PNEUMONIA DETECTION USING ARTIFICIAL NEURAL NETWORKS AND TRANSFER LEARNING MODEL

A PROJECT REPORT

Submitted by

HITARTH PANDEY [Reg No: RA1511003010433]
ABHISHEK SURYAVANSHI [Reg No: RA1511003010499]

Under the guidance of

Mr. U.M. Prakash

(Assistant Professor (O.G), Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2019

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**PNEUMONIA DETECTION USING ARTIFICIAL NEURAL NETWORKS AND TRANSFER LEARNING MODEL**" is the bonafide work of " **HITARTH PANDEY [Reg No: RA1511003010433], ABHISHEK SURYAVANSHI [Reg No: RA1511003010499]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr. U.M. Prakash
GUIDE
Assistant Professor (O.G)
Dept. of Computer Science & Engineering

SIGNATURE

Dr. B. Amutha
HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Developing a system that helps in detecting pneumonia in CXR images of lungs, at a high accuracy. Firstly, a raw image is being preprocessed with the help of Otsu Thresholding and an equalizer. This helps in detecting pneumonia clouds and identifying the ratio of healthy lung region to the total region minimum. The above task is determined by importing the original CXR image in the dataset and then calculating the ratio using cropping of image and applying openCV. The preprocessed data is then fed into Convolution Neural Network that accurately predicts the percentage of viral and bacterial pneumonia. This helps various radiologist in identifying various prescribed drugs. Moreover, using transfer learning for better and greater performance. This helps in identifying pneumonia and determining the prescribed drugs and help in diagnosing and clearing off the symptoms.

ACKNOWLEDGEMENTS

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

We extend our sincere thanks to Dr. C. Muthamizhchelvan, Director, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank Dr. B. Amutha, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor Dr. K. Annapurani, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her great support at all the stages of project work.

We would like to convey our thanks to our Panel Head, Dr. S. Prabakaran, Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his / her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, Mrs. S. Sharanya, Assistant Professor (O.G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, Mr. U.M. Prakash, Assistant Professor (O.G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his mentorship. He provided me the freedom and support to explore the research topics of my interest. His passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

HITARTH PANDEY

ABHISHEK SURYAVANSHI

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Pneumonia

In 1875, Edwin Klebs was the first person to observe the bacteria in the airways of a person that died due to pneumonia. What pneumonia is basically, an early late disease. It comes from a Greek word "pneumon" meaning "lung". It is usually caused by the infection of bacteria and viruses in healthy part of the lung region. It can be present in on or both the lungs. It causes inflammation in your air sacs which is known as alveoli. Pneumonia however a very much curable disease, are present in most in children and hence making their breathing insufferable. Mostly, pneumonia in children causes more problems in the lung than any old person. According to National Institute of Health-care, Chest X-Ray images are the best way to detect pneumonia. The various pneumonia present are viral, bacterial, and fungal. The most common type of pneumonia is bacterial pneumonia. There are various ways to cure pneumonia on the basis of what the patient might have. If the patient has a severe case, he/she might get hospitalized and certain therapy for example oxygen therapy is needed if the patient is low on oxygen. Moreover, it can also be cured by not smoking and washing hands. A study shows that if the person is a smoker it would definitely have a greater chance of pneumonia disease.

Pneumonia affects approximately 450 million people globally and results in 4 million deaths per year. It is also known as "the captain of the men of death" by William Osler. Certainly, there's an improve in the survival as the introduction of vaccines. But the leading cause of death still remains puzzled.

Hence, pneumonia is one of the most dangerous diseases of all time and hence detecting it could solve many problems.

1.2 Purpose of the Project

If we could determine the nature and where exactly the pneumonia has occurred, we could solve the problems faced by doctors which leads to late reports and time complexity. To ensure this, we must develop a system in which there are lots of training sets of person's history and background check and most importantly a Chest X-Ray image. This would do the trick and hence by the power of ANN we could determine the effects of the person and hence solve the problem.

1.3 Scope of Project

The project aims to modify the present review system in the health care. There are a large number of datasets in the present systems, so it becomes quite tedious for the client to read through all the data. We try to demonstrate a system with more insights on the patients health. This would be much more entertaining and user friendly.

1.4 Concept of Artificial Neural Network

Artificial Neural Network is a major concept for data analysis and provides a building block of how an artificial brain could work. Consider a system, where we need to determine the difference between an apple and a bomb. Now, for this we need the basic features of what exactly is the difference. These features then get trained by the model to determine and hence releases insights on how this is an apple or a bomb. Similarly, in our day to day scenarios we come across various applications which helps in detecting objects and getting to know the insights. These insights provide a way to solve various problems and then includes true cut accuracy which leads to greater performance in less time.

1.5 Dataset

The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal).

1.6 Transfer Learning

Transfer learning model is a model that is reused to define a specific task from the starting point. Apart from Machine Learning, it uses pre trained model that has been used for something else and it quick starts the processes by a new task format with the previous model performance. It is more effective performance wise. Below are the figures that helps us understand transfer learning in a more effective way.

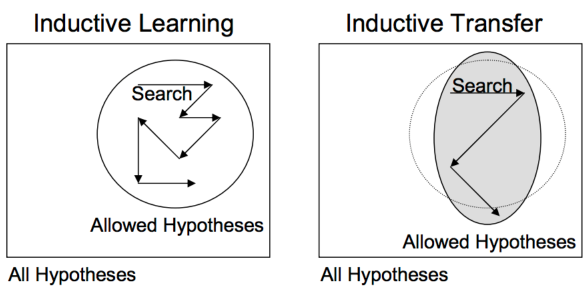


Figure 1.1: Depiction of Inductive Transfer Learning

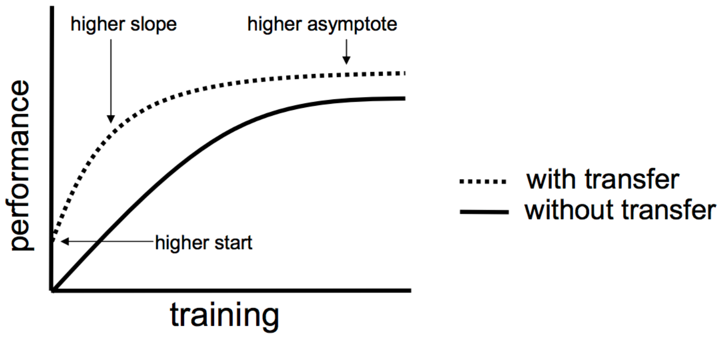


Figure 1.2: Performance graph of with and without Transfer Learning

CHAPTER 2

LITERATURE SURVEY

2.1 Survey Paper Analysis

2.1.1 Detection of Pneumonia Clouds

This paper [1] includes a very detailed analysis of image processing which determines the cloud region present in the lungs. Moreover, the model uses an algorithm known as Otsu Thresholding which determines the healthy lung region to the total region. This is then fed to filter and hence gets the minimum of the ratio. This minimum region is where the clouds are present. When looked closely, the dataset uses Chest X-Ray images of only 10,000 images and has an accuracy of 80%. The future insights given are very much understanding and provides a relationship between the healthy to the total region.

Moreover, the method adapts various image processing techniques from OpenCV and uses a wide range of equalizers. This is then used for inner sharpness and gradients of the image where the image shows some blurred areas which consist of clouds. This in fact provides a better understanding of how the image can be preprocessed and the clear idea indicates a pattern as to where the clouds might be present. The early model suggested the less accurate value, whereas this improved version provides a better look into the image.

The major drawback of this image is, it is not capable of rendering complex image structures and hence decreases its performance. As the dataset is really small compared to other large dataset, it fails to take up values beyond 10,000 and hence this indicates underfitting of the model. The testing set also have no insights about how the data could affect in future.

2.1.2 CheXnet Radiology Level Detection

This model [2] detects radiology level pneumonia with the help of a 121-layer Convolution Neural Network. The base paper gives us a closer look on the model detection which provides different layers to create a Convolution Neural Network. These layers are basically based on the different features of the images and gives us a clear idea about the different lines and curves to solve the problem.

The paper provides a singular approach of Convolution Neural Network and uses performance gradient techniques to make the model more and more precise. A clear idea provides simple classification of pneumonia and non-pneumonia. The 121 Layers provide stability to the model and gives insights based on how radiology level is determined. It includes various boosting techniques which provides definitive approach. Since the model provides a clear and creative insight, it fails to perform as presented. This model creates various other dependencies and creates a complex structure which is difficult to understand. Although, there is a 5% increase in the accuracy it generally makes it seemingly very much difficult and uneasy to work with. The model also provides a clear understanding of the difference in radiology level vs the 121- layer model, however it seems unclear as the requirements are quite unlikely.

2.1.3 Chest X-Ray Classification a Multi-Label and Fine-Grained Problem

This particular paper [3] depicts the Chest X Ray using multi-label classification. The classification is based on extraction of features which uses fine-grained learning model. This offers a variety of pretrained models which are classified as CNN architectures. This paper demonstrates the effectiveness using multi-label loss function (MSML). These provide a clear idea about different types of model used to determine the multi-label classification.

It also uses multi-label learning loss and bilinear polling to determine the methodologies involved in the classification of pneumonia chest images. These provide a different set of approach which results in greater performance issues. The dataset contains

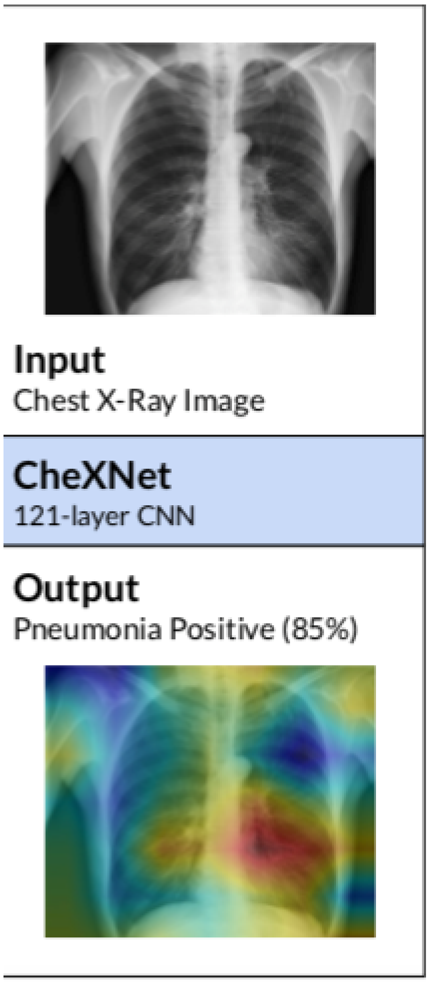
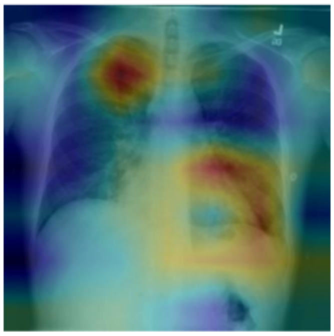
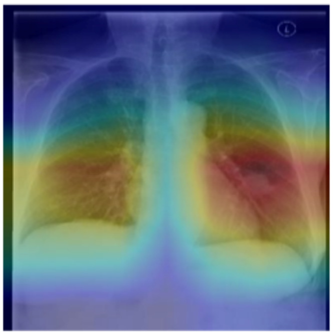


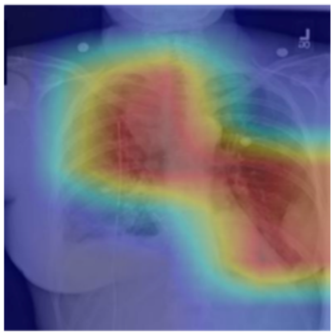
Figure 2.1: CheXnet Radiology Level Pneumonia Detection



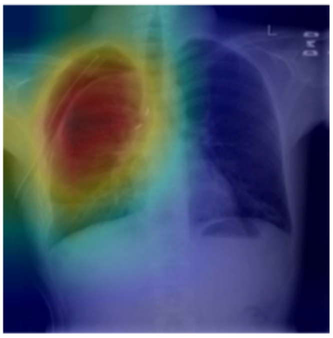
(a) Patient with multifocal community acquired pneumonia. The model correctly detects the airspace disease in the left lower and right upper lobes to arrive at the pneumonia diagnosis.



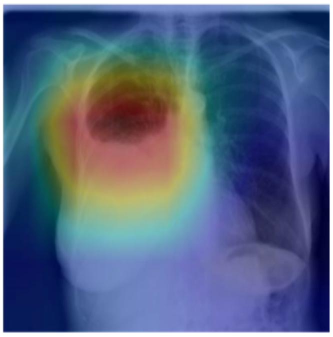
(b) Patient with a left lung nodule. The model identifies the left lower lobe lung nodule and correctly classifies the pathology.



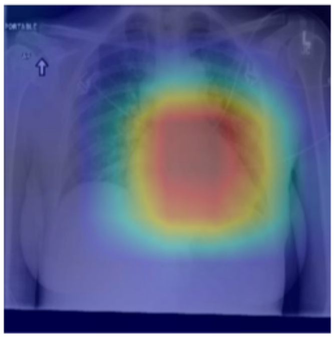
(c) Patient with primary lung malignancy and two large masses, one in the left lower lobe and one in the right upper lobe adjacent to the mediastinum. The model correctly identifies both masses in the X-ray.



(d) Patient with a right-sided pneumothroax and chest tube. The model detects the abnormal lung to correctly predict the presence of pneumothorax (collapsed lung).



(e) Patient with a large right pleural effusion (fluid in the pleural space). The model correctly labels the effusion and focuses on the right lower chest.



(f) Patient with congestive heart failure and cardiomegaly (enlarged heart). The model correctly identifies the enlarged cardiac silhouette.

Figure 2.2: The above figures depict the difference of regions of patients having pneumonia positive.

70% for training ,10% of validation and 20% of testing. These good splits up provide a clear picture of how to represent a particular dataset and hence provides greater relationships.

However, the main drawback is the interdependencies on the datasets. The reason behind is the case of overfitting where the training data is more and also does not hold good in these situations. However, the idea behind is of greater effect and hence determines a good way of detecting using labels.

2.1.4 Automatic Pneumonia Detection Based on Ultrasound Video Analysis

Provides [4] a broad perspective in detecting the images with ultrasound techniques. There is various background check to be considered as a matter of fact and relates in identification of two important cues. The analysis is based on lung sonography which indicates pleural lines present in the lung and the other one is the presence of lung consolidation. The algorithm provides a clear idea of how the detection is done. Firstly, a video frame is added into the process for thresholding and morphological operations. After this, the upper frame gets eliminated leaving only the lung region. This is done due to unnecessary lines present inside the image blocking the overall view. After this process the algorithm further investigates the pleural lines and identifies them based on the sonography. There is further selection based on the centroid analysis which is done with the help of clustering. This clustering results in grouping of high intensity signatures of the detecting lines and hence providing the necessary output.

One of the main benefits of using video analysis is it can be very efficient in planning out different classifications of various images. On the other hand, we cannot distinguish between the terms we need to train or test. It cannot be determined for lack of training provided it can be used for a preprocessed system.

2.1.5 Detecting Out-of-Bed Activities to prevent Pneumonia for Hospitalized Patient using Microsoft Kinect V2

It actually indicates that how much of a pneumonia patient is active in an out-of-bed environment[5]. It is given that the HAP is closely related to the patients activity during its stay. It offers a detection system of patients active time out of bed. Since, most of the diseases are caused in the hospitals via flow of viruses, it seems that detecting those might help improve the effectiveness of patients routine and quantity time spend in bed. The basic data recording and modelling is done by the Kinect V2. This trained dataset is then pushed into a classifier which results in the output of how much exercise or some activities a patient could do to improve.

The method does not provide any insights as to what could have gone wrong or any detailed analysis of the patients health reports. It simply states that it records the activities but clearly is unable to provide any relevant details. Moreover, hospitalized patients are encouraged by the healthcare to move as early as possible after there is treatment. The lack of insights and unclearness of data makes it less appealing and unable to understand. It also provides a background look of a studio which is quite favorable on its side.

2.1.6 Smart Algorithms for Detecting Childhood Pneumonia

The fact that more than 60% of pneumonia are present in children is the only reason for the paper to create some meaningful insights and smart resource constrained algorithms to provide necessary feedbacks[6]. It uses various methods and parameters for detecting. They are clinical parameters, observational parameters and conventional vital signs. The analysis is performed in matlab2014. The methodologies include similar to what the base paper has and that is, preprocessing, feature selection and finally classification. It also provides a clear data of various child patients and hence is considered good for various application.

Since, the dataset is constrained only for child in effect there are several factors like smoking are missing. This is because many children below the age do not smoke. But overall, the paper provides a positive insight as to why these many children are gaining pneumonia on a much smaller level.

		Relevance	mRMR	Relief	Gram-Schmidt	Lasso	Elastic net	sLDA	Measurable	Point-of-care	Affordable
Vital signs	Temperature	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
	HR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	RR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Osat	N	Y	Y	N	N	N	Y	Y	Y	Y
Measurable clinical features	WBC count	Y	N	N	N	N	N	N	Y	N	N
	Neutrophils	Y	Y	N	Y	Y	Y	N	Y	N	N
	Lymphocytes	N	Y	N	Y	Y	Y	N	Y	N	N
	WHZ	N	Y	N	Y	Y	Y	Y	Y	Y	Y
	Age	N	N	Y	N	N	N	Y	Y	Y	Y
Lung sounds	Grunting	Y	N	N	N	N	N	N	Y	Y	Y
Observational clinical features	Sleepy	Y	N	Y	N	N	N	N	N	N	Y
	Pallor	Y	N	N	N	N	N	N	N	N	Y
	Head nodding	Y	N	N	N	N	N	N	N	N	Y
	Cough	N	Y	Y	Y	Y	Y	Y	N	N	Y
	Yellow fever	N	N	Y	N	N	N	N	N	N	Y
	Unwell	N	N	N	N	N	N	Y	N	N	Y
	Cyanosis	N	N	N	N	N	N	Y	N	N	Y

Figure 2.3: The Feature Selected Data consisting of 4 factors and their response

2.1.7 Preliminary Study of Pneumonia Symptoms Detection using Cellular Neural Network

This pape[7]r demonstrates a series of Cellular Neural Network templates that gives a pre-check for pneumonia detection. Here a whole bunch of area is detected using the templates with the help of an algorithm. The template is described as median, shift-n, D-median, half-toning. What is interesting about this is the area of pneumonia calculated within a time frame and is very much quick. Moreover, the study shows a nonlinear dynamical view, this view is obtained as a true output and hence considered the best. A set of data validation is also obtained. However, the data validation curves are not represented and hence requires more background knowledge. The software used is Candy which is unaware of these days and cannot be established.

Various images are shown as a stepping stone. These images are obtained after the processing has done. Hence, there is no clear idea about how the images are formed.

The use of templates is explained but looks like incomplete. The paper adopted a unique 3-D or a dynamic view.

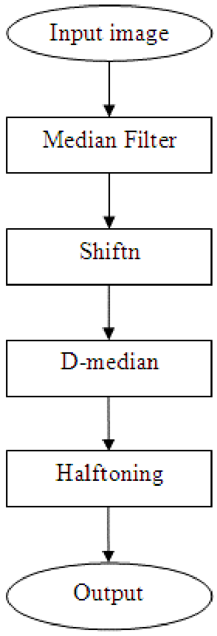


Figure 2.4: Flowchart of the analogic CNN Algorithm

2.1.8 Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for CXR Analysis of Lung Cancer

This paper[8] has an all new different kind of process. It adopts a Machine Learning technique that uses lung segmentation which results in dealing with each lung separately. Also the shadowing that appears on the CXR images are eliminated giving a clear picture of detection of a direct lung cancer. This might not result in any of the insights that we needed, it helps in various data preprocessing techniques which makes the model more better performance wise. These segmentation and elimination of Lungs and the middle bone ribs gives us a look for clouds present.

The amount of work done is based on just OpenCV and ML. These things gives us various remodeling techniques to obtain detailed analysis of the lung and its behavior.

Figure 2.5: Shows the process of lung segmentation with bones

2.1.9 Automatically Diagnosing Diseases on CXR Using Deep Neural Networks

An automatic approach[9] for deep neural network is quite fascinating. This provides a whole lot of network for a multi-label approach. The model is similar to the fine-grained problem and hence requires different set of functions. In this particular model, various loss function are obtained by entropy techniques. The various loss equations are as follows : The equation 1-5 are cross weighted cross entropy loss function defined in cross weight.

$$L_{W-CEL}(f(\vec{x}), \vec{y}) = \beta_P \sum_{y_c=1} -\ln(f(x_c)) + \beta_N \sum_{y_c=0} -\ln(1 - f(x_c)), \quad (1)$$

$$\beta_P = \frac{|P| + |N|}{|P|}, \quad (2)$$

$$\beta_N = \frac{|P| + |N|}{|N|}. \quad (3)$$

For updates, the weights obey the following equations:

$$W \leftarrow W - \Delta W, \quad (4)$$

$$\Delta W = \beta_P \sum_{y_c=1} \frac{-\partial ln(f(x_c))}{\partial W} + \beta_N \sum_{y_c=0} \frac{-\partial ln(1-f(x_c))}{\partial W} \quad (5)$$

The below figure provides a clear idea about how the process is obtained. The model has clearly been the significant of all.

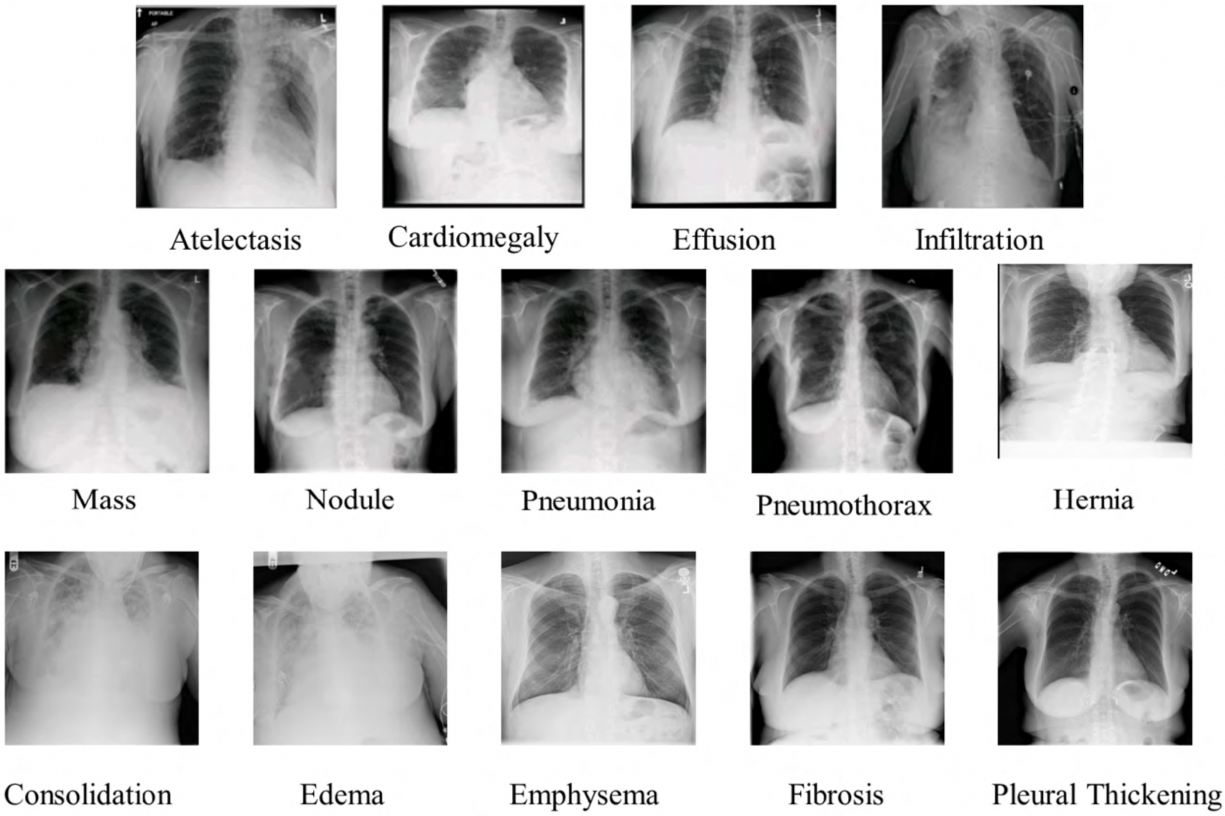


Figure 2.6: Sample of 14 diseases after performing the training on the dataset

2.1.10 Tabla: An Acoustic Device Designed for Low Cost Pneumonia Detection

The paper[10] develops a prototype model to detect pneumonia based on physiological sensing and spectrogram with the help of a transfer function. The function appears to be sensing the amount of vitals per minute. The device design engine is basically a surface exciter that deals with a PCB and a battery.

Moreover, device uses various mechanism to provide a safe keeping of every data and also results in accurate results. The main problem is lack of performance. The more general datasets are more powerful and reliable. Here, only a doctor can be handy. The overall hardware design requires lot of background knowledge about different positions of the PCB and how it is maintained. Here is a clear view of how the process is going to work.

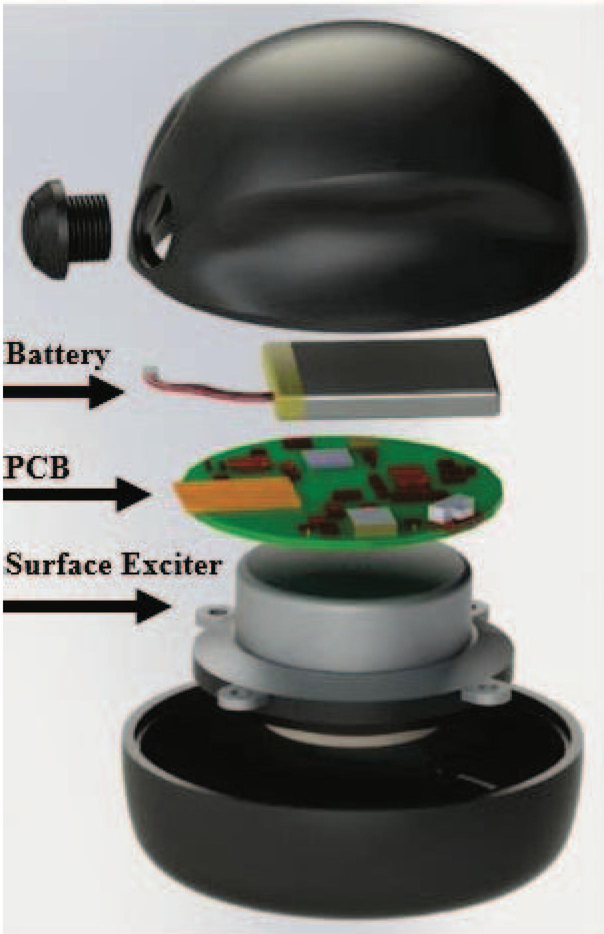


Figure 2.7: Device Assembly

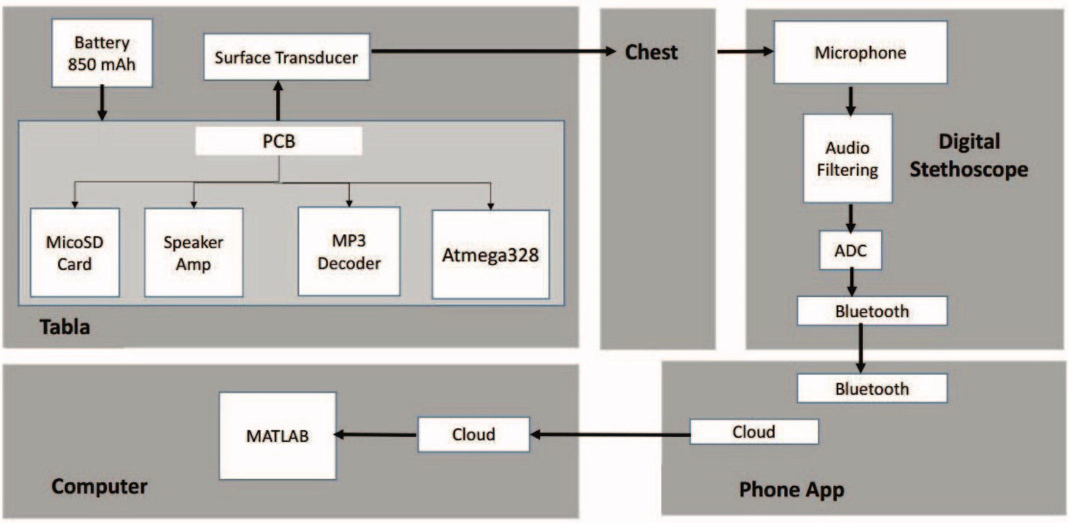


Figure 2.8: Block Diagram of the Device

2.2 Inference from the Survey

After reading various papers and analyzing them, the solution is to develop a mere simple app to determine pneumonia detection. This also offer a combination of all the other models which develops system generated performance on the basis of images and hence provide useful insights on development. A recent development carries new tasks, where we happen to obtain great results on such events. These then requires different sets to solve the problem.

The main theme provides a detailed graph-based system and concludes the accuracy with transfer learning capabilities and hence a convolution matrix is obtained. This offers a predicted accuracy to maintain an attained system which intern results in proper aspects of solving the problem. These then creates various analogies that what if a person has different type of pneumonia or how the medicine is really curing the disease.

Another important aspect is to select the right amount of dataset. These CXR images provide a different and clear view of how these problems happen to be. The papers offered provide a clear way of representing these dataset on such levels.

CHAPTER 3

ANALYSIS OF THE EXISTING SYSTEM

3.1 Introduction

The existing system provides a less performance-based system in which the process is very much complex and tedious. The systemâĂŹs environment is first based on the input of data, using CNN for training of the data and then obtaining insights. The process might have a negative feedback on how important performance and accuracy is. It also happens to be created based on the data that was given in very old. Most of this data happens to be lost. The system offers different F1 scores to increase the performance based on radiology level and also determines different approaches for the same.

Table 3.1: F1 Scores

	F1 Score (95% CI)
Radiologist 1	0.383 (0.309, 0.453)
Radiologist 2	0.356 (0.282, 0.428)
Radiologist 3	0.365 (0.291, 0.435)
Radiologist 4	0.442 (0.390, 0.492)
Radiologist Avg.	0.387 (0.330, 0.442)
CheXNet	0.435 (0.387, 0.481)

The above table explains about how the radiologist got the F1 scores. Mostly it outlines the other system which explains why CheXnet offers to be one of the prominent existing systems. Moreover, the table explains how are the scores in relation with the model and maintains a sight of action planned for the existing system. The role is to offer a greater performance in short amount of time. Another important aspect of this system is to maintain a certain amount of features in a singular time frame.

The system also deals with the insights on a level based on a not well-trained model. The CNN approach often falls into losses and hence is considered to be effective yet outdated. The system offers no UI control system and hence uses Machine Learning

Tools which is quite beneficiary. In these processes the lung model is not at all used for segmentation purposes but adopts an easy understanding. A contradictory aspect is fine seen by the estimation during testing and hence the methodologies is quite not application based. Hence, a new system with a positive feedback and a ground breaking performance is needed.

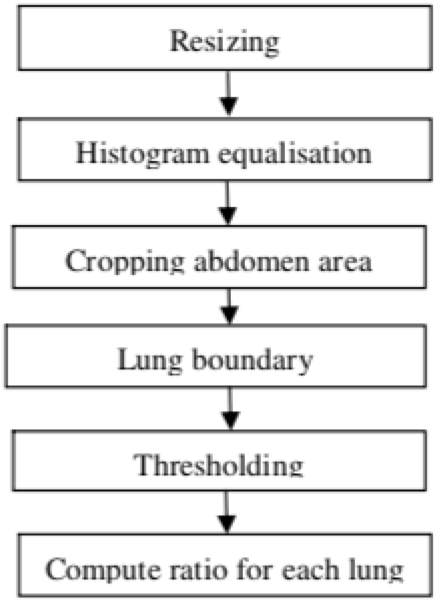


Figure 3.1: Present System for preprocessing and then training using CNN

The figure offers a clear idea of the steps applied on the scales. It consists of the steps which happened to be applied on each and every image to be trained. Firstly, the image is resized to get a basic image portion to be trained. Next, it is fed for histogram equalization. This provides a more sharper view of the image. Then, the abdomen area is cropped. A lung boundary is then left for detection and hence it is used for computation of each lung.

This offers a clear area as to where is the cloud regions that happen to be pneumonia positive. Then the data is fed into CNN for get a performance graph for testing. This method is tedious as we don't know what happens to the other regions and their parts. These also distinguishes two lungs and hence happened to be a very complex system.

CHAPTER 4

PROPOSED WORK

4.1 Introduction

The proposed work is basically a combination of all the existed work. It consists of a new method of training the data, the transfer learning model. A transfer learning model uses a pre-trained model to obtain a more greater accuracy. These model uses an old tasks to create a new and refrained tasks. Transfer learning is a new methodology which makes a very much adaptive learning for datasets on any format.

The idea behind a pneumonia detection was obtained by famous data scientist Andrew Ng. This created a more drastic change in the healthcare field. The prior knowledge required were very much in making it more efficient. The work offers different processes to be done in a single dataset which then is used for further investigation.

4.2 Steps for Process

The primary steps that are being followed for pneumonia detection are

1. Data Preprocessing
2. Model Building
3. Using Transfer Learning for Training
4. Obtaining Results
5. Providing Insights from the Results

4.3 Tools and Methodologies

The tools and methodologies are

1. OpenCV (Data Preprocessing)

2. Convolution Neural Network (Data Modelling)

3. Transfer Learning (Training)

4. Scikit-Learn (Graphical Insights)

The tools and methodologies given above are done by Python3.7 framework and with the help of Kaggle Environment.

4.4 Hardware and Software Requirements

The hardware and software requirements are

1. Hardware:

1.1 16GB RAM

1.2 Greater processing power like i5 or i7

1.3 Greater GPU capabilities

2. Software:

2.1 Anaconda Navigator

2.2 Jupyter Notebook

2.3 Dataset containing folders (NORMAL and PNEUMONIA)

2.4 Star UML (Module Design)

4.5 Implementation of Proposed Work

The model offers stability upon providing a combination of the previous works. This happened to be done by giving different scenarios where the image is gone through various processes. Then the image gets optimized which results in perfect accuracy. The model predicts it in a more convenient way. Hence, it offers every agile process and represents good model building.

CHAPTER 5

MODULE DESIGN

5.1 Introduction

Module operations and design is a key role in identifying a prototype of a system. Generally, it happens to play a vital role in how your system might act in different environments. Various operations are performed to develop a structure for design scenarios. The module design contains a short hand description of the system and its capabilities. The 6 major description used are
1. Model Preprocessing
2. Model Building
3. Model Training
4. Validation
5. Evaluation
6. Classification
Here we will see one by one what each module is doing.

5.1.1 Model Preprocessing

Model Preprocessing provides a set of various preprocessing techniques for CXR images. The dataset which consists of CXR images is fed into this model to obtain a preprocessed image. Several techniques are used to determine this operation. The most efficient one is the Otsu Thresholding which creates a sharp image that could happen to be very dull. This is then made visible with the help of equalizer technique.

5.1.2 Model Building

Model Building builds the preprocessed model into set of convolutions which creates different sets of matrices in a 2-D approach. The matrix provides a clear-cut idea about the preprocessed image as to where we can train the model. These then offers a trained model of convolutions which are yet to be trained.

5.1.3 Model Training

Model Training trains the model from the build. It uses InceptionV3 algorithm to determine pneumonia positive in the lung region. This uses an algorithm as to if found a set of areas where the ratio is minimum then detect it else leave it as it is for new task. This offers to deal with the performance issues and hence makes it easier. The output obtained is then fallen under the validation and evaluation.

5.1.4 Evaluation and Validation

Evaluation provides evaluating the results based on test data. The data is calculated using various techniques such as mean squared error where we determine the average data to the test data for accuracy. The score for the accuracy obtained is 95

5.1.5 Classification

Model Classification creates classified folders as to Normal vs Pneumonia. The classified images are then put under insights for further investigations. These are then provided for clearing off the symptoms present in the image. It gives a handy guide as to what the patient has to perform to undergo in removing of pneumonia. The classified image is the final accurate output and hence the system gives action based on the classified image. The classification is done as to what type of pneumonia the person has and how much he/she might undergo some operation for the removal. In ideal conditions there are graphs and plots that provide guide to the patient. This also helps doctor to get a clear report.

CHAPTER 6

IMPLEMENTATION OF MODULES

6.1 Introduction

Module implementations provide different aspects of knowledge and understanding. Here are some of the figures that shows us how the system gets implemented. Other than that, the approach is thoroughly discussed and hence provides a diagrammatic scenario for the modules.

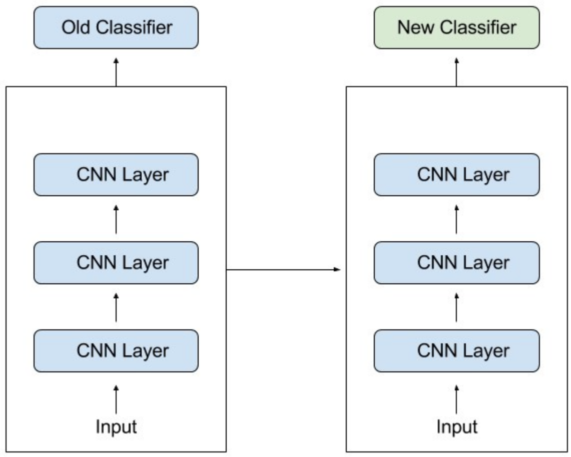


Figure 6.1: A Schematic Representation of how Transfer Learning Works

The new prototype with a diagrammatic representation is shown below. It provides all the problems solutions to the above factors presented. This provides a more closer perspective of how the model works.

The model shown consists of all the things the new system provides.

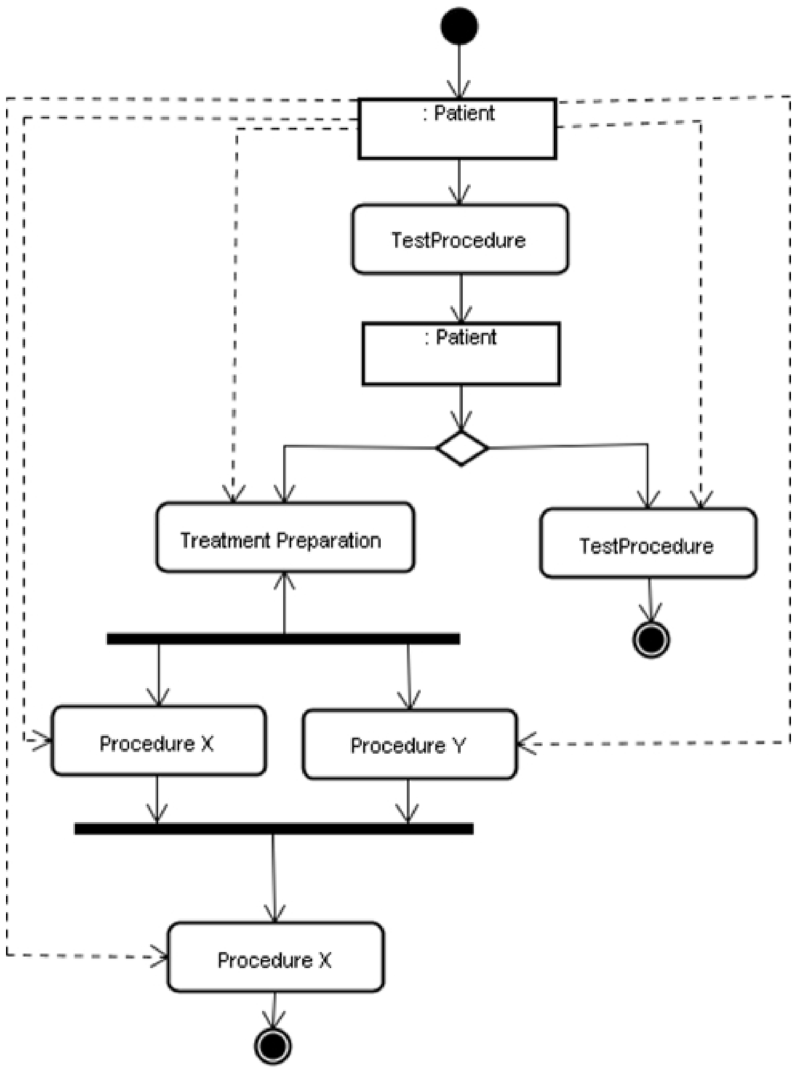


Figure 6.2: A Schematic Representation of an activity of a patient that suffers from pneumonia

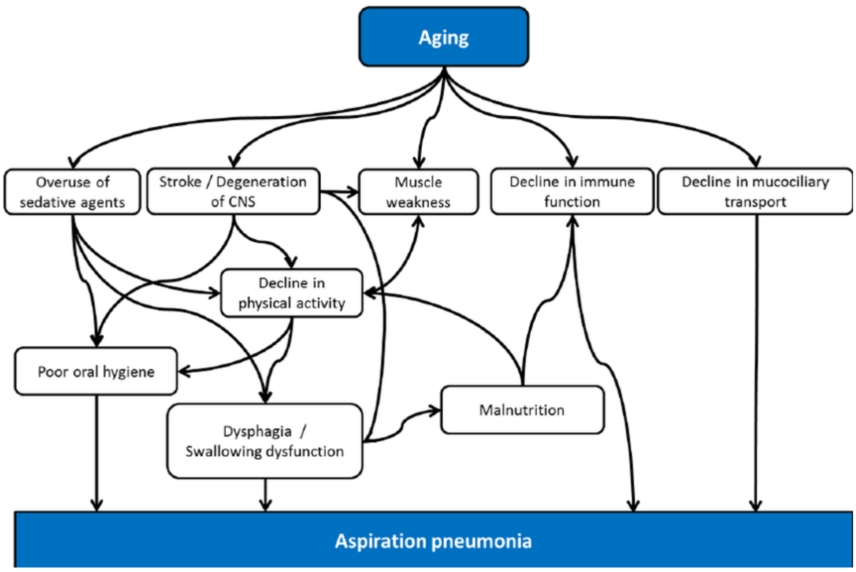


Figure 6.3: Various Risk Factors Associated with Pneumonia

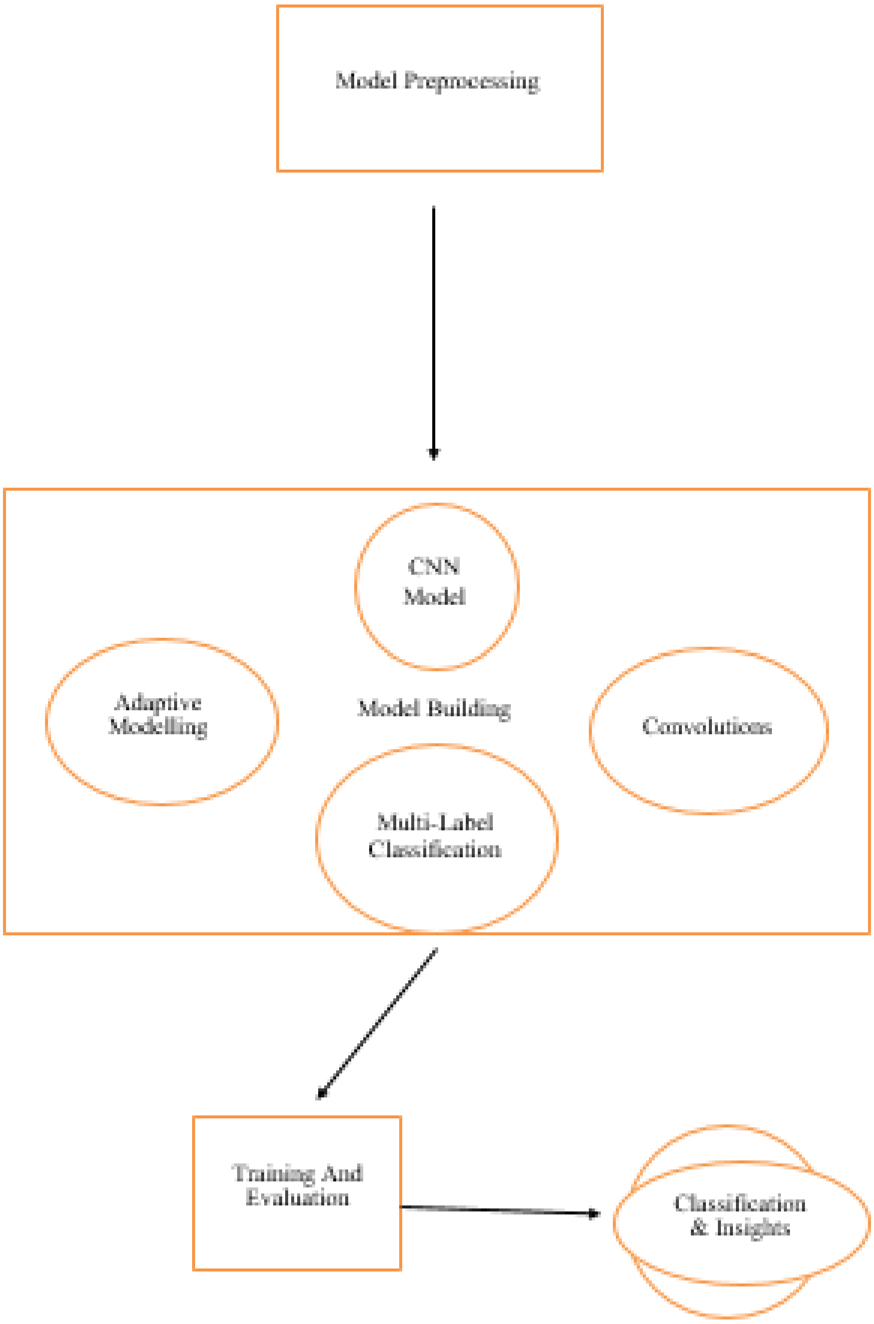


Figure 6.4: Proposed work model

CHAPTER 7

CONCLUSION

Pneumonia detection has often puzzled many earlier with no adaptations in the technology. However, as the growth of technology increases it is nearly seemed possible to attain the operations. Earlier various advancements were not made and hence people wanted to come up with some of the ideations. The important aspect is that we need more and more advancements for health sciences and also adapt the new technology to determine more and more insights and get deep into a new technological world.

The idea above provides a seemingly combined approach of everything that was dealt with. These adaptations require heavy performance on a large scale if the product is sent to the market. The many ideations adopted and recombined to create Machine Learning and other advancements has been a vital part of how a system thinks and behaves. These adaptations provide a modern day solutions to various other problems too.

Pneumonia is mainly now a problem but when comes to solving it in a more faster and easier way, people think to have a prior approach in mind. Hence, a refreshment model and a powerful tool offers seemingly ease of work.

The approach gathers a lot of information which is represented in different datasets. These information often acts as a selector which intern makes the model in a more responsive, reliable and adaptive style.

CHAPTER 8

FUTURE ENHANCEMENT

The system offers the approach which can be used in future for an app development. An app that determines pneumonia while scanning its lungs with help of a camera or providing its CXR image provided by doctors. A main report might be obtained in future for which every patient can diagnose it on its own. A beginning of healthcare carries various opportunities and offers new models and new possibilities.

A different set of function can also be used where person can share reports with the doctor and has different login schemes. Adaptation will be obtained and makes different set of operations such as patientâĂŹs health, vital reports, blood reports, rescanning etc. It can also provide various feedbacks to the patient and improve every set of details using more powerful techniques.A wide range of possibilities are there and hence improving in every aspect is key in this system.

REFERENCES

1. Abhishek Sharma,Daniel Raju,Sutapa Ranjan "Detection of Pneumonia clouds in Chest X- ray using Image processing approachâĂİ-2017

2. Cisneros-Velarde, P., Correa, M., Mayta, H., Anticona, C., Pajuelo, M., Oberhelman, R., ... Castaneda, B. (2016). Automatic pneumonia detection based on ultrasound video analysis. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). -2016

3. Pranav Rajpurkar,Jeremy Irvin,Kaylie Zhu,Brandon Yang,Hershel Mehta,Tony Duan,Daisy Ding,Andrew Y Ng âĂIJCheXNet: Radiologist- Level Pneumonia Detection on Chest X-Rays with Deep LearningâĂİ- December 2017

4. https://medium.com/technologymadeeasy/ the-best-explanation-of- convolutional-neural-networks-on-the-internet- fbb8b1ad5df8

5. https://becominghuman.ai/transfer-learning- retraining-inception-v3-for-custom-image-classification-2820f653c557

6. Pranav Rajpurkar, Jeremy Irvin, et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X- Rays with Deep Learning, https://arxiv.org/pdf/ 1711.05225.pdf 2017.

7. Diederik P. Kingma, Jimmy Lei Ba. Adam: A Method for Stochastic Optimization https:// arxiv.org/ pdf/1412.6980.pdf

8. Laurens van der Maaten, Geoffrey Hinton Visualiz- ing Data using t-SNE 2008 http:// www.cs.toronto. edu/âĞăhinton/absps/tsne.pdf

9. S. Hwang, H.-E. Kim, J. Jeong, and H.-J. Kim, âĂIJA novel approach for tuberculosis screening based on deep convolutional neural networks,âĂİ in SPIE Medi- cal Imaging. International Society for Optics and Photonics, 2016, pp. 97 852WâĂŞ97 852W.

10. https://www.kaggle.com/tentotheminus9/ normal-vs-pneumonia-keras-in-r

11. A. Kumar, Y.-Y. Wang, K.-C. Liu, I.-C. Tsai, C.- C. Huang, and N. Hung, âĂIJDistinguishing normal and pulmonary edema chest x-ray using gabor filter and svm,âĂİ in Bioelectronics and Bioinformatics (ISBB), 2014 IEEE International Symposium on. IEEE, 2014, pp. 1âĂŞ4.

12. Campadelli and E. Casiraghi, âĂIJLung field segmen- tation in digital postero-anterior chest radiographs,âĂİ in International Conference on Pattern Recognition and Image Analysis. Springer, 2005, pp. 736âĂŞ745.

13. https://medium.com/datadriveninvestor/ detecting-pneumonia-with-deep-learning-a- soft-introduction-to-convolutional-neural- networks-b3c6b6c23a88

14. R. Sousa et al, âĂIJComparative performance analysis of machine learning classifiers and dimensionality reduction algorithms in detection of childhood pneumoniaâĂİ, Procedia Computer Science, vol. 18, pp. 2579-2582, 2013.

APPENDIX A

CODE AND RESPECTIVE OUTPUT

A.1 Data Preprocessing & CNN Models(78% Precision Score)

Pneumonia Diagnosis using Lungs' XRays

1]:
```
import pandas as pd
import cv2
import numpy as np
import os
from random import shuffle
from tqdm import tqdm
import scipy
import skimage
from skimage.transform import resize
print(os.listdir("../input/chest_xray/chest_xray"))
```

```
['val', 'train', 'test', '.DS_Store']
```

2]:
```
print(os.listdir("../input/chest_xray/chest_xray/train/"))
```

```
['NORMAL', '.DS_Store', 'PNEUMONIA']
```

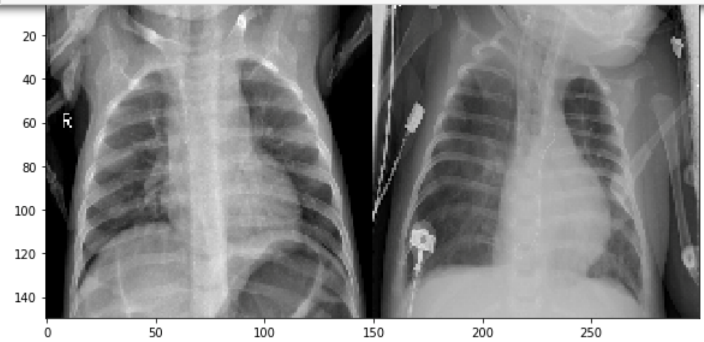
3]:
```
TRAIN_DIR = "../input/chest_xray/chest_xray/train/"
TEST_DIR =  "../input/chest_xray/chest_xray/test/"
```

Preprocessing

4]:
```
def get_label(Dir):
    for nextdir in os.listdir(Dir):
        if not nextdir.startswith('.'):
            if nextdir in ['NORMAL']:
                label = 0
            elif nextdir in ['PNEUMONIA']:
```

Figure A.1: Importing Data

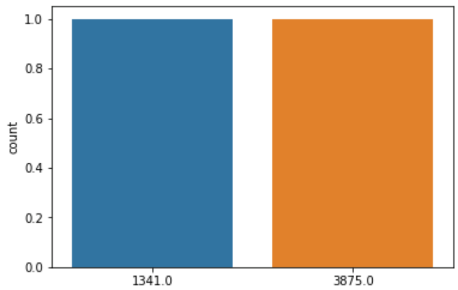
```
In [15]: import matplotlib.pyplot as plt
         def plotter(i):
             imagep1 = cv2.imread(TRAIN_DIR+"PNEUMONIA/"+Pimages[i])
             imagep1 = skimage.transform.resize(imagep1, (150, 150, 3) , mode = 'reflect')
             imagen1 = cv2.imread(TRAIN_DIR+"NORMAL/"+Nimages[i])
             imagen1 = skimage.transform.resize(imagen1, (150, 150, 3))
             pair = np.concatenate((imagen1, imagep1), axis=1)
             print("(Left) - No Pneumonia Vs (Right) - Pneumonia")
             print("-----------------------------------------------------------------------------------
             plt.figure(figsize=(10,5))
             plt.imshow(pair)
             plt.show()
         for i in range(0,5):
             plotter(i)
```




```
(Left) - No Pneumonia Vs (Right) - Pneumonia
--------------------------------------------------------------------------------------------
```

Figure A.2: Data Selection

```
In [16]: import seaborn as sns
         count = y_train.sum(axis = 0)
         sns.countplot(x = count)

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f656978a940>
```



The classes are imbalanced therefore validation accuracy won't be a good metric to analyze the model performance , We will also have to take precision , recall and confusion matrix into account.

```
In [17]: from keras.callbacks import ReduceLROnPlateau , ModelCheckpoint
         lr_reduce = ReduceLROnPlateau(monitor='val_acc', factor=0.1, epsilon=0.0001, patience=1, verbose=1)

/opt/conda/lib/python3.6/site-packages/Keras-2.1.5-py3.6.egg/keras/callbacks.py:919: UserWarning: `epsilon` argument
is deprecated and will be removed, use `min_delta` insted.
```

Callbacks to reduce learning rate timely after monitoring a quantity.

Figure A.3: Splitting the Dataset

In [17]:
```
from keras.callbacks import ReduceLROnPlateau , ModelCheckpoint
lr_reduce = ReduceLROnPlateau(monitor='val_acc', factor=0.1, epsilon=0.0001, patience=1, verbose=1)
```
```
/opt/conda/lib/python3.6/site-packages/Keras-2.1.5-py3.6.egg/keras/callbacks.py:919: UserWarning: `epsilon` argument
is deprecated and will be removed, use `min_delta` insted.
```

Callbacks to reduce learning rate timely after monitoring a quantity.

In [18]:
```
filepath="weights.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')
```

Making checkpoints timely to check and save the best model performance till last and also avoiding further validation accuracy drop due to overfitting.

In [30]:
```
from keras.models import Sequential
from keras.layers import Dense , Activation
from keras.layers import Dropout
from keras.layers import Flatten
from keras.constraints import maxnorm
from keras.optimizers import SGD , RMSprop
from keras.layers import Conv2D , BatchNormalization
from keras.layers import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
K.set_image_dim_ordering('th')
from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasClassifier
```

In [20]:
```
X_train=X_train.reshape(5216,3,150,150)
X_test=X_test.reshape(624,3,150,150)
```

Figure A.4: Importing CNN

Tried different model architectures , the best I could achieve was 83.75 % validation accuracy without any pre-trained CNN models. The architecture is different from the best and could give 83.01 % . But again our main criteria is not accuracy but the precision and recall.

Other Hyperparameters like learning rates, epochs, batch size , no. of filters , activation function have been tuned repeatedly to achieve better results.

]:
```
def swish_activation(x):
    return (K.sigmoid(x) * x)

model = Sequential()
model.add(Conv2D(16, (3, 3), activation='relu', padding="same", input_shape=(3,150,150)))
model.add(Conv2D(16, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3), activation='relu', padding="same", input_shape=(3,150,150)))
model.add(Conv2D(32, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu', padding="same"))
model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(96, (3, 3), dilation_rate=(2, 2), activation='relu', padding="same"))
model.add(Conv2D(96, (3, 3), padding="valid", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), dilation_rate=(2, 2), activation='relu', padding="same"))
model.add(Conv2D(128, (3, 3), padding="valid", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
```

Figure A.5: Defining the Model

```
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_97 (Conv2D)           (None, 16, 150, 150)      448
_________________________________________________________________
conv2d_98 (Conv2D)           (None, 16, 150, 150)      2320
_________________________________________________________________
max_pooling2d_47 (MaxPooling (None, 16, 75, 75)        0
_________________________________________________________________
conv2d_99 (Conv2D)           (None, 32, 75, 75)        4640
_________________________________________________________________
conv2d_100 (Conv2D)          (None, 32, 75, 75)        9248
_________________________________________________________________
max_pooling2d_48 (MaxPooling (None, 32, 37, 37)        0
_________________________________________________________________
conv2d_101 (Conv2D)          (None, 64, 37, 37)        18496
_________________________________________________________________
conv2d_102 (Conv2D)          (None, 64, 37, 37)        36928
_________________________________________________________________
max_pooling2d_49 (MaxPooling (None, 64, 18, 18)        0
_________________________________________________________________
conv2d_103 (Conv2D)          (None, 96, 18, 18)        55392
_________________________________________________________________
conv2d_104 (Conv2D)          (None, 96, 16, 16)        83040
_________________________________________________________________
max_pooling2d_50 (MaxPooling (None, 96, 8, 8)          0
_________________________________________________________________
conv2d_105 (Conv2D)          (None, 128, 8, 8)         110720
_________________________________________________________________
conv2d_106 (Conv2D)          (None, 128, 6, 6)         147584
_________________________________________________________________
max_pooling2d_51 (MaxPooling (None, 128, 3, 3)         0
_________________________________________________________________
```

Figure A.6: Declaring And Calling CNN

```
_________________________________________________________________
max_pooling2d_51 (MaxPooling (None, 128, 3, 3)         0
_________________________________________________________________
flatten_11 (Flatten)         (None, 1152)              0
_________________________________________________________________
dense_21 (Dense)             (None, 64)                73792
_________________________________________________________________
dropout_11 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_22 (Dense)             (None, 2)                 130
=================================================================
Total params: 542,738
Trainable params: 542,738
Non-trainable params: 0
_________________________________________________________________
None
```

```
[61]: batch_size = 256
      epochs = 6
```

```
[62]: history = model.fit(X_train, y_train, validation_data = (X_test , y_test) ,callbacks=[lr_reduce,checkpoint] ,
                epochs=epochs)
```

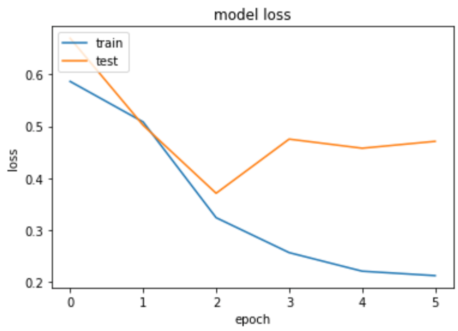
```
Train on 5216 samples, validate on 624 samples
Epoch 1/6
5216/5216 [==============================] - 17s 3ms/step - loss: 0.5862 - acc: 0.7403 - val_loss: 0.6692 - val_acc:
0.6250

Epoch 00001: val_acc did not improve
Epoch 2/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.5083 - acc: 0.7579 - val_loss: 0.5020 - val_acc:
0.7452

Epoch 00002: val_acc did not improve
Epoch 3/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.3242 - acc: 0.8599 - val_loss: 0.3712 - val_acc:
```

Figure A.7: Model Training

```
Epoch 00002: val_acc did not improve
Epoch 3/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.3242 - acc: 0.8599 - val_loss: 0.3712 - val_acc:
0.8301

Epoch 00003: val_acc did not improve
Epoch 4/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.2571 - acc: 0.8904 - val_loss: 0.4753 - val_acc:
0.7724

Epoch 00004: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.

Epoch 00004: val_acc did not improve
Epoch 5/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.2214 - acc: 0.9112 - val_loss: 0.4578 - val_acc:
0.8085

Epoch 00005: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-07.

Epoch 00005: val_acc did not improve
Epoch 6/6
5216/5216 [==============================] - 15s 3ms/step - loss: 0.2128 - acc: 0.9126 - val_loss: 0.4710 - val_acc:
0.8093

Epoch 00006: ReduceLROnPlateau reducing learning rate to 4.999999987376214e-08.

Epoch 00006: val_acc did not improve
```

```
import matplotlib.pyplot as plt

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
```

Figure A.8: Model Prediction



The model will try to overfit itself but rather save it prior going to the next epoch using necessary callbacks.

For better performance use exponential decaying learning rate and specify steps_per_epoch

```
from sklearn.metrics import confusion_matrix
pred = model.predict(X_test)
pred = np.argmax(pred,axis = 1)
y_true = np.argmax(y_test,axis = 1)
```

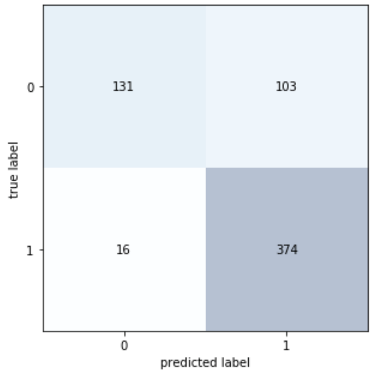
Using mlxtend library for quick implementation of confusion matrix.

```
CM = confusion_matrix(y_true, pred)
from mlxtend.plotting import plot_confusion_matrix
fig, ax = plot_confusion_matrix(conf_mat=CM ,  figsize=(5, 5))
plt.show()
```

Figure A.9: Model Building

Using mlxtend library for quick implementation of confusion matrix.

```
[65]: CM = confusion_matrix(y_true, pred)
      from mlxtend.plotting import plot_confusion_matrix
      fig, ax = plot_confusion_matrix(conf_mat=CM ,  figsize=(5, 5))
      plt.show()
```



Now, a model not good for validation accuracy might be actually good for precision or recall. So better tune according to the metric or your need.

```
[9]: 374 / (374 + 103)
[9]: 0.7840670859538784
```

Figure A.10: Convolution Matrix

Precision is of 78.40 %

Recall is of 95.89 % or approx. 96 % which is quite good.

```
0]: 374 / (374 + 16)
0]: 0.958974358974359
```

Here , recall is most significant quantity even more than accuracy and precision.

Since we are having unequal number of people in both the classes , therefore we can't take accuracy as an alone metric to calculate model efficieny .

precision = True Positive / (True Positive + False Positive)

recall = True Positive / (True Positive + False Negative)

Also precision can't be taken as alone metric and has less significance than recall in this particular dataset because we have to minimize false negative and that is in the denominator and thus finally increasing 'Recall' .

False negative has to be intuitively minimized because falsely diagnosing a patient of pneumonia as not having a pneumonia is a much larger deal than falsely diagnosing a healthy person as a pneumonia patient which is our major concern . That is why we are making this model . To reduce the mistakes done by doctors accidentally . ¶

Figure A.11: Model Accuracy

Reducing the learning rate timely using callbacks in Keras and also checkpointing the model when achieved the best so far quantity that is monitored , in this dataset I am monitoring validation accuracy.

Saving the weights of the best model after checkpointing in transferlearning_weights.hdf5 .

```
filepath="transferlearning_weights.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')
```

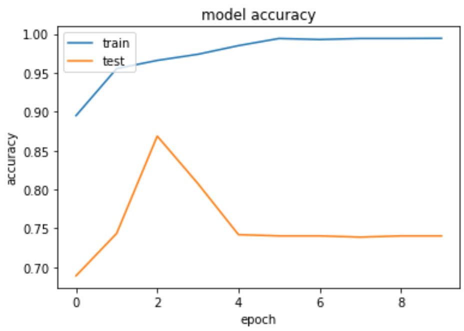
```
from keras.models import Sequential , Model
from keras.layers import Dense , Activation
from keras.layers import Dropout , GlobalAveragePooling2D
from keras.layers import Flatten
from keras.constraints import maxnorm
from keras.optimizers import SGD , RMSprop , Adadelta , Adam
from keras.layers import Conv2D , BatchNormalization
from keras.layers import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
K.set_image_dim_ordering('th')
from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasClassifier
```

```
X_train=X_train.reshape(5216,3,150,150)
X_test=X_test.reshape(624,3,150,150)
```

Importing InceptionV3 from Keras but with no weights. Also define the necessary input shape of the resized images which were resized initially. The default image size is 299 X 299 for InceptionV3.

Figure A.12: Importing Graphs

```
import matplotlib.pyplot as plt

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



model loss

Figure A.13: Validation Curve(For Transfer Learning)

Confusion matrix is very much necessary for the above model because we are having unequal number of people with pneumonia and no pneumonia. In the above dataset we are having more people suffering from pneumonia than normal people. So, as I told in other notebook , accuracy won't be the soul criteria for determining model performance.

```
CM = confusion_matrix(y_true, pred)
from mlxtend.plotting import plot_confusion_matrix
fig, ax = plot_confusion_matrix(conf_mat=CM ,  figsize=(5, 5))
plt.show()
```

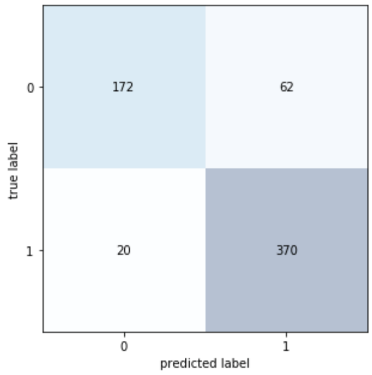


Figure A.14: Predicted Result

```
370 / (370 + 62)
```

```
0.8564814814814815
```

This model even has greater precision than before . Earlier the precision was 78.4 % but now it is having 85.64 % as its precision . Therefore I have minimized false positive in the denominator of the formula . This will even make more sure than before that a person not suffering from pneumonia shouldn't be diagnosed as being suffering from pneumonia. This is what precision is .

```
370 / (370 + 20)
```

```
0.9487179487179487
```

The recall for the above trained model is approx. 95 % which is also quite impressive . Also as I told in other notebook more priority is to given to recall when compare to precision for this dataset.

```
2*0.9487*0.8564 / (0.9487 + 0.8564)
```

```
0.9001902166084982
```

The f1 score is 90.01 % . The harmonic mean of precision and recall.

Figure A.15: Accuracy Description

APPENDIX B

OUTPUTS

B.1 Home page

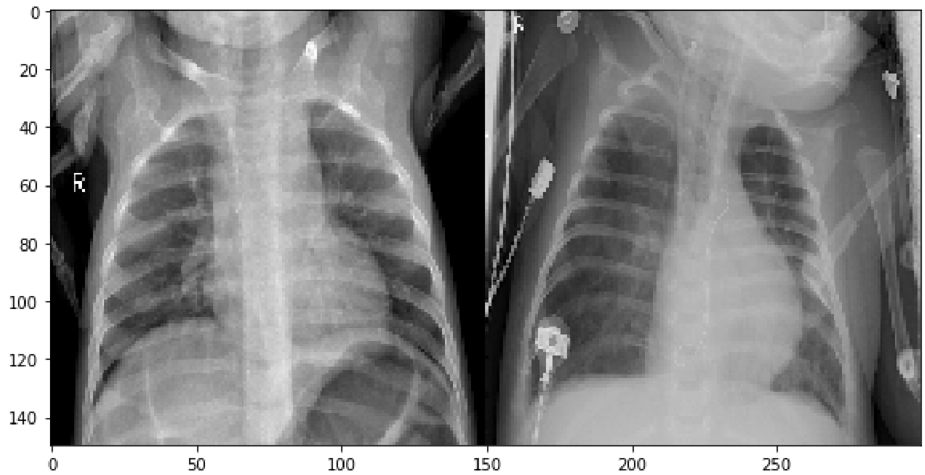


Figure B.1: Left Pneumonia vs Right No Pneumonia

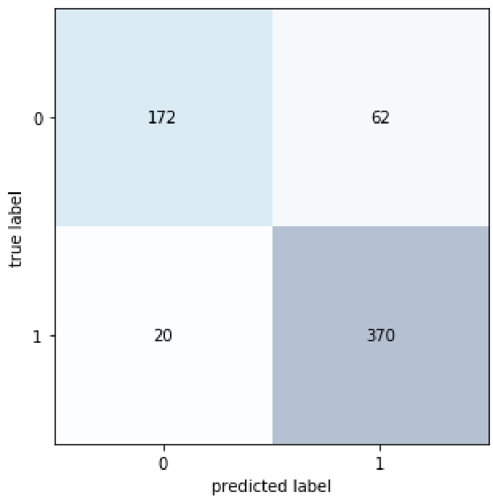


Figure B.2: Convolution Matrix(Overall)

APPENDIX C

ACCEPTANCE/SUBMISSION OF PAPER

ICAIECES'19 - Tentative Technical Sessions Schedule

ICAIECES-2019 <icaieces2019@easychair.org> Wed, Apr 3, 4:50 PM

to me

Dear Hitarth Pandey,

Warm Greetings from SRMIST,

I would like to formally invite you to the 4th International Conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems (ICAIECES'19) hosted by the Department of Software Engineering, School of Computing, SRMIST on 11-04-2019.

We are expecting your presence and cooperation to make this conference a great success.

The Tentative Technical Sessions schedule is accessible now using the below link. Kindly check it. (Final schedule will be updated in the conference website later)

https://drive.google.com/file/d/1nKs5vk-x7Xmu_gJDK2DX2RG3qnQbscR5/view?usp=sharing

PLEASE BRING THE FOLLOWING THINGS WHEN YOU ARE COMING TO THE CONFERENCE.

- Two hard copy of your accepted paper
- College / Organization Identity Card

Feel free to contact us if you are having any queries.

Thank you.

Figure C.1: Acceptance mail

APPENDIX D

CONTRIBUTION OF TEAM MEMBERS

D.1 Hitarth Pandey [RA1511003010433]

D.1.1 Model Training

D.1.2 Model Building

D.1.3 Validation and Evaluation

D.2 Abhishek Suryavanshi [RA1511003010499]

D.2.1 Model Preprocessing

D.2.2 Dataset Selection

D.2.3 Classification