages:

- Shallow features or earlier layers consistently provide higher detection accuracy compared to deep features.

Disadvantages:

- The localization fails for pointed features like lung nodule or bone fracture.

Chest Abnormality Detection from X-ray using Deep Learning [7]

This paper is based on the concept of deep learning to find abnormalities in chest x-rays. In this paper, they are using convolutional neural network to train the neurons which will help in finding the input x-rays as normal & abnormal images. They also used digital image processing techniques and expert radiologist advice to create a pipeline that will be applied to neural network which will result in finding a particular disease.

Advantages:

- In this paper, they have decided to use MATLAB which is a high performance programming language and it also provides higher graphical capabilities.

6

Disadvantages:

- The system becomes complicated when a data set is large in number and also becomes time consuming.

XRAY AI: Lung Disease Prediction Using Machine Learning [8]

In this paper, deep convolutional neural network is used to recognize and locate the common disease patterns in chest X-Ray images and they have tried to integrate their automated detection system with normal hospital management system. They have performed supervised learning technique with the NIH Chest X-Ray dataset which comprises of 112,120X-ray images with disease labels from 30,000 unique patients.

Advantages:

- The proposed system can be utilized as a mean for verification for the doctors.

Disadvantages:

- The accuracy of the system can be enhanced by attaching attributes like rate of smoking, family history etc to the data list entry.

Detecting Lung Abnormalities From X-rays Using an Improved SSL Algorithm [9]

In this paper, they propose a new semi-supervised learning algorithm for the classification of lung abnormalities from X-rays based on an ensemble philosophy. The efficacy of the presented algorithm is demonstrated by numerical experiments, illustrating that reliable prediction models could be developed by incorporating ensemble methodologies in the semi-supervised framework.

Advantages:

- The proposed algorithm combines the individual predictions of

efficient self-labeled algorithms utilizing a majority voting methodology.

Disadvantages:

- Comparison of proposed EnSL with other algorithm such as generative mixture models, transductive SVMs, as well as graph-based methods, hence questioning it's relative efficiency.

Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning [10]

In this paper, 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. A novel approach based on a weighted classifier is introduced, which combines the weighted predictions from the state-of-the-art deep learning models such as ResNet18, Xception, InceptionV3, DenseNet121, and MobileNetV3 in an optimal way. Here a supervised learning approach is performed, in which the network predicts the result based on the quality of the dataset used.

Transfer learning is used to fine-tune the deep learning models to obtain higher training and validation accuracy.

Advantages:

- The proposed weighted classifier is able to outperform all the individual models and partial data augmentation techniques are employed to increase the training dataset in a balanced way.

Disadvantages:

- One of the limitations of this approach was the scarcity of available data. Another limitation was that the results of the deep learning models could not be properly explained.

A Lung Disease Classification Based on Feature Fusion Convolutional Neural Network with X-ray Image Enhancement [11]

In this paper, they present a feature fusion convolutional neural network (CNN) model to detect pneumothorax from chest X-ray images. Firstly, the preprocessed image samples are enhanced by two methods. Then, a feature fusion CNN model is introduced to combine the Gabor features with the enhanced information extracted from the images and implement the final classification.

Advantages:
- Comprehensive qualitative and quantitative experiments demonstrate that their proposed model achieve better results in multi-angle views.

Disadvantages:
- Diagnosis of the lung disease can be difficult. A chest Xray images is usually used to confirm its presence. The symptoms of these diseases showed in this kind of medical image can be vague and inconclusive, especially in most cases appearing similar characteristics that quite smaller for visual detection, such as shade, tissue swelling, or vessels gathering, which makes diagnosis hard and burdensome.

Pneumonia Detection Using CNN based Feature Extraction [12]

In this paper, they appraise the functionality of pre-trained CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. They analytically determine the optimal CNN model for the purpose. Statistical results obtained demonstrates that pre-trained CNN models employed along with supervised classifier algorithms can be very beneficial in analyzing chest X-ray images, specifically to detect Pneumonia.

Advantages:
- They showed that performing hyper parameter optimization in the

classification stage ameliorated the model performance.

Disadvantages:
- There is no history of the associated patient considered in our evaluation model. The model exercises a lot of convolutional layers, the model need very high computational power otherwise it'll eat up a lot of time in computations.

Diagnosis of Chest Diseases in X-Ray images using Deep Convolutional Neural Network [13]

In this paper, they proposed and evaluated a deep convolutional neural network, designed for classifying the Chest Diseases. The proposed model consists of Convolutional layers, ReLU Activations, Pooling layer, and Fully connected layer, Last full connected layer which consists of fifteen output units. Each output unit will predict the probability of one of the fifteen diseases. The proposed technique is best suited for classifying the multiclass medical images for different thorax diseases.

Advantages:
- The proposed architecture also takes care of the bias-variance problem and the use of early stopping technique reduced the loss as well as the over-fitting problem.
- This model also has the capacity to offer the insight feature of the convolutional neural network.

Disadvantages:
- Performance of execution is slow.

Development of a Deep Learning Model for Chest X-Ray Screening [14]

In this paper, an AI trained chest X-ray abnormal interpretation model by using DesNet121 neural network was developed and deep neural network shows the potential of accountable methods to help lung classifications for normal and abnormal screening in clinics. In this study they have collected 365,892 chest X-ray images and clinical diagnosis reports through retrospective analysis, and compared five different input image sizes and images that generated by clinical labeling pre-processing in the classification model building and testing.

Advantages:

- They have developed their own deep neural network classification model, which has a good performance in interpreting the abnormality of chest X-ray films.

Disadvantages:

- There was an obvious conflict between the clinical report and X-ray image: no finding or normal in the report but abnormal chest X-ray image, patient history was reviewed, as in the former literature, it has been proved that history of patients would affect the accuracy of radiologist interpret chest X-ray.

Deep convolutional neural network based medical image classification for disease diagnosis [15]

This paper researches how to apply the convolutional neural network (CNN) based algorithm on a chest X-ray dataset to classify pneumonia. Three techniques are evaluated through experiments. These are linear support vector machine classifier with local rotation and orientation free features, transfer learning on two convolutional neural network models: Visual Geometry Group i.e., VGG16 and InceptionV3, and a capsule network training from scratch.

Advantages:

- With the help of transfer learning, they have retrained that specific features on a new target dataset is essential to improve performance.

Disadvantages:

- Visualization needs to be added to improve the understanding and explanation of the results of the CNNbased system, because those are essential for the adoption of a CNNbased system in real clinical applications.

2.1. PROPOSED SYSTEM

In this project, a Deep Convolutional Neural Network model is built for detecting the thoracic lung diseases present in a given Chest X-ray Image. In this project, MobileNet is used as the base model and by applying transfer learning on base model, a modified model is developed. The model will now trained on a large dataset that contains 15,662 Chest X-Ray images. After sufficient number of training and testing, the modified MobileNet model, will classify the given Chest X-Ray images into any one of the 16 pre-defined classes (15 diseases and "no findings" class). If the given Chest X-Ray image is identified as a diseased one, then by applying LIME algorithm, localization will be done in order to explain the model's detection.

2.2. ADVANTAGES OF THE PROPOSED SYSTEM

- This model can be considered as a valuable second opinion for radiologists.

- In this model, uses the latest dataset, hence minimizing the errors in classification.

- LIME algorithm is used to explain the model's predictions.

- This system accurately identifies and localizes findings on chest radiographs.

CHAPTER 3

SYSTEM DESIGN

3.1 OVERVIEW

System Design deals with the various Unified Modeling Language (UML) diagrams for the implementation of projects. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished products.

3.1.1 ARCHITECTURE DIAGRAM
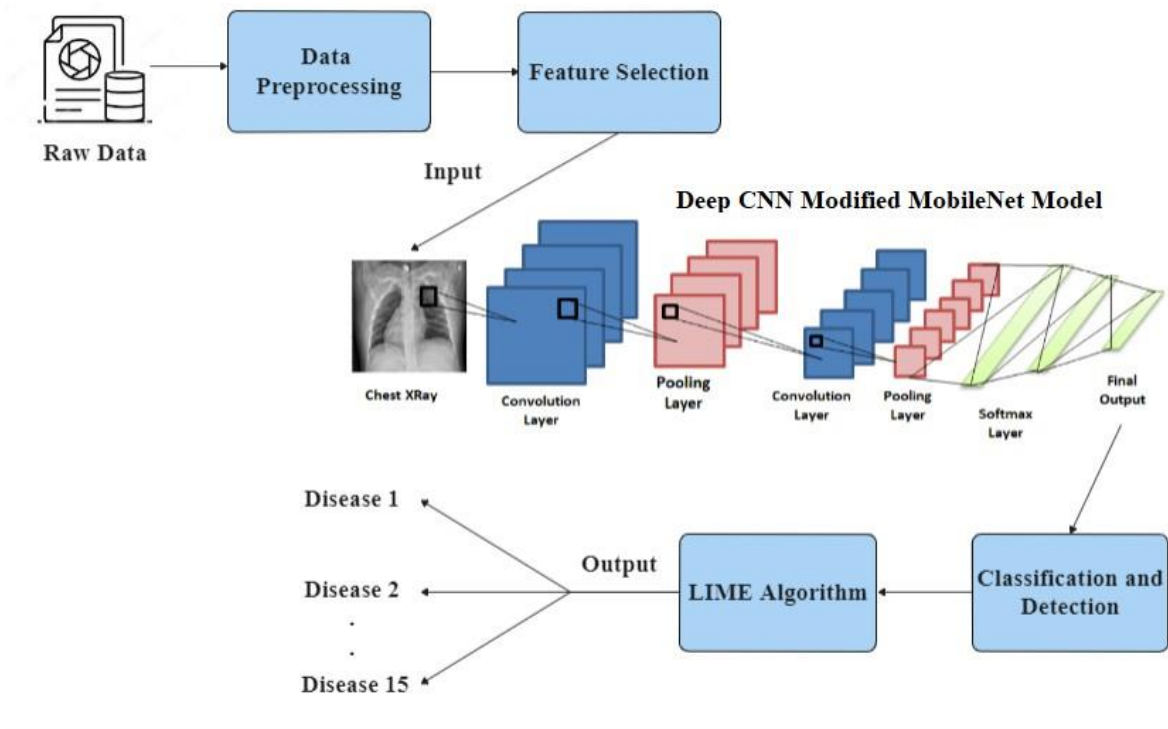
Fig: 3.1.1 Architecture Diagram

The Architecture Diagram shows how an image is taken as input and this data is preprocessed and by performing feature selection this new data is now given as input to the Modified CNN MobileNet model. Output obtained from the modified MobileNet model predicts and classifies the diseases. After detecting the disease LIME algorithm is applied to localize or highlight the critical areas.

3.1.2 DEVELOPMENT ENVIRONMENT

HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should, therefore, be a complete and consistent specification of the whole system. They are used by the software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

• Processor : Intel core i5 8th Gen or higher

• RAM : 8GB (Minimum)

• Hard disk : 220 GB of free space (Minimum)

 • Monitor : 15.6"

• Processor speed : Minimum 1.8 GHz

• System Type : 64-bit Operating System, x64- based processor

• Graphics Card : NVIDIA GEFORCE or RADEON (Minimum 650 GPU)

SOFTWARE REQUIREMENTS:

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's progress throughout the development activity.

• OS : Windows 10 or MAC

• IDE : Python IDLE 3.8

• TensorFlow : Version 2.4.1

• Keras : Version 2.4.3

• Numpy : Version 1.19.5

• Pandas : Version 1.2.2

• Scikit-learn: Version 0.24.1

• Matplotlib : Version 3.3.4

• OpenCV: Version 4.5.1.48

3.1.3 DESIGN OF THE ENTIRE SYSTEM

3.1.3.1 USE CASE DIAGRAM
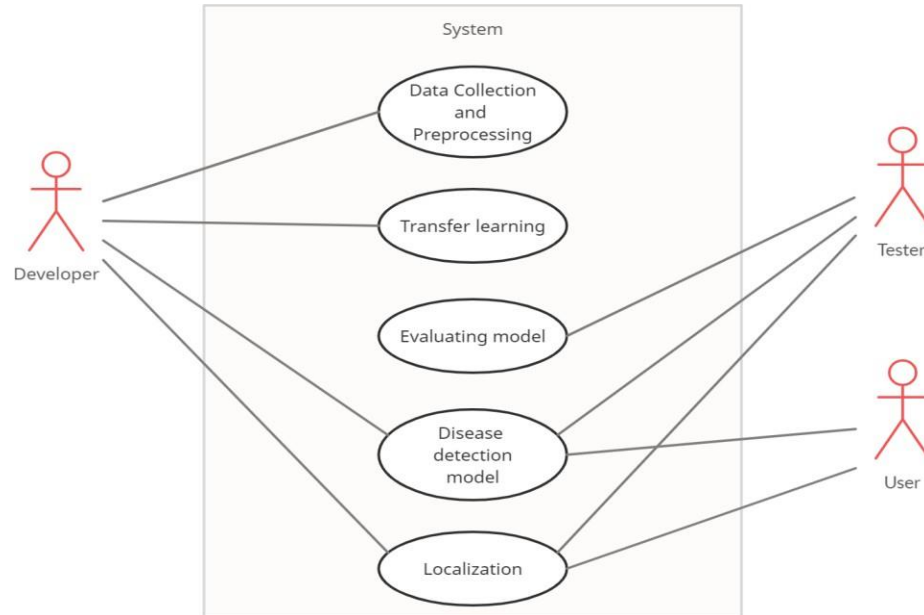
Fig 3.1.3.1 Use case diagram

There are three actors namely Developer, Tester and User. The developer collects
data, preprocesses it and performs transfer learning. The tester evaluates the models

output and finally the user can detect the disease along with localization of critical regions.

3.1.3.2 ACTIVITY DIAGRAM
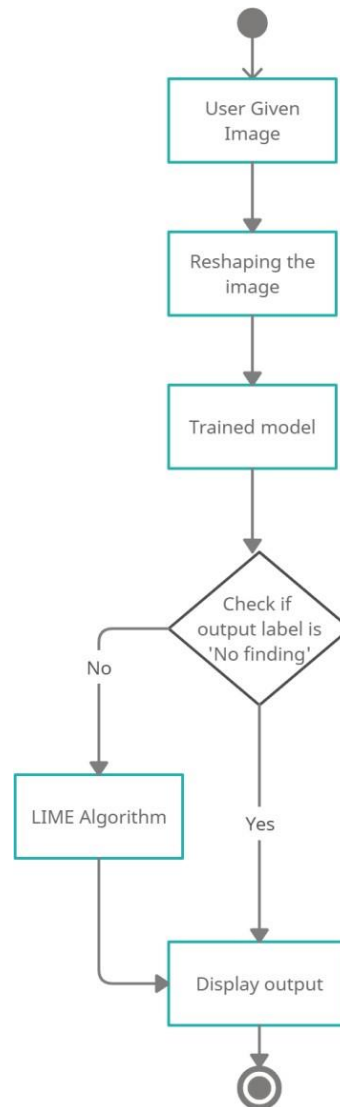
Fig 3.1.3.2 Activity diagram

Initially the input is taken as an image from the user. This image is now reshaped according to the preferred format. This reshaped image is now passed on to the trained model for predicting the output. If the output label is 'No finding' then

the output is displayed as 'No finding' else LIME algorithm is applied on the output to highlight the critical region and respective output is finally displayed.

3.1.3.3 SEQUENCE DIAGRAM

The participants are Developer, Tester and User. The object is System. A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. Sequence diagrams depict workflow or activity over time using Messages passed from element to element. The sequence of message is initially the developer does data collection and preprocessing, then he/she applies transfer learning using MobileNet model. Now this modified model is used for training. The system sends the output class labels for evaluation to the tester which is finally accessed by the end user (Classified diseases).
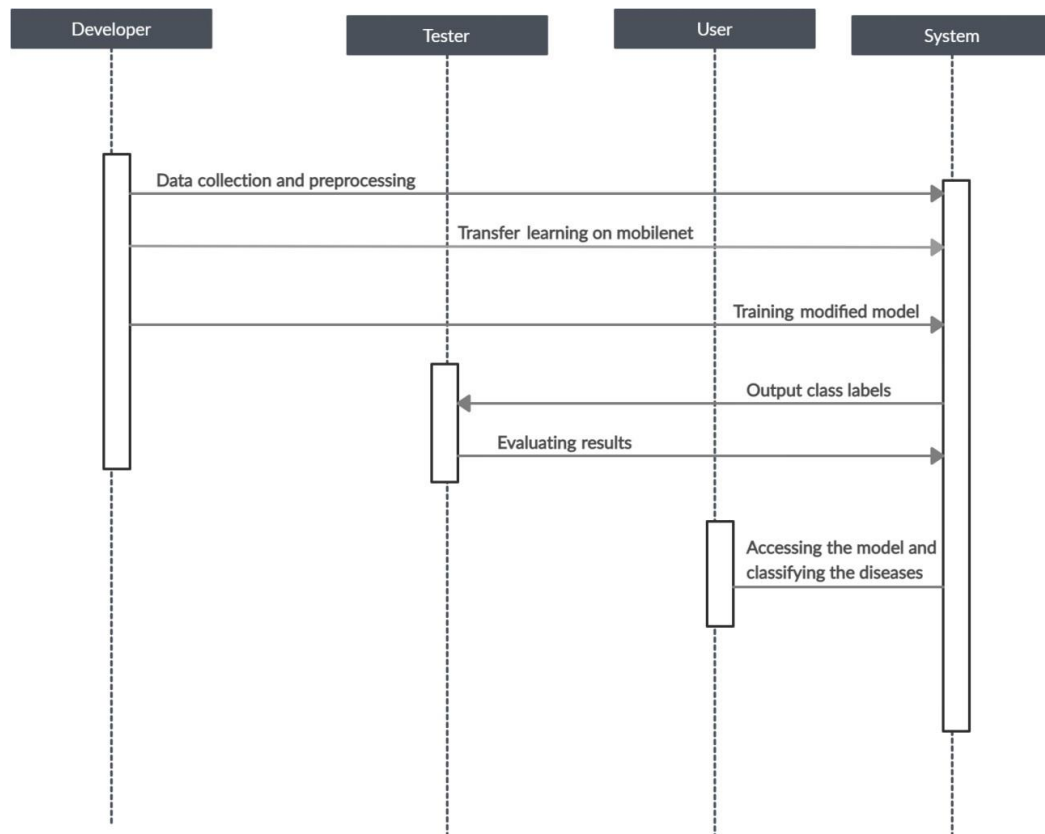
Fig 3.1.3.3 Sequence diagram

CHAPTER 4

PROJECT DESCRIPTION

4.1 GENERAL

In this project, a deep convolutional neural network model is built for detecting the thoracic lung diseases present in a given chest x-ray image. MobileNet is used as base model and by applying transfer learning on base model, the modified model was developed.

A chest x-ray image will be given as input to the model. The given image will be reshaped to required dimensions, and then be represented 2-Dimensional matrix, so that it can be passed as input to CNN model. CNN's are basically used for image classifications and they will classify images into any one of the pre- defined classes. It scans images from top to bottom as well as left to right in order to get all the important features and combines all of them.

Now all the extracted features will be used by the modified MobileNet model, to classify the given chest x-ray images into any one of the 16 pre-defined classes (15 diseases and "no findings" class). If the given Chest X-Ray image is identified as a diseased one, then by applying LIME algorithm, localization will be done in order to explain the model's detection.

4.2 METHODOLOGIES

4.2.1 MODLUES

1. DATASET PREPROCESSING AND FEATURE SELECTION.
2. TRANSFER LEARNING ON MOBILENET MODEL.
3. TRAINING MODIFIED MODEL.
4. TESTING AND VALIDATING.
5. IMPLEMENING EXPLAINABLE AI USING LIME ALGORITHM.

4.2.2 MODULES DESCRIPTION

1. DATASET PREPROCESSING AND FEATURE SELECTION

VinDr-CXR Dataset has been taken from Kaggle which comprises of 18,000 postero-anterior (PA) CXR scans in DICOM format, which were de-identified to protect patient privacy. All images were labeled by a panel of experienced radiologists for the presence of 14 critical radiographic findings as: Aortic enlargement, Atelectasis, Calcification, Cardiomegaly, Consolidation, ILD, Infiltration, Lung Opacity, Nodule/Mass, Other lesion, Pleural effusion, Pleural thickening, Pneumothorax, Pulmonary fibrosis. This dataset consists of 8 columns namely image_id, class_name, class_id, rad_id, x_min, y_min, x_max, y_max and 15,000 rows. Another dataset, Shenzhen Dataset has been taken from National Library of Medicine which comprises of 662 X-Ray images for Tuberculosis classification.

Dataset Preprocessing

The first step in Dataset Preprocessing is converting the images from DICOM format to JPEG format. The next step is removal or fixing of missing values in the .csv file. The next would be addition of the Shenzhen Data into the .csv file. Finally all the unnecessary features would be removed.

2. TRANSFER LEARNING ON MOBILENET

MobileNet-v2 is a convolutional neural network that is 53 layers deep. This model was developed at Google, pre-trained on the ImageNet dataset with 1.4M images. This model can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. Transfer learning, used in deep learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve

generalization about another. In this module, the last layer of MobileNet will modified to classify the input X-Ray images into, either 15 diseases or "Normal" class.

3. TRAINING MODIFIED MODEL

The Dataset with more than 15,000 Chest X-Ray Images is split into train and test data randomly. The train data will consists of 12,000 images and the test data will contain the remaining 3,000 images. The modified MobileNet model will classify the images into 16 classes (15 diseases and "Normal"). The Softmax activation function will be used in training the model.

4. TESTING AND VALIDATING

3,000 images of test data is used in testing the proposed model's performance. The training and testing accuracy are plotted for examining the model performance. Accuracy and Confusion Matrix will be calculated to analyze the performance.

5. IMPLEMENTING EXPLAINABLE AI USING LIME ALGORITHM
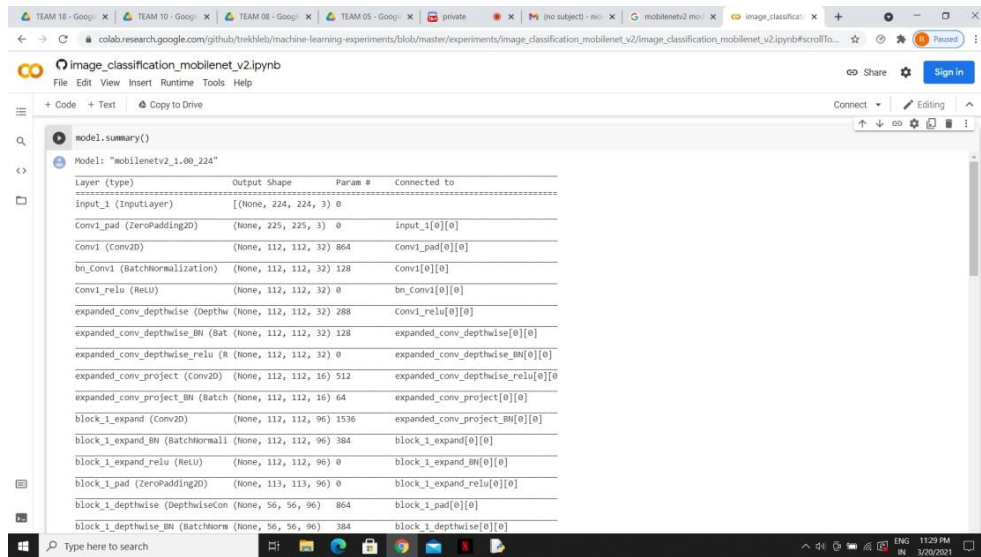
Local Interpretable Model-Agnostic Explanations LIME) is able to explain any black box classifier, with two or more classes. LIME assumes a black box machine learning model and investigates the relationship between input and output, represented by the model. LIME works by making alterations on different features on a particular input and seeing which of those alterations make the biggest difference to the output classification, hence highlighting the most relevant features.

CHAPTER 5

IMPLEMENTATION AND RESULT DISCUSSION
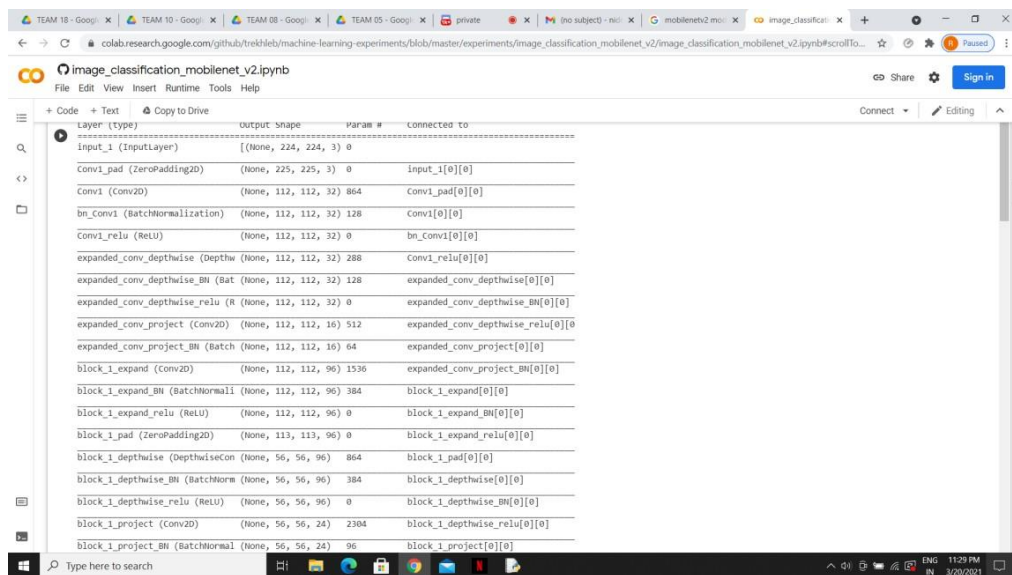
SCREENSHOTS

Fig 5.1& 5.2 shows the summary of the original MobileNet model and Fig 5.3 shows the summary of modified MobileNet model.

Fig 5.1 Summary of the original model – part1

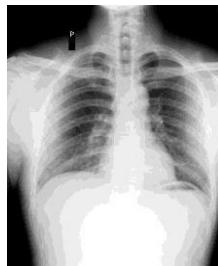**Fig 5.2 Summary of the original model – part2**

```
Python 3.8.5 Shell                                          —    □    ✕

File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM ^
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\prith\Downloads\mobilenet.py ===============
Model: "sequential"
_____
Layer (type)                 Output Shape              Param #
=================================================================
keras_layer (KerasLayer)     (None, 1280)              2257984
_____
dense (Dense)                (None, 16)                20496
=================================================================
Total params: 2,278,480
Trainable params: 20,496
Non-trainable params: 2,257,984
_____
>>>
```

Fig 5.3 Summary of modified model

The chest x-ray image represented as fig 5.4 was given as input and the output was predicted as shown in fig 5.5.

Fig 5.4 Sample input - 1

```
Python 3.8.5 Shell

File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:/Users/prith/Downloads/mobilenet.py ===============
No finding
>>>
```

Fig 5.5 Sample output - 1

The chest x-ray image represented as fig 5.6 was given as input and the output was predicted as shown in fig 5.7.

Fig 5.6 Sample input - 2


```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\prith\Downloads\mobilenet.py ================
Cardiomegaly
>>>
```

Fig 5.7 Sample output - 2

The chest x-ray image represented as fig 5.8 was given as input and the output was predicted as shown in fig 5.9.

Fig 5.8 Sample input - 3


```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\prith\Downloads\mobilenet.py ================
Pulmonary tuberculosis
>>>
```

Fig 5.9 Sample output - 3

Fig 5.10 & 5.11 shows the output of training and validation of the model with training accuracy of 77% approx. and testing accuracy of 75% approx. achieved with just 5 epochs.

```
353/368 [=============================>..] - ETA: 26s - loss: 0.7378 - acc: 0.7712
354/368 [=============================>..] - ETA: 24s - loss: 0.7378 - acc: 0.7712
355/368 [=============================>..] - ETA: 22s - loss: 0.7378 - acc: 0.7712
356/368 [=============================>.] - ETA: 20s - loss: 0.7378 - acc: 0.7712
357/368 [=============================>.] - ETA: 19s - loss: 0.7379 - acc: 0.7712
358/368 [=============================>.] - ETA: 17s - loss: 0.7379 - acc: 0.7712
359/368 [=============================>.] - ETA: 15s - loss: 0.7379 - acc: 0.7712
360/368 [=============================>.] - ETA: 14s - loss: 0.7379 - acc: 0.7712
361/368 [=============================>.] - ETA: 12s - loss: 0.7379 - acc: 0.7712
362/368 [=============================>.] - ETA: 10s - loss: 0.7379 - acc: 0.7712
363/368 [=============================>.] - ETA: 8s - loss: 0.7379 - acc: 0.7712
364/368 [=============================>.] - ETA: 7s - loss: 0.7379 - acc: 0.7712
365/368 [=============================>.] - ETA: 5s - loss: 0.7379 - acc: 0.7712
366/368 [=============================>.] - ETA: 3s - loss: 0.7379 - acc: 0.7712
367/368 [=============================>.] - ETA: 1s - loss: 0.7379 - acc: 0.7712
368/368 [==============================] - ETA: 0s - loss: 0.7379 - acc: 0.7712
368/368 [==============================] - 652s 2s/step - loss: 0.7379 - acc: 0.7712

Training Done

  1/123 [..............................] - ETA: 6:55 - loss: 0.9782 - acc: 0.7812
  2/123 [..............................] - ETA: 1:53 - loss: 1.1689 - acc: 0.7031
  3/123 [..............................] - ETA: 2:08 - loss: 1.2553 - acc: 0.6771
  4/123 [..............................] - ETA: 2:12 - loss: 1.1445 - acc: 0.7031
  5/123 [>.............................] - ETA: 2:18 - loss: 1.0857 - acc: 0.7125
  6/123 [>.............................] - ETA: 2:20 - loss: 1.0538 - acc: 0.7135
  7/123 [>.............................] - ETA: 2:21 - loss: 0.9998 - acc: 0.7232
  8/123 [>.............................] - ETA: 2:19 - loss: 0.9652 - acc: 0.7383
  9/123 [=>............................] - ETA: 2:19 - loss: 0.9630 - acc: 0.7361
 10/123 [=>............................] - ETA: 2:18 - loss: 0.9299 - acc: 0.7469
 11/123 [=>............................] - ETA: 2:17 - loss: 0.9106 - acc: 0.7528
```

Fig 5.10 Training model output - 1

```
 92/123 [=====================>........] - ETA: 33s - loss: 0.8680 - acc: 0.7527
 93/123 [=====================>........] - ETA: 32s - loss: 0.8690 - acc: 0.7527
 94/123 [=====================>........] - ETA: 30s - loss: 0.8672 - acc: 0.7523
 95/123 [======================>.......] - ETA: 29s - loss: 0.8643 - acc: 0.7530
 96/123 [======================>.......] - ETA: 28s - loss: 0.8623 - acc: 0.7539
 97/123 [======================>.......] - ETA: 27s - loss: 0.8649 - acc: 0.7532
 98/123 [======================>.......] - ETA: 26s - loss: 0.8676 - acc: 0.7529
 99/123 [=======================>......] - ETA: 25s - loss: 0.8656 - acc: 0.7532
100/123 [=======================>......] - ETA: 24s - loss: 0.8698 - acc: 0.7531
101/123 [=======================>......] - ETA: 23s - loss: 0.8723 - acc: 0.7525
102/123 [=======================>......] - ETA: 22s - loss: 0.8723 - acc: 0.7525
103/123 [========================>.....] - ETA: 21s - loss: 0.8770 - acc: 0.7515
104/123 [========================>.....] - ETA: 20s - loss: 0.8763 - acc: 0.7515
105/123 [========================>.....] - ETA: 19s - loss: 0.8736 - acc: 0.7524
106/123 [========================>.....] - ETA: 18s - loss: 0.8738 - acc: 0.7527
107/123 [=========================>....] - ETA: 17s - loss: 0.8791 - acc: 0.7512
108/123 [=========================>....] - ETA: 16s - loss: 0.8763 - acc: 0.7517
109/123 [=========================>....] - ETA: 14s - loss: 0.8746 - acc: 0.7523
110/123 [=========================>....] - ETA: 13s - loss: 0.8765 - acc: 0.7517
111/123 [==========================>...] - ETA: 12s - loss: 0.8767 - acc: 0.7514
112/123 [==========================>...] - ETA: 11s - loss: 0.8729 - acc: 0.7528
113/123 [==========================>...] - ETA: 10s - loss: 0.8693 - acc: 0.7539
114/123 [==========================>...] - ETA: 9s - loss: 0.8698 - acc: 0.7536
115/123 [===========================>..] - ETA: 8s - loss: 0.8718 - acc: 0.7524
116/123 [===========================>..] - ETA: 7s - loss: 0.8692 - acc: 0.7530
117/123 [===========================>..] - ETA: 6s - loss: 0.8660 - acc: 0.7537
118/123 [===========================>..] - ETA: 5s - loss: 0.8652 - acc: 0.7542
119/123 [============================>.] - ETA: 4s - loss: 0.8635 - acc: 0.7547
120/123 [============================>.] - ETA: 3s - loss: 0.8624 - acc: 0.7552
121/123 [============================>.] - ETA: 2s - loss: 0.8612 - acc: 0.7554
122/123 [============================>.] - ETA: 1s - loss: 0.8611 - acc: 0.7554
123/123 [==============================] - ETA: 0s - loss: 0.8597 - acc: 0.7559
123/123 [==============================] - 135s 1s/step - loss: 0.8597 - acc: 0.7559
>>>
```

Fig 5.11 Training model output - 2

Fig 5.12 & 5.13 shows the output of training the LIME explainer algorithm.
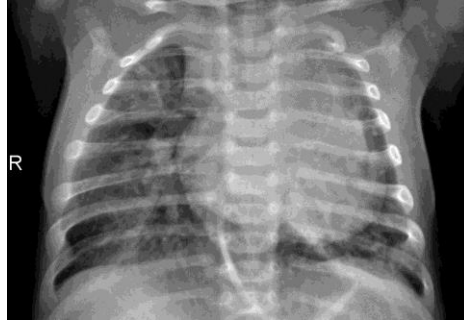

Fig 5.12 Training LIME Explainer algorithm part – 1

Fig 5.13 Training LIME Explainer algorithm part – 2

The chest x-ray image represented as fig 5.14 was given as input to LIME Explainer algorithm and the model's predicted output is explained by the algorithm as represented in fig 5.15, fig 5.16 and fig 5.17.

Fig 5.14 Sample input - 4

```
In [288]: temp, mask = explanation.get_image_and_mask(label=1, positive_only=True,
                                                      num_features=15, hide_rest=True,
                                                      min_weight=0.0000004)
          tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
          plt.rcParams['figure.figsize'] = (16.0, 12.0)
          plt.imshow(mark_boundaries(tempp, mask));
```
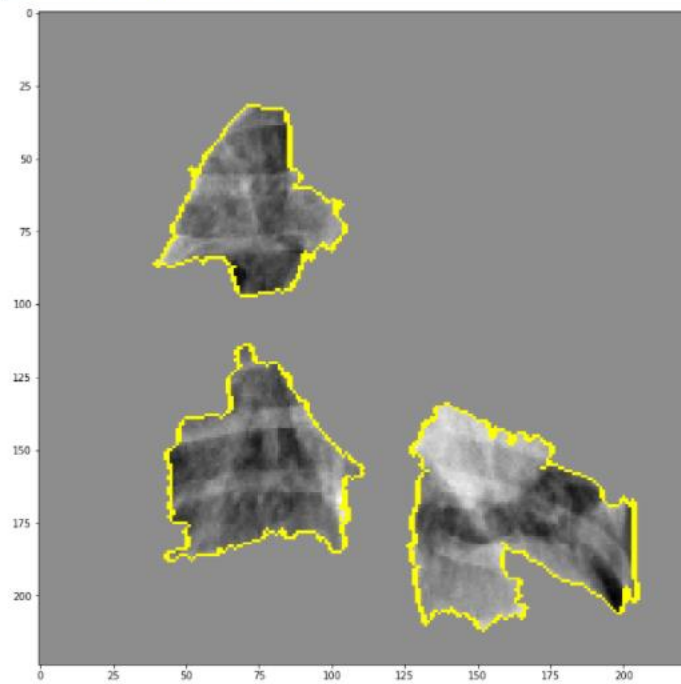

Fig 5.15 Sample output - 4 (part - 1)

```
In [289]: temp, mask = explanation.get_image_and_mask(label=1, positive_only=True,
                                                      num_features=15, hide_rest=False,
                                                      min_weight=0.0000004)

          tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
          plt.imshow(mark_boundaries(tempp, mask));
```



```
In [290]: temp, mask = explanation.get_image_and_mask(label=1, positive_only=False,
                                                      num_features=15, hide_rest=False,
                                                      min_weight=0.0000004)
          tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
          plt.imshow(mark_boundaries(tempp, mask));
```


Fig 5.16 Sample output - 4 (part - 2)


```
In [290]: temp, mask = explanation.get_image_and_mask(label=1, positive_only=False,
                                                      num_features=15, hide_rest=False,
                                                      min_weight=0.0000004)
          tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
          plt.imshow(mark_boundaries(tempp, mask));
```
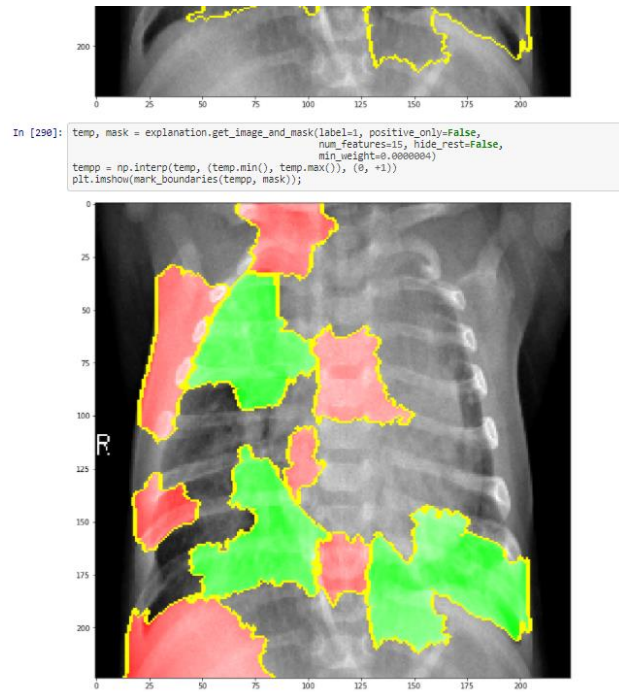

Fig 5.17 Sample output - 4 (part - 3)

The chest x-ray image represented as fig 5.18 was given as input to LIME Explainer algorithm and the model's predicted output is explained by the algorithm as represented in fig 5.19, fig 5.20 and fig 5.21.
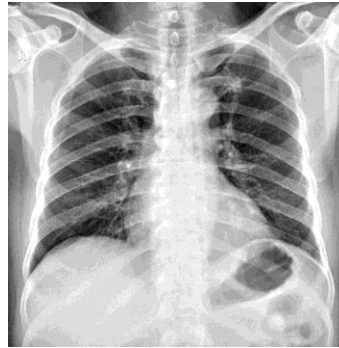
Fig 5.18 Sample input - 5
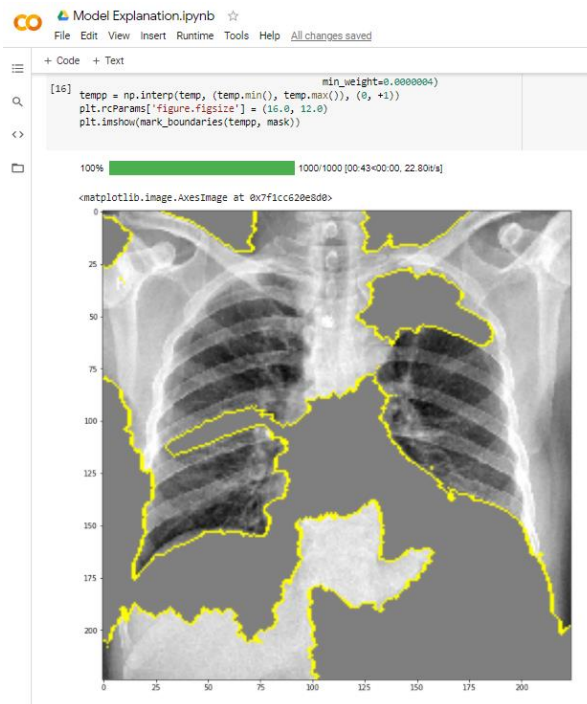
Fig 5.19 Sample output - 5 (part - 1)

Fig 5.20 Sample output - 5 (part - 2)

Fig 5.21 Sample output - 5 (part - 3)

The chest x-ray image represented as fig 5.22 was given as input to LIME Explainer algorithm and the model's predicted output is explained by the algorithm as represented in fig 5.23, fig 5.24 and fig 5.25.

Fig 5.22 Sample input – 6
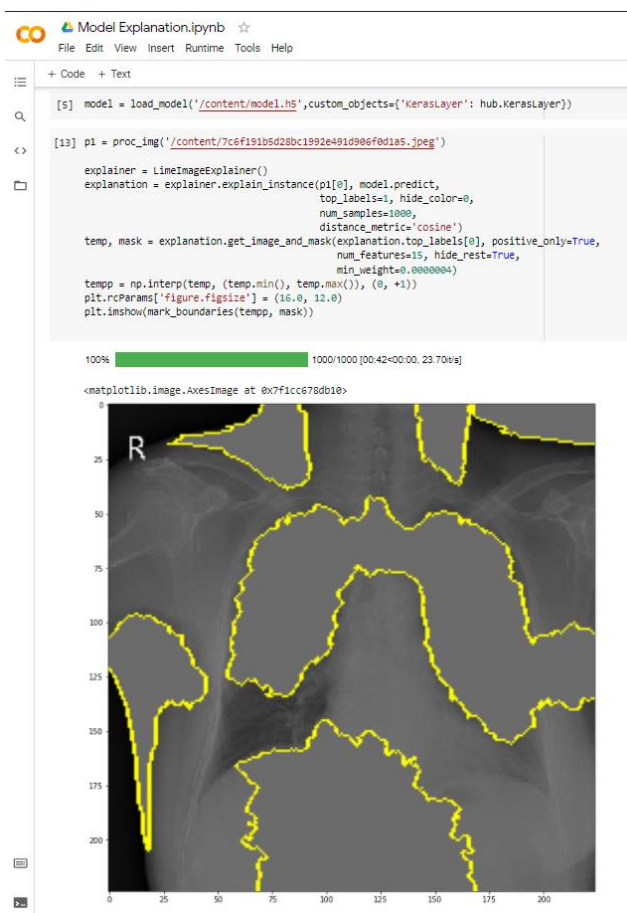
Fig 5.23 Sample output - 6 (part - 1)

Fig 5.24 Sample output - 6 (part - 2)

Fig 5.25 Sample output - 6 (part - 3)

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. Below Fig 5.26 shows the confusion matrix of the model.
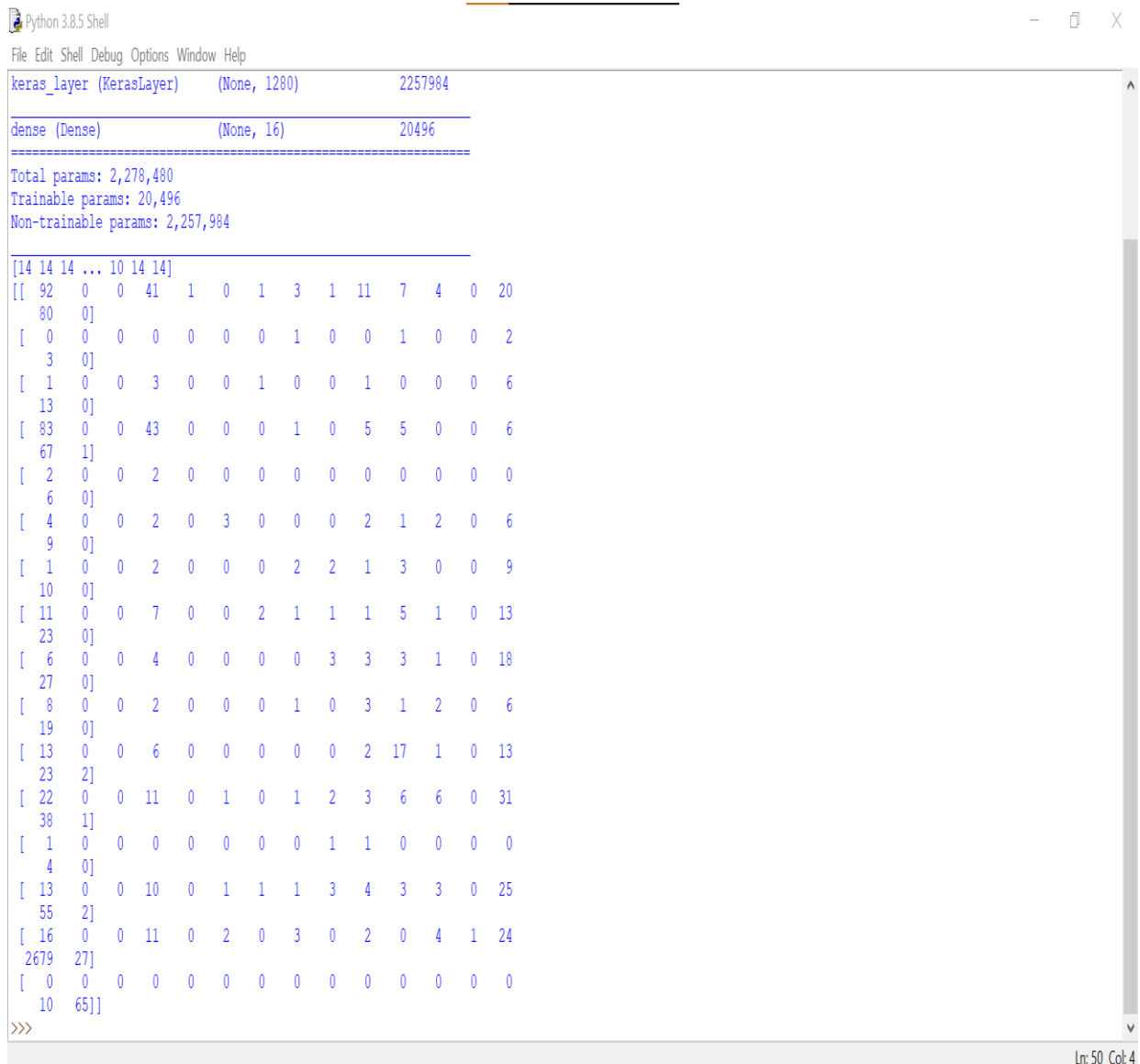
Fig 5.26 Confusion Matrix

Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR). The best sensitivity is 1.0, whereas the worst is 0.0.

Specificity measures the proportion of actual negatives that are correctly identified as such (tn / (tn + fp)). This metric creates four local variables, true_positives , true_negatives , false_positives and false_negatives that are used to compute the specificity at the given sensitivity.

Class Label	Sensitivity	Specificity
Aortic enlargement	78.24	65.04
Atelectasis	45.09	95.37
Calcification	42.89	96.67
Cardiomegaly	70.45	80.29
Consolidation	20.19	97.27
ILD	80.58	50.14
Infiltration	45.79	89.42
Lung Opacity	53.21	72.82
Nodule/Mass	75.43	65.49
Other lesion	81.52	50.11
Pleural effusion	79.23	40.01
Pleural thickening	49.18	70.24
Pneumothorax	85.23	72.35
Pulmonary fibrosis	80.24	50.33
No finding	96.74	66.25
Pulmonary tuberculosis	86.66	99.11

Table 5.1 Sensitivity and Specificity

CHAPTER 6

SYSTEM TESTING AND MAINTENANCE

6.1 GENERAL

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the user. It is performed to test the system beyond the bounds. The dataset is split as training, validation and testing set with 10000 images for training, 2000 for validation and 3000 for testing respectively.

VALIDATION SET:

A validation dataset is a dataset of examples used to tune the hyperparameters (i.e., the architecture) of a classifier. It is sometimes also called the development set or the "dev set", with each epoch during training, the model will be trained on the data in the training set. Well, it will also simultaneously be validated on the data in the validation set. The data in the validation set is separate from the data in the training set. So, when the model is validating on this data, this data does not consist of samples that the model already is familiar with from training. One of the major reasons of using a validation set is to ensure that the model is not overfitting to the data in the training set.
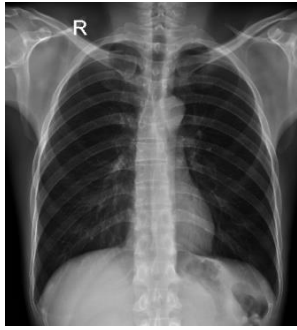
Label: No Finding   **Label:** Aortic enlargement   **Label:** Pneumothorax
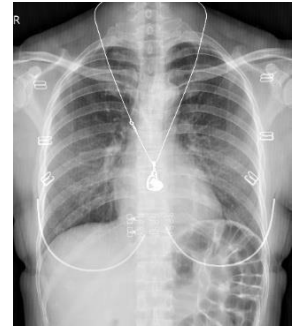
Label: Lung Opacity   **Label:** Cardiomegaly

Fig 6.1 VALIDATION IMAGES

In accordance with the above result, training the model with validation set has been effective and helped us with checking the parameters to adjust so that overfitting can be prevented. Overfitting is a modeling error that occurs when a function is too closely fit to a limited set of data points. Thus, attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power. Hence with the help of validation set and testing set the model is modified or corrected to predict correct caption almost 80 percent of times thus making it efficacious. As discussed in the above section implementations and results have shown the accuracy of how well the machine detects the diseases has along with system testing. To make the model more accurate the model has to be trained and validated with more set of dataset so that it can be used at production level.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 FUTURE ENHANCEMENT

In this model, by using the latest available datasets, 15 common thoracic lung diseases has been predicted which can be increased to any number of diseases in future as there will be more number of publicly available datasets. Many features such as patient's previous medical records, their age, and their living environment can be taken as factors to build the model, so that the diseases can be deducted more accurately and efficiently.

7.2 CONCLUSION

This Deep Convolutional Neural Network Model will be able to detect the thoracic lung diseases present in a given Chest X-Ray image accurately. Further with the use of LIME Algorithm, localization is done which explains the model's prediction.

APPENDIX

SOURCE CODE

Dicom_to_JPEG.py

```python
import os
import tensorflow as tf
import tensorflow_io as tfio


# Reading DICOM images
def read_dicom(path):
    image_bytes = tf.io.read_file(path)
    image = tfio.image.decode_dicom_image(
        image_bytes,
        dtype = tf.uint16
    )
    image = tf.squeeze(image, axis = 0)
    image = tf.image.resize(
        image,
        (500, 500),
        preserve_aspect_ratio = True
    )
    image = image - tf.reduce_min(image)
    image = image / tf.reduce_max(image)
    image = tf.cast(image * 255, tf.uint8)

    return image


# Destination folder
destination = "E:\\VIN-Train-jpeg\\"
os.makedirs("train_jpeg", exist_ok = True)

# Source folder
source = "E:\\VinDr-CXR\\train\\"
for name in os.listdir(source):
    image = read_dicom(os.path.join(source, name))
    image = tf.io.encode_jpeg(
        image,
        quality = 100,
```

```
        format = 'grayscale'
    )
    name = name.replace(".dicom", ".jpeg")
    tf.io.write_file(os.path.join(destination, name), image)
)
```

Png_to_JPEG.py

```
from PIL import Image
import os
c=0
source="E:\\tubercolosis dataset\\archive
(4)\\ChinaSet_AllFiles\\ChinaSet_AllFiles\\CXR_png\\"
for names in os.listdir(source):
    name=names[:-4]
    im1 = Image.open(source+names).convert("RGB")
    im1.save('E:\\TB JPEG\\'+name+'.jpeg')
    c=c+1
    if(c%100==0):
        print("Done")
```

Getnames_of_TBimg.py

```
import os
source="E:\\tubercolosis dataset\\archive
(4)\\ChinaSet_AllFiles\\ChinaSet_AllFiles\\CXR_png\\"

f=open("C:\\Users\\prith\\Downloads\\names.txt",'w')
for names in os.listdir(source):
    name=names[:-4]
    if(name[-1]=='0'):
        name+=',No finding,14'
    else:
        name+=',Pulmonary tuberculosis,15'
    f.write(name+"\n")

f.close()
```

Add_TB_data.py

```python
from csv import reader

with open('D:\\Downloads\\train.csv', 'r') as read_obj:
    f = []
    d = {}
    csv_reader = reader(read_obj)
    i = 0
    for row in csv_reader:
        if(i == 0):
            i = 1
            continue
        image_id = row[0]
        image_class_id = int(row[2])
        if(image_id not in f):
            f.append(image_id)
            arr = [0] * 16
            arr[image_class_id] = 1
            d[image_id] = arr
        else:
            arr = d[image_id]
            arr[image_class_id] = 1
            d[image_id] = arr
    with open('D:\\Downloads\\t.txt', 'w') as f:
        for key in d:
            value = key + ".jpeg,"
            string = ""
            for i in range(0,15):
                v = str(d[key][i]) + ","
                string += v
            string += str(d[key][15])
            value += string
            f.write(value + "\n")

f.close()
```

MobileNet.py

```python
import numpy as np
import cv2
import pandas as pd
import PIL.Image as Image
import os

import matplotlib.pylab as plt

import tensorflow as tf
import tensorflow_hub as hub
from sklearn.model_selection import train_test_split
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

IMAGE_SHAPE = (224, 224)

X, y = [], []

df=pd.read_csv('C:\\Users\\prith\\Downloads\\train1.csv')
path='C:\\Users\\prith\\Downloads\\VIN-Train-jpeg\\'

for i,row in df.iterrows():
        img_path=path+str(row['image_id'])+'.jpeg'
        img = cv2.imread(img_path)
        resized_img = cv2.resize(img,(224,224))
        X.append(resized_img)
        y.append(row['class_id'])

X = np.array(X)
y = np.array(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=0)

X_train_scaled = X_train / 255
```

```python
X_test_scaled = X_test / 255

feature_extractor_model = "https://tfhub.dev/google/tf2-
preview/mobilenet_v2/feature_vector/4"

pretrained_model_without_top_layer = hub.KerasLayer(
    feature_extractor_model, input_shape=(224, 224, 3),
trainable=False)

num_of_diseases = 16

model = tf.keras.Sequential([
  pretrained_model_without_top_layer,
  tf.keras.layers.Dense(num_of_diseases)
])

model.summary()

model.compile(
  optimizer="adam",

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['acc'])

model.fit(X_train_scaled, y_train, epochs=7)
print("Training Done")

model.evaluate(X_test_scaled,y_test)

model.save("model.h5")
```

Model.py

```python
import numpy as np
import cv2
import pandas as pd
import PIL.Image as Image
import os
```

```python
from keras.preprocessing import image
from keras.applications.mobilenet import preprocess_input

import matplotlib.pylab as plt

import tensorflow as tf
import tensorflow_hub as hub
from sklearn.model_selection import train_test_split
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from keras.models import load_model
from lime.lime_image import LimeImageExplainer
from skimage.segmentation import mark_boundaries


model = load_model('model.h5',custom_objects={'KerasLayer':
hub.KerasLayer})

model.summary()

img = cv2.imread('C:\\Users\\prith\\Downloads\\VIN-Train-
jpeg\\7c6f191b5d28bc1992e491d906f0d1a5.jpeg')
print(img.shape)
img1 = cv2.resize(img,(224,224))
print(img1.shape)
X = np.array([img1])
X=X/255

id_to_label = {
        'Aortic enlargement':0,
        'Atelectasis':1,
        'Calcification':2,
        'Cardiomegaly':3,
        'Consolidation':4,
        'ILD':5,
        'Infiltration':6,
        'Lung Opacity':7,
```

```
        'Nodule/Mass':8,
        'Other lesion':9,
        'Pleural effusion':10,
        'Pleural thickening':11,
        'Pneumothorax':12,
        'Pulmonary fibrosis':13,
        'No finding':14,
        'Pulmonary tuberculosis':15,
}


y=model.predict(X)
Id=np.argmax(y,axis=1)
print(ID)
print(id_to_label[ID])


y_pred = model.predict(X_test_scaled)
y_pred1 = np.argmax(y_pred, axis = 1)
print(y_pred1)


results = confusion_matrix(y_test,y_pred1)
print(results)
```

Model_Explainer.py

```
import numpy as np
import cv2
import pandas as pd
import PIL.Image as Image
import os


from keras.preprocessing import image
from keras.applications.mobilenet import preprocess_input


import matplotlib.pylab as plt


import tensorflow as tf
import tensorflow_hub as hub
from sklearn.model_selection import train_test_split
```

```python
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from keras.models import load_model
from lime.lime_image import LimeImageExplainer
from skimage.segmentation import mark_boundaries

def proc_img(img_path):
  "process the image"
  # `img` is a PIL image of size 224x224
  img = image.load_img(img_path, target_size=(224, 224))

  # `x` is a float32 Numpy array of shape (224, 224, 3)
  x_img = image.img_to_array(img, dtype='double')

  # We add a dimension to transform our array into a matrix
  # of size (1, 224, 224, 3)
  x_img = np.expand_dims(x_img, axis=0)

  # Finally we preprocess the batch
  # (this does channel-wise color normalization)
  #x_img = preprocess_input(x_img)
  x_img = preprocess_input(x_img)
  return x_img

model = load_model('/content/model.h5',custom_objects={'KerasLayer':
hub.KerasLayer})

p1 = proc_img('/content/73c08b6efeac411a439eb28d4c1aa0a1.jpeg')

explainer = LimeImageExplainer()
explanation = explainer.explain_instance(p1[0], model.predict,
                                   top_labels=1, hide_color=0,
                                   num_samples=1000,
                                   distance_metric='cosine')
temp, mask = explanation.get_image_and_mask(explanation.top_labels[0],
positive_only=True,
                                       num_features=15,
hide_rest=True,
```

```python
                                             min_weight=0.0000004)
tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
plt.rcParams['figure.figsize'] = (16.0, 12.0)
plt.imshow(mark_boundaries(tempp, mask))


temp, mask = explanation.get_image_and_mask(explanation.top_labels[0],
positive_only=True,
                                             num_features=15,
hide_rest=False,
                                             min_weight=0.0000004)


tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
plt.imshow(mark_boundaries(tempp, mask))


temp, mask = explanation.get_image_and_mask(explanation.top_labels[0],
positive_only=False,
                                             num_features=15,
hide_rest=False,
                                             min_weight=0.0000004)
tempp = np.interp(temp, (temp.min(), temp.max()), (0, +1))
plt.imshow(mark_boundaries(tempp, mask))
```

REFERENCES

[1] Worawate Ausawalaithong, Arjaree Thirach, Sanparith Marukatat, Theerawit Wilaiprasitporn – '*Automatic Lung Cancer Prediction from Chest X-ray Images Using the Deep Learning Approach*' , 2018 Biomedical Engineering International Conference (BMEiCON-2018).

[2] Rahib H Abiyev and Mohammad Khaleel Sallam Ma'aitah – '*Deep Convolutional Neural Networks for Chest Diseases Detection*', Hindawi Journal of Healthcare Engineering Volume 2018.

[3] Subrato,Bharati ,Prajoy Podder, M.Rubaiyat Hossain Mondal – '*Hybrid deep learning for detecting lung diseases from X-ray images*', (ELSEVIER) Informatics in Medicine Unlocked 2020.

[4] Anuradha D. Gunasinghe, Achala C. Aponso, Harsha Thirimannaand – '*Early Prediction of Lung Diseases*', 2019 IEEE 5th International Conference for Convergence in Technology (I2CT).

[5] Mrs. Rasika Naik, Mr. Tejas Wani, Ms. Sakshi Bajaj, Mr. Shiva Ahir, Mr. Atharva Joshi – '*Detection of Lung Diseases using Deep Learning*', Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020.

[6] Esha Dilipkumar Bhandari, Prof. Mahesh S. Badmera – '*Chest Abnormality Detection from X-ray using Deep Learning*', 2017 IEEE 3rd International Conference for Convergence in Technology (I2CT).

[7] Justin Monsi , Justine Saji , Keerthy Vinod , Liya Joy , Jis Joe Mathew – '*XRAY AI: Lung Disease Prediction Using Machine Learning*', International Research Journal of Engineering and Technology (IRJET) Volume: 06 Issue: 11 | Nov 2019.

[8] Ioannis Livieris, Andreas Kanavos, Panagiotis Pintelas – '*Detecting Lung Abnormalities From X-raysUsing anImproved SSL Algorithm*', Volume 8, No.2, March - April 2019.

[9] Yue Cheng, Jinchao Feng and Kebin Jiam – '*A Lung Disease Classification Based on Feature Fusion Convolutional Neural Network with X-ray Image Enhancement*', (ELSEVIER) Electronic Notes in Theoretical Computer Science 343 (2019).

[10] Mohammad Farukh Hashmi, Satyarth Katiyar, Avinash G Keskar, Neeraj Dhanraj Bokde and Zong Woo Geem – '*Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning*', Hindawi Journal of Healthcare Engineering Volume 2017.

[11] Yue Cheng , Jinchao Feng and Kebin Jia – '*A Lung Disease Classification Based on Feature Fusion Convolutional Neural Network with X-ray Image Enhancement*', 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).

[12] Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, Ankush Mittal – '*Pneumonia Detection Using CNN based Feature Extraction*', 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT).

[13] Arjun Choudhary, Abhishek Hazra, Prakash Choudhary – '*Diagnosis of Chest Diseases in X-Ray images using Deep Convolutional Neural Network*', 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[14] W.H. Hsu, F.J. Tsai, G. Zhang, C.K. Chang, P.H. Hsieh, S.N. Yang, S.S. Sun, KenYK Liao, Eddie TC Huang – '*Development of a Deep Learning Model for Chest X-Ray Screening*', MEDICAL PHYSICS INTERNATIONAL Journal, vol.7, No.3, 2019.

[15] Samir S. Yadav and Shivajirao M. Jadhav – 'Deep convolutional neural network based medical image classification for disease diagnosis', Journal of Big Data volume 6, Article number: 113 (2019).