

TUBERCULOSIS DETECTION USING IMAGE PROCESSING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

**LAKKI REDDY VARUN REDDY(38110620)
KOKU VASU (38110622)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH – 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **LAKKI REDDY VARUN REDDY(38110620)** and **KOKU VASU (38110622)** who carried out the project entitled "**TUBERCULOSIS DETECTION USING IMAGE PROCESSING**" under my supervision from November 2021 to March 2022.

Internal Guide

Dr. R. Rajalakshmi, ME., PhD.,

Head of the Department

Dr S. VIGNESHWARI, M.E, PH.D

Dr.L.Lakshmanan M.E., Ph.D. ,

Submitted for Viva voce Examination held on _____

DECLARATION

I hereby declare that the work described in this thesis, entitled “**TUBERCULOSIS DETECTION USING IMAGE PROCESSING**” which is being submitted by me in partial fulfilment for the VII-Semester of the degree of Bachelor of Technology in the Department of **Computer Science and Engineering** to the National Institute of Technology Raipur is the result of investigations carried out by me under the guidance of **Dr.Dilip Singh Sisodia, Assistant Professor, Computer Science & Engineering Department.**

The work is original and has not been submitted for any Degree/Diploma of this or any other Institute/university.

Signature

Name of the Candidate
LAKKI VARUN REDDY(38110620)
KOKU VASU (38110622)

Place:CHENNAI Date:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Mr.Murari Devakannan Kamalesh** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work. I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Tuberculosis is one of the most ancient diseases and still it is one of the top 10 causes of death across the world. Most people who get infected with tuberculosis can be saved with proper treatment and their Life can be saved but due to lack of medical support to detect tuberculosis in most parts of World still mortality rate due to tuberculosis is high. This project helps to detect tuberculosis by using image processing techniques over chest x-rays.

Our objective is to prepare a model which classifies the chest x-ray. This model contains two classes normal and abnormal (infected with TB) we need to classify between these two classes and also to achieve high accuracy while classifying.

In our proposed approach we will be improving the accuracy of model by using deep neural networks to train the model and that model helps us in classifying new chest x-ray given as input to the model thus meeting our objective.

To achieve good accuracy, we need to pre-process the images first we pre-processed the images we took images from both datasets Shenzhen and Montgomery and together there are 800 chest x rays we did augmentation over these images and normalized them followed by giving these pre-processed images as inputs to our models.

In this project we used two models baseline CNN model and pretrained VGG16 model and gave pre-processed images as inputs to these both models and evaluated the models to see which performed better comparing using different performance metrics like accuracy, specificity, sensitivity, precision and f1-score and depicted them using graphs and tables From the above results we made a classification reports for both models.

CONTENT

DESCRIPTION	PAGE NO.
MINOR PROJECT REPORT	1
MINOR PROJECT REPORT	2
DECLARATION	3
DECLARATION	4
CERTIFICATE	5
CERTIFICATE	6
ACKNOWLEDGEMENT	7
ABSTRACT	8
CONTENT	9
LIST OF FIGURES	12
LIST OF TABLES	13
CHAPTER-1	14
INTRODUCTION	14
1.1 OVERVIEW	15
1.2 Importance of Project and objectives	16
1.3 Scope of Project	16
1.4 Motivation	16
1.5 Organisation of Project Report	17
CHAPTER-2	18
LITERATURE REVIEW	18
2.1 OVERVIEW	19
2.2 SUMMARY	20
CHAPTER-3	22
METHODOLOGY	22
3.1 Proposed Method	23
3.2 Datasets used	23

3.3 Tools and technologies used	25
3.3.1 SciKit Learn	25
3.3.2 Pandas	25
3.3.3 NumPy	26
3.3.4 Keras	26
3.3.5 Kaggle Kernel	26
3.4 Pre-processing	27
3.5 About CNN	28
3.5.1 Convolutional layers:	29
3.5.2 Pooling layers:	29
3.5.3 Fully connected layers:	30
3.5.4 Feature Extraction	30
3.6 Architecture of models	30
3.6.1 Architecture of baseline CNN model	30
3.6.2 Architecture of VGG16 model	34
3.7 Flow diagram for models and evaluation of models	35
3.7.1 Flow diagram for baseline CNN model	35
3.7.2 Flow diagram for VGG16 model	36
3.7.3 Training and evaluation of models	38
CHAPTER-4	40
RESULT AND PERFORMANCE EVALUATION	40
4.1 preprocessing over images	41
4.2 Performance evaluation over validation set	41
4.2.1 Baseline CNN	42
4.2.1 Pretrained VGG16	43
4.3 Training and validation accuracy and loss versus epochs	43
4.3.1 Baseline CNN	43
4.3.2 Pretrained VGG16	45
4.4 Comparison between Models for different performance metrics	46
4.4.1 Accuracy	46

4.4.2 Specificity	47
4.4.3 Sensitivity	48
4.4.4 Precision and F1-score	49
4.5 Confusion Matrix	49
4.5.1 CM for baseline CNN model	49
4.5.2 Confusion matrix for VGG16 model	50
4.6 Overall classification report	51
4.6.1 Classification report for baseline CNN model	51
4.6.2 Classification report for VGG16 model	52
4.7 Why VGG16 performed better compared to baseline CNN	53
CHAPTER-5	55
CONCLUSION AND FUTURE SCOPE	55
5.1 Conclusion	56
5.2 Future Scope	57
References and useful links	58

LIST OF FIGURES

DESCRIPTION	PAGE NO.
Figure 1:proposed flow of work	23
Figure 2: Shenzhen dataset images	24
Figure 3 Montgomery dataset images	25
Figure 4 : Images after augmentation	28
Figure 5 CNN architecture (ref: research gate)	29
Figure 6:Baseline CNN architecture	31
Figure 7:layers in baseline CNN model	32
Figure 8:Summary of baseline CNN model	33
Figure 9 Architecture of VGG16	34
Figure 10:flow diagram for baseline CNN model	35
Figure 11:flow of VGG16 model	37
Figure 12:Neural networks depicting dropout	38
Figure 13:Epochs versus loss for baseline CNN	44
Figure 14:epochs versus accuracy for baseline CNN	44
Figure 15:epochs versus loss for VGG16	45
Figure 16:epochs versus accuracy for VGG 16	45

LIST OF TABLES

DESCRIPTION	PAGE NO.
Table 1: Accuracy and loss over validation set	42
Table 2: Accuracy obtained by Models	47
Table 3: Specificity achieved by models	48
Table 4: Sensitivity obtained by models	48
Table 5: Precision and f1-score comparison	49
Table 6: Confusion matrix for Baseline CNN model	50
Table 7: Confusion matrix for VGG16 model	51
Table 8: Classification Report for Baseline CNN mode	52
Table 9: Classification report for VGG16 model	53

CHAPTER-1
INTRODUCTION
N

1.1 OVERVIEW

According to WHO TB (tuberculosis) [1] is one of the top 10 causes of death across the world in 2018 around 10.4 million people fell ill from TB i.e. around 28,500 people per day out of them 1.8 million people have died i.e. around 4500 per day. With proper treatment TB can be cured but most of the people have no access for treatment there aren't many specialists to check whether people are infected or not. Manually checking the chest x-rays and detecting the infection needs radiology experts and there are not many experts compared to number of people getting infected.

Image processing is that the wide used modern field in automating utterly totally different techniques and procedures. Image processing has nice impact in medical sector [2] as a result of as patients unit increasing day by day so treating them manually is very powerful, so this approach is utilized for quick automation. we tend to tend to automatize utterly totally different medical procedures previously and it helped at intervals the designation and treatment of various diseases. Variety of the designed automatic systems unit CT scan processed axial imaging digital analysis associate disease that's transmitted through a medium that's a bacterium referred to as mycobacteria. The chest x rays samples of T.B patients unit taken as a medium for screening. The paper collectively includes discussion and analyzation of the techniques that unit previously designed for T.B detection mistreatment analysis photos and chest x rays.

Major issues involved in tuberculosis control programs [2] are the management of pulmonary tuberculosis and Case-finding. The countries which are affected by epidemics of human immunodeficiency virus infection are facing tuberculosis as a more serious problem. The diagnosis of tuberculosis using accurate methods is one of the crucial steps involved to control the occurrence and prevalence of TB. However, the manual diagnosis of tuberculosis is quite complex these days, so there is no standard method at present.

Hence, we need to develop a solution detects whether the person is infected or not with high accuracy using image processing.

1.2 Importance of Project and objectives

Objective of this project is to detect whether the person is infected with TB or not using the chest x-ray of the person.

- To classify across various chest x-rays with high accuracy.
- With this it is easy to detect whether the person is infected or not without involvement of a specialist.
- Thus, it helps many people to know infection at an early stage and take necessary steps to cure it.
- Deaths caused by this epidemic will be drastically reduced as detection becomes easy.

1.3 Scope of Project

- This project has a lot of scope in medical sector as there are many TB infected patients compared to radiology specialists.
- Requirement to achieve this project is dataset containing CXR of normal and infected images.

1.4 Motivation

- Many people are dying across the world because of TB and most of these can be saved by just giving the proper treatment at right time i.e. at early stages.
- First step to do that would be detecting whether a person is infected or not.
- Thus, many lives across the world can be saved and most of the developing countries are facing the problem of TB thus helps them.

1.5 Organisation of Project Report

Organisation of project is done in the following way, initially we started with introduction part where we explained the overview of project including objective, scope of project and motivation.

In the preceding chapters we explained all the research papers we used as a reference to this project. we went through all the papers and studied various advantages and limitations of these papers and explained the summary of all papers in the below sections. In the next chapters we explained the technologies being used, proposed methodology, results and performance evaluation along with conclusion and future scope followed by reference of the project.

SciKit Learn, Pandas library, NumPy, Keras, Kaggle kernel are the technologies being used in this project.

In the proposed methodology we explained about the datasets being used along with pre- processing techniques and baseline CNN architecture and pretrained VGG16 model architecture and followed by the steps involved in training the models.

In chapter 4 evaluation of project is explained and performance results are depicted using tables and graphs followed by conclusion and future scope of project in chapter 5.

CHAPTER-2
LITERATURE REVIEW

2.1 OVERVIEW

- In a paper [3] published in 2017 at “IEEE International Conference on Signal and Image method Applications (ICSIPA)” detailed study is done and written about the potential technique for communicable disease classification victimization CNN is used which classifies the images into 2 categories, throughout this paper they have used a CNN style with 7 convolutional layers and 3 all connected layers and performance of assorted optimizers are a validation accuracy of 82.09% has been obtained.
- In a paper [4] Deep Learning at Chest radiography published by Paras Lakhani and Baskaran Sundaram both AlexNet pre-trained, AlexNet untrained and GoogleNet pre-trained and GoogleNet untrained CNN networks were used and compared pretrained models have provided a better accuracy compared to untrained models and image augmentation has also improved the accuracy of the models. Untrained models have shown an AUC of 0.88 and with augmentation it has improved slightly. Advantage with deep learning is it provides a better accuracy with higher dimensional datasets like images and pre-trained models have performed better compared to untrained model as starting layers of neural networks are the same for all images like edges and blobs.
- In a paper [5] published in 2018 in NCBI (National centre for biotechnology information) focusing on different technologies and methods that can be used for pre-processing segmentation and processing of images paper reviews on common methods of computer aided detection of chest radiographs based on AI. Usually rib structure is not removed but removing rib structure and clavicle the accuracy can be improved. Traditional algorithms like SVM (support vector machine) and random forest may be better, but deep learning methods are providing better performance compared thus they deep learning methods are becoming mainstream in terms of image processing.
- In paper [6] “Comparing deep learning models for population screening using chest radiography” compared across many models and published the results in this

paper over Shenzhen dataset highest accuracy is obtained by AlexNet an accuracy of 0.82 and over Montgomery an accuracy of 0.725 and remaining models like VGG-16 , VGG-19 ,Xception ,ResNet -50 has given less accuracy compared to AlexNet.

2.2 SUMMARY

S. N O	PAPER TITLE	AUTHORS	YEAR OF ISSUE OF PUBLICATION	conclusion
1.	Computer aided detection in chest radiography based on AI	Chunli Qin, Denim Yao, Yonghong shi, Zhijian shong	2018	In segmentation deep learning models are better compared to rule-based methods and in classification SVM and random forest are traditional DL methods have become mainstream.
2.	Deep learning a potential method for tuberculosis detection using chest radiography	Rahul Hooda, Sanjeev sofat, simranpreet kaur, Ajay Mittal	2017	Used a CNN architecture with 7convolutional layers and 3 fully connected layers given an accuracy of 82%

3.	Deep learning at chest radiography	Paras Lakhani, baskaran sundaram	2017	Deep learning with DCNNs can accurately classify TB at chest radiography with an AUC of 0.97.
4.	Classification of lung diseases by image processing techniques using computed tomography images	Bhuvaneswari, C Aruna, P Loganathan, D	2014	Different techniques used for classification and feature extraction of images are shown using mode like decision trees ,SVM.
5.	Chest pathology detection using deep learning with non-medical training	Yaniv Bar,Idi Diamant,Lior wolf, Hayit Greenspan	2016	In this paper best performance was obtained by using CNN and GIST features and obtained an AUC od 0.87-0.94 for different chest pathologies

CHAPTER-3
METHODOLOG
Y

3.1 Proposed Method

In this project we used two models one is baseline CNN (convolutional neural network) [8] as our model for classification, neural network used contains 12 convolutional layers and 2 fully connected layers and another is pre trained VGG16 model for classification. Figure 1 shows general steps on how we perform the project using different models. Models come in place of classifier and different steps are performed to pre-process the data according to our requirement before processing it.

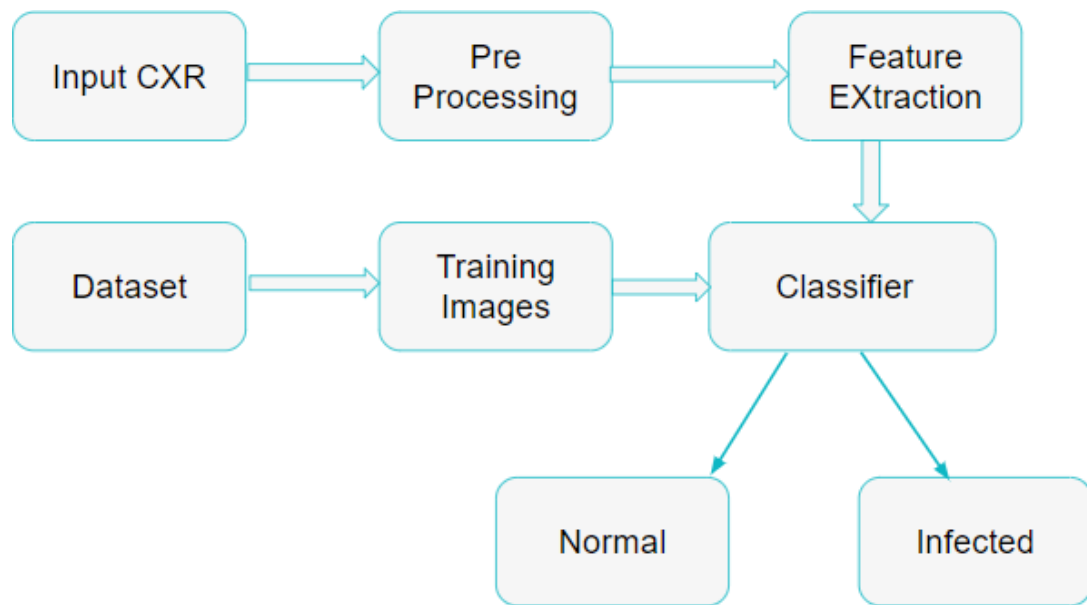


Figure 1:proposed flow of work

3.2 Datasets used

In this project we used two datasets Shenzhen and Montgomery[9].

Montgomery County dataset is collected from the “Department of Health and Human Services, Montgomery County, Maryland, USA”.

The Shenzhen dataset has been collected from “Shenzhen No.3 People’s Hospital”.

The MC dataset includes 138 frontal CXRs out of which 58 cases are infected and remaining cases are with normal i.e. no TB infection.

The Shenzhen dataset includes 662 frontal CXRs out of them 336 cases are infected with tuberculosis, and the remaining cases are normal i.e. without infection of TB.

All names of image files exist in the following pattern: CHNCXR_****_N.png

Where

**** represents a 4-digit number which is an identifier.

N value is either a '0' or '1'.

'0' represents a normal X-ray.

'1' represents an abnormal X-ray.

Each X-ray consists of a text file and a .png file text file contains details about X-ray and .png image is X-ray itself.

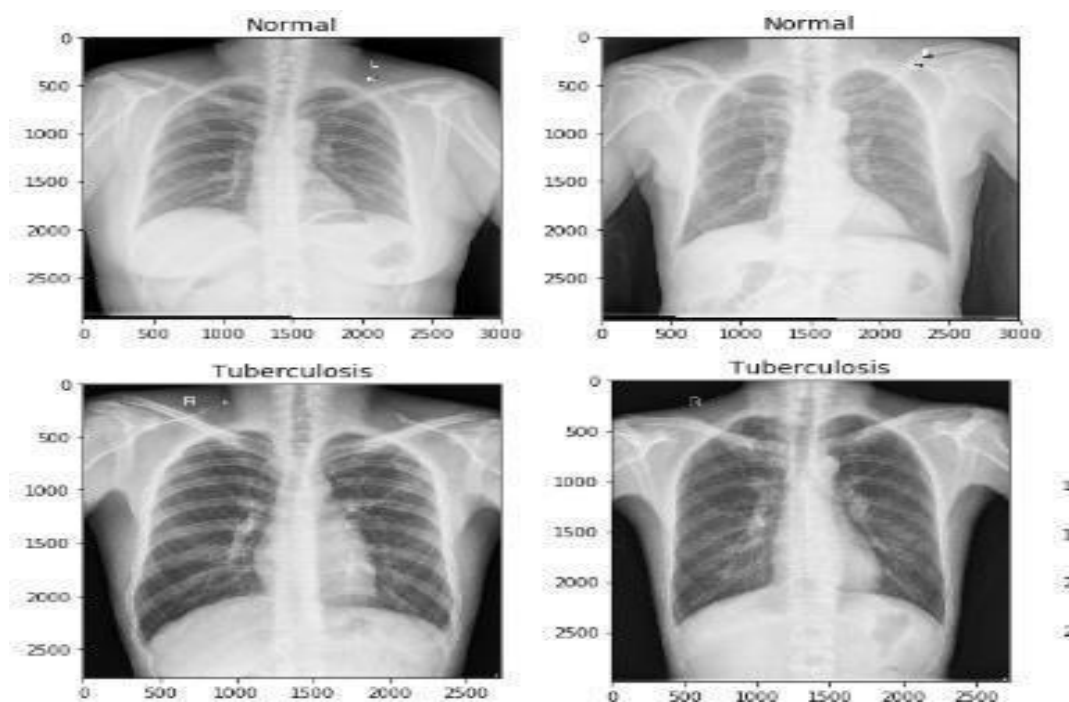


Figure 2: Shenzhen dataset images

Figure 2 displays some random images of Shenzhen dataset which contains a total of 662 chest X-rays.

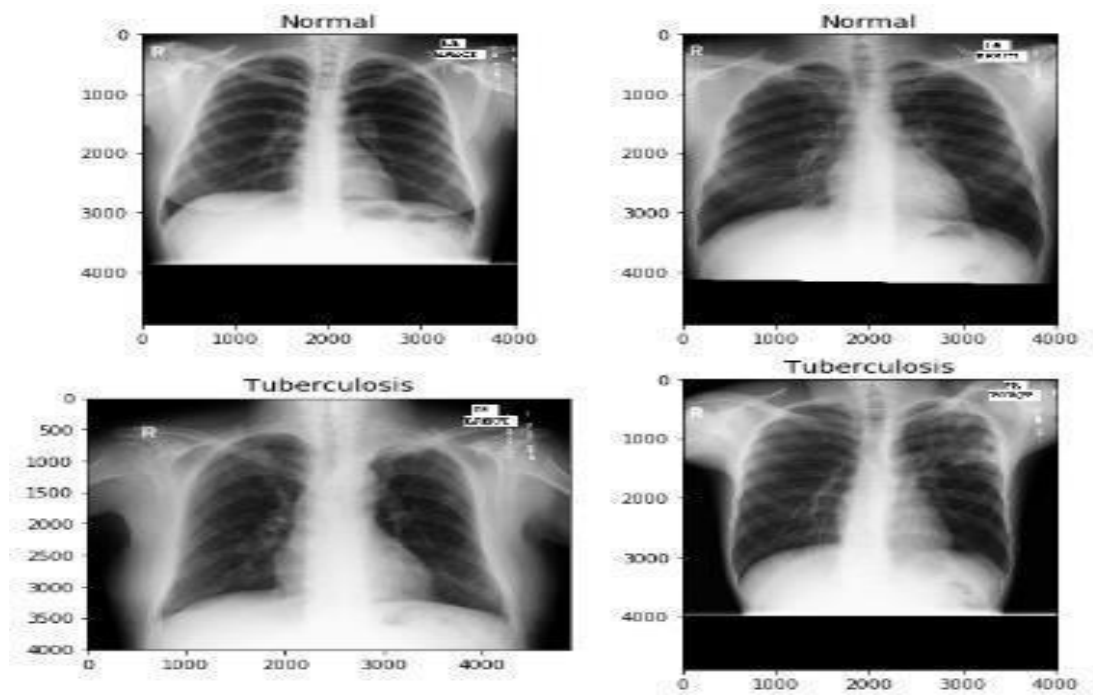


Figure 3 Montgomery dataset images

Figure 3 displays some random images of Montgomery dataset which contains total of 138 frontal Chest X-rays.

3.3 Tools and technologies used

3.3.1 SciKit Learn

SciKit-learn is also a free software package machine learning library for the python artificial language. It consists of assorted algorithms like classification, regression and cluster algorithms moreover as random forests, support vector machines (SVM), gradient boosting etc. It is used to operate with the python numerical and scientific libraries such as NumPy and SciPy.

3.3.2 Pandas

Panda can be treated as a library of data structures and data analysis tools for python. It is also an open source. It also leads to high performance.

Python with pandas can be applied in different fields such as Academic or commercial domain. Using pandas, we can apply the steps in the processing and analysis of data, load, prepare, manipulate, model and analyse.

3.3.3 NumPy

NumPy is a package that is used for scientific computations. It can be used as multi layered container. NumPy can easily generate data of any dimensions and can work on it and it comes bundled with python. It consists of -

- Objects of N - dimensional array
- Complex methods
- Linear algebra, random number and Fourier transform functionalities.

3.3.4 Keras

Keras is a library that is written in python which is used for networking that runs above tensor flow. It is open source. Keras is not responsible for handling low-level computations instead it uses another library called “backend” to do those tasks. Hence keras is an API wrapper for low-level API allowed to run above Tensor Flow. It is responsible for guiding the way models are made, setting up layers, setting up multiple types of input-output models.

- Easy use.
- Predefined library
- Data structures
- Free of cost
- Data Structures
- Productivity and speed

3.3.5 Kaggle Kernel

Kaggle notebook [7] is an open source platform that is used to run jupyter networks in browser. It is recently upgraded to improve compute power and memory. The length of time you can run a process is up to 60 minutes. Kaggle is used to create and edit the code in real time i.e. multiple users can access and write the code at the same time. Kaggle also

used for plotting to represent the data and can be used for data visualization. It is launched by Anaconda Navigator.

3.4 Pre-processing

In pre-processing step, we do some operations on images and enhance them for getting better accuracy [6]. In this project images in both Shenzhen and Montgomery datasets are taken, and they are all reduced to the same sizes for giving them as input to the model. For increasing the accuracy of the model Image data augmentation is performed (Image data augmentation is the process of increasing the number of images available from existing training data images because deep learning models perform better with large amount of data) we used ImageDataGenerator class from keras library to do this with this total image count for training data used became 2040 images. We also normalise the images for getting better contrast i.e. we rescale the ranges of RGB values in images which is between 0-255 to 0-1 we do this by using ImageDataGenerator class and resize method of that class.

Contrast, noise and edge features in images have large consequence on the outcome of the classification of any abnormalities.[10] To get the accurate details from unclear and dull

parts of chest X-ray (CXR) images, chest X-ray images should be improved or heightened in such a way to easily identify the structural information of abnormalities and also to remove any noise in the chest X-ray images so that we can get a better output accuracy in classification. This includes many methods like filtering (removing unnecessary information), noise suppression (removing noise from image), edge sharpening (to highlight the edges), contrast enhancement. In contrast enhancement we improve the brightness range in the image which helps in making image features clearer, thus improving the brightness of the image overall contrast and making the image more accurate and clearer.

Process involved in pre-processing augmentation [3]:

- Rotation range: -10
- Width shift limit for images: - 0.1
- Height shift limit for images: -0.1

- Zoom: -0.1
- Horizontal flip



Figure 4 : Images after augmentation

After applying the augmentation to images some of the images are converted as shown in the figure 4, from the figure we can see that some images got rotated some got shifted by height and width and some got flipped and some got zoomed these are done according to the given ranges for ImageDataGenerator object.

3.5 About CNN

Baseline CNN model we used in this consists of 12 Relu convolutional layers and 2 fully connected layers i.e. dense layers and another model used is pre trained VGG16.

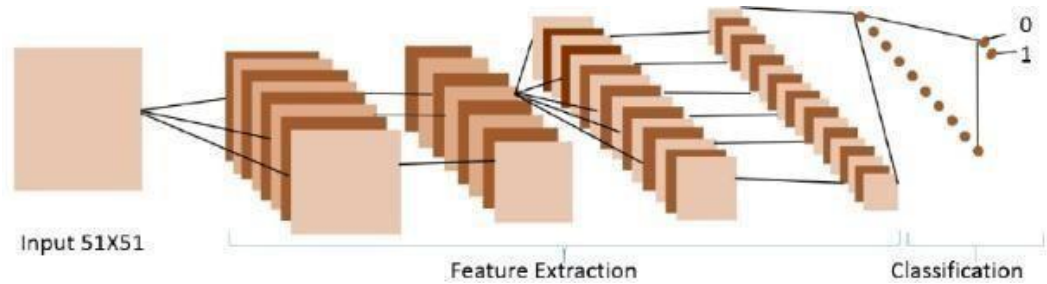


Figure 5 CNN architecture (ref: research gate)

3.5.1 Convolutional layers:

When using a Convolutional Neural Network for processing any data, each and every convolutional layer present in the network should contain the following attributes:

- Convolutional kernels whose depth needs to be equal to that of the image depth, and whose height and width are hyper-parameters. Convolutional layers take the input and produce the output and this output is passed to the next layer. This is like the response given by neuron to a stimulus in visual cortex.
- Input is used to give an image shape it is tensor with form (image width) x (image height) x (number of channels) basing upon channel first or channel last.

3.5.2 Pooling layers:

It works on the output of convolutional Layer and reduce the scale of feature map obtained from convolution layer while not distorting the knowledge. It divides the feature map into some parallelogram regions to perform pooling operation on that. It converges quicker, better, generalisation and sturdy. it's used at the top of convolution layer. There are 3 kinds of pooling.

- Average pooling
- Max pooling
- Min pooling

But the most important thing is that pooling conserves important value of important information.

3.5.3 Fully connected layers:

In these layers (dense layers), every neuron of one layer will be linked to every other neuron in the following layer i.e. if there are n , m neurons in two layers respectively then a total of $n*m$ connections will be available on whole ion fully connected layers. This works on the same principle of multilayer perceptron neural networks. After flattened (decrement of conversion of higher dimensions to lower dimensions) the matrix it is given as input to a fully connected layer these layers then classify the images.

3.5.4 Feature Extraction

Image specimens are known as Features of an image [11]. Sometimes image data can be large to be given as an input to any algorithm for processing, then we do the feature extraction over images to extract some of the required features from images and the work or process on that data thus it is used to transform input data from one mode to another. Feature extraction is used for Transforming the input data of images to the set of required features for further processing. In a conventional neural network model, if we use features extracted from input data instead of full input data then it is expected that the only required information necessary for processing will be extracted from the input data only if the features required for the processing are chosen carefully and no important feature is missed .

3.6 Architecture of models

3.6.1 Architecture of baseline CNN model

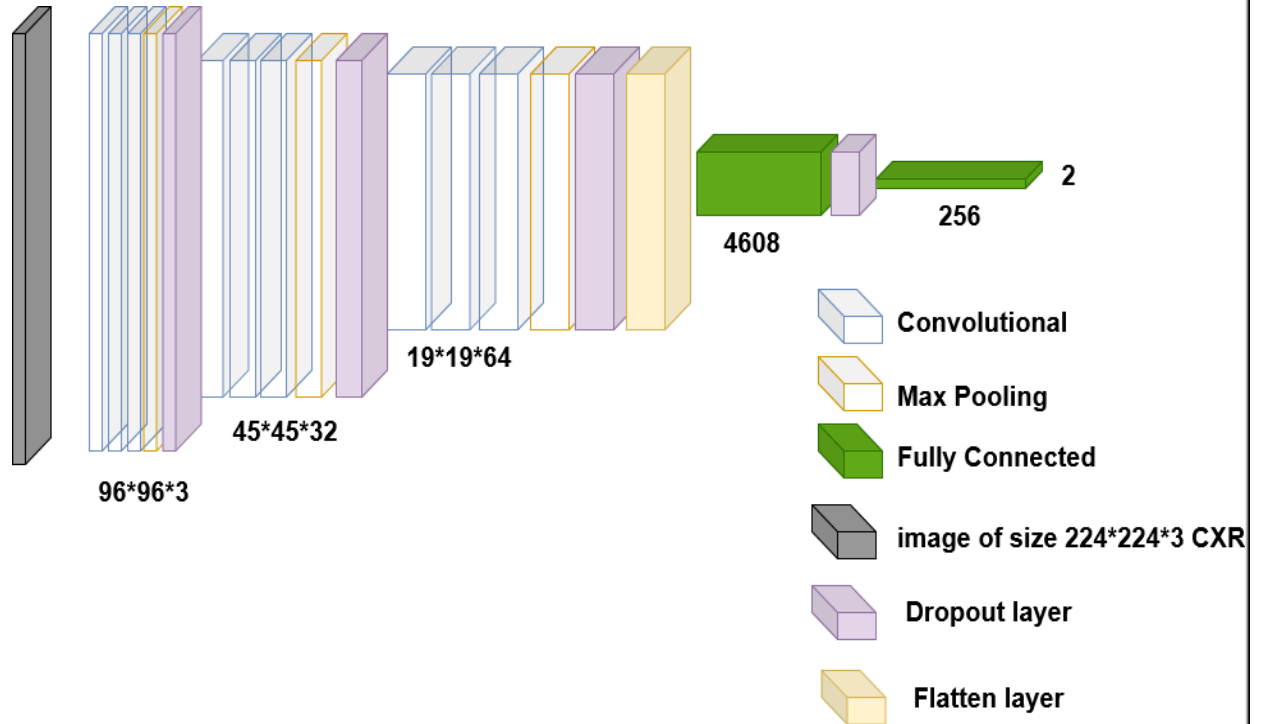


Figure 6: Baseline CNN architecture

Figure 6 shows the architecture of CNN used it contains nine convolutional layers, two fully connected layers or dense layers and three max pooling layers. For this model we used 9 convolutional and 2 dense layers and 3 max pooling layers initially to first convolutional layer input is given as an image of size $96 \times 96 \times 3$ and output from max pooling layer after 3 convolutional layers is $45 \times 45 \times 32$ which is taken as input to next convolutional layer and max pooling layer after that converts image and its output is $19 \times 19 \times 64$ after this a flatten layer is present to reduce the dimensions of image to give it as input to dense layers and after two dense layers image is classified into any one of two classes, we can see that from figure 6 in between some dropout layers are present dropout value we used for this model is 0.3 and is required to stop overfitting of data.

Figure 7 shows the detailed view of each layer in the model and its input and output shapes and number of parameters are displayed and implementation of model is displayed in figure 8 and also model summary is given in figure 8.

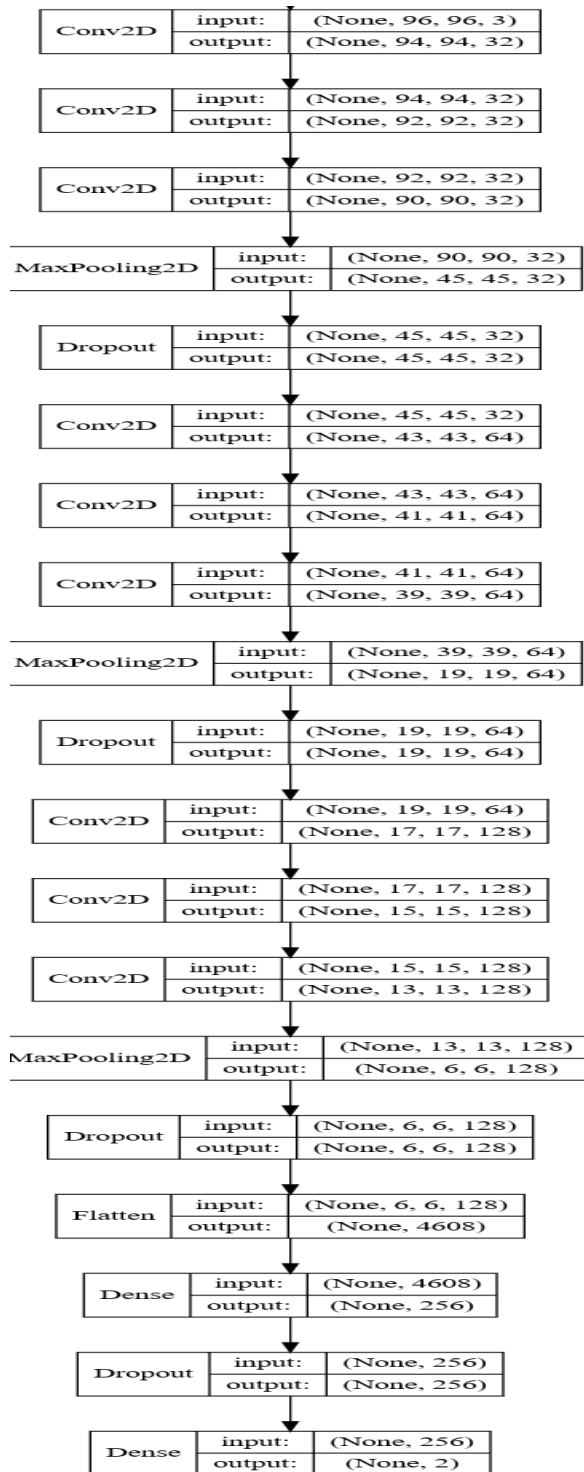


Figure 7: layers in baseline CNN model

In figure 7 all the layers along with input output to each layer and parameters are shown.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 94, 94, 32)	896
conv2d_1 (Conv2D)	(None, 92, 92, 32)	9248
conv2d_2 (Conv2D)	(None, 90, 90, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 45, 45, 32)	0
dropout (Dropout)	(None, 45, 45, 32)	0
conv2d_3 (Conv2D)	(None, 43, 43, 64)	18496
conv2d_4 (Conv2D)	(None, 41, 41, 64)	36928
conv2d_5 (Conv2D)	(None, 39, 39, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 19, 19, 64)	0
dropout_1 (Dropout)	(None, 19, 19, 64)	0
conv2d_6 (Conv2D)	(None, 17, 17, 128)	73856
conv2d_7 (Conv2D)	(None, 15, 15, 128)	147584
conv2d_8 (Conv2D)	(None, 13, 13, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 2)	514
Total params: 1,661,186		
Trainable params: 1,661,186		
Non-trainable params: 0		

Figure 8: Summary of baseline CNN model

Figure 8 shows the implementation code snap of baseline CNN model which is summary of baseline CNN model.

3.6.2 Architecture of VGG16 model

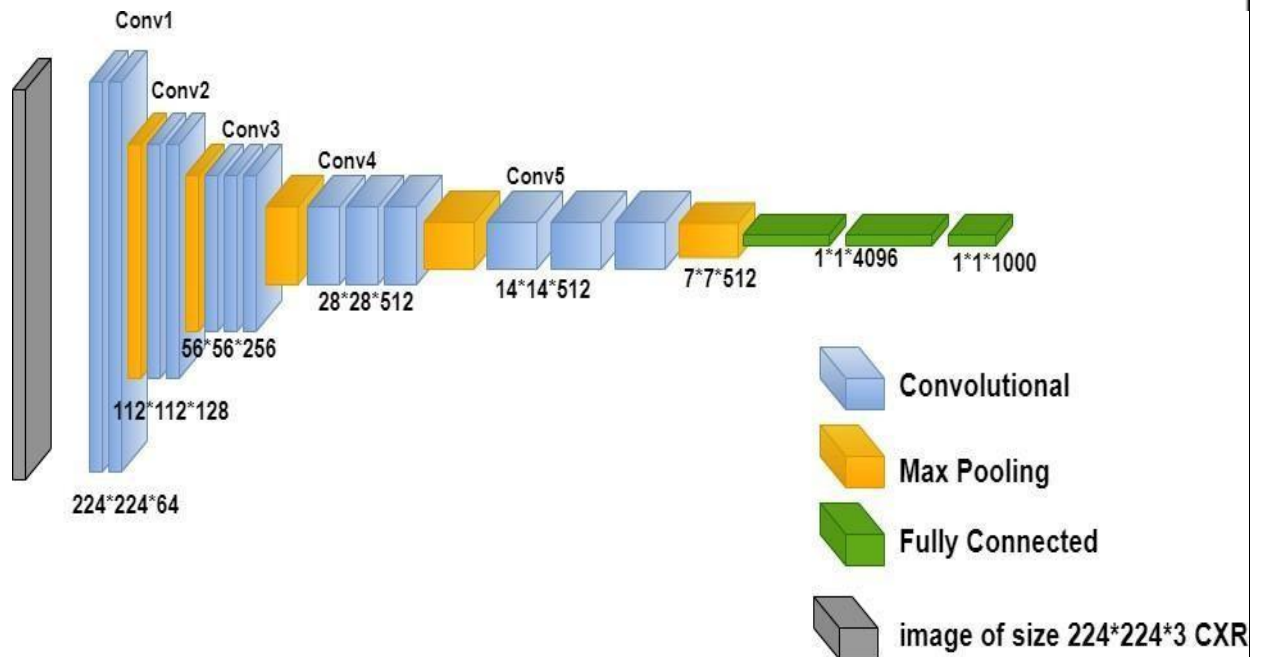


Figure 9 Architecture of VGG16

Figure 9 shows the architecture of VGG16 model. In this project we used pretrained VGG16 original VGG6 model is trained on ImageNet dataset and contains 1000 classes of images, VGG16 contains set of convolutional layers and dense layers and max pooling layers input to VGG16 is taken as an image of shape 224×224 this model contains 4 max pooling layers and 13 convolutional layers and 3 fully connected or dense layers and activation function used is relu activation, The input to cov1 layer is of fixed size 224×224 RGB image. The image is passed through a stack of convolutional layers, where the filters were used with a very small receptive field.

Three fully connected layers follow a stack of convolutional layers the first two have 4096 channels each, the third performs 1000-way classification and thus contains 1000 channels one for each class. The final layer is the soft-max layer, In the model we use we remove this final fully connected layers and replace them by two fully connected layers of 512 neurons and 2 neurons one for each class it needs to classify.

3.7 Flow diagram for models and evaluation of models

In 3.7.1 and 3.7.2 flow diagram for each model is depicted and is explained in detail.

3.7.1 Flow diagram for baseline CNN model

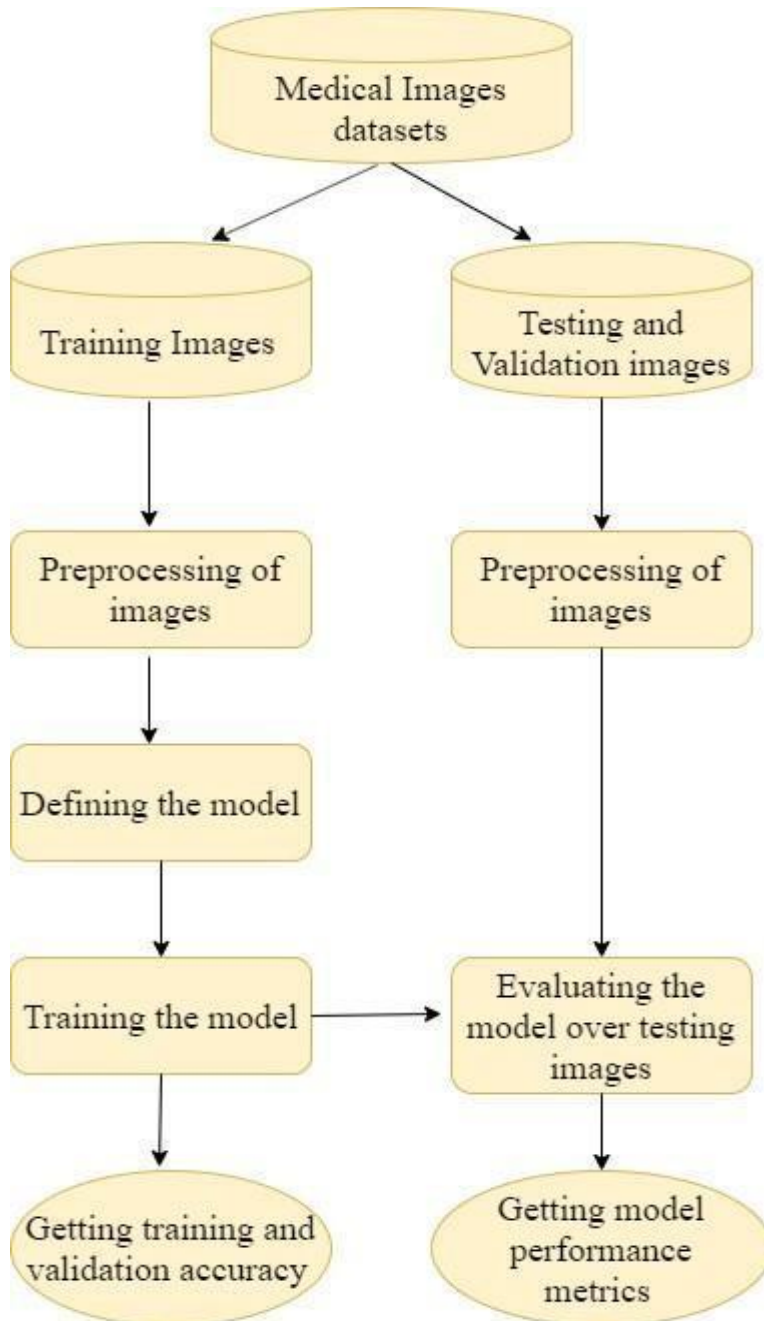


Figure 10:flow diagram for baseline CNN model

Figure 10 depicts the flow of baseline CNN model first we took 800 images combined from Shenzhen and Montgomery datasets out of them 680 images were used for training of model and remaining 120 were used as testing set. We performed preprocessing over training images this includes augmentation and normalization of images this is done using ImageDataGenerator object of keras library then generators are generated for these images and given for training to model defined. Model is then trained over given images and is ready to classify new images given to it, training and validation accuracy are noted and model is ready for evaluation now model is run over testing images and different performance metrics are calculated preprocessing for testing images is also done except augmentation these images are changed to shape $96 \times 96 \times 3$ and are given as input to model for classification.

3.7.2 Flow diagram for VGG16 model

Figure 11 depicts the flow of VGG16 model this is a pretrained model and is already trained over ImageNet dataset the model which is trained on ImageNet dataset is imported and its weights are used now top layer of imported model is removed and our new layers are added and then out 800 images combined from Shenzhen and Montgomery datasets available 680 images were used for training of model and remaining 120 were used as testing set. We performed preprocessing over training images this includes augmentation and normalization of images this is done using ImageDataGenerator object of keras library then generators are generated for these images and given for training to model.

Model is then trained over given images and is ready to classify new images given to it, training and validation accuracy are noted and model is ready for evaluation now model is run over testing images and different performance metrics are calculated preprocessing for testing images is also done except augmentation these images are changed to shape $224 \times 224 \times 3$ and are given as input to model for classification.

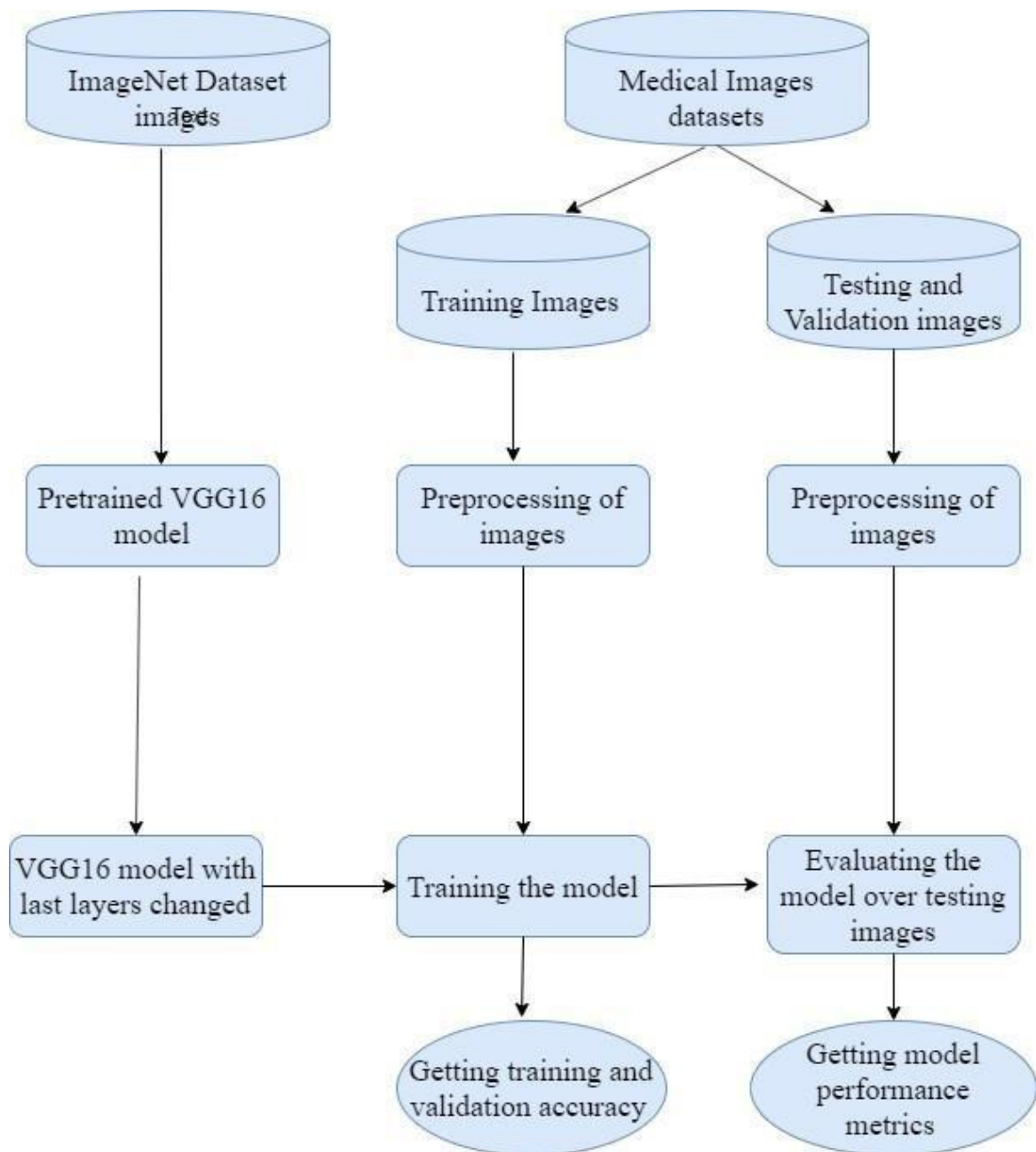


Figure 11:flow of VGG16 model

In figure 11 flow of vgg16 model is depicted

3.7.3 Training and evaluation of models

In training the baseline CNN model [9], we used images of both Shenzhen, and Montgomery datasets and images are augmented so total of 2040 images are used for training the model.

100 epochs are used for training the model we used Adam optimizer for training the model and also used binary cross-entropy loss.

Kernel size used for model is

3*3 Pool size is 2*2

First-filters 32

Second filters 64

Third filters 128

Dropout convolutional 0.3

Dropout dense 0.3

Image size given to the initial layer of model is 94 * 94

We also used max pooling for pooling layer.

We have used a dropout of 0.3 thus overfitting can be avoided

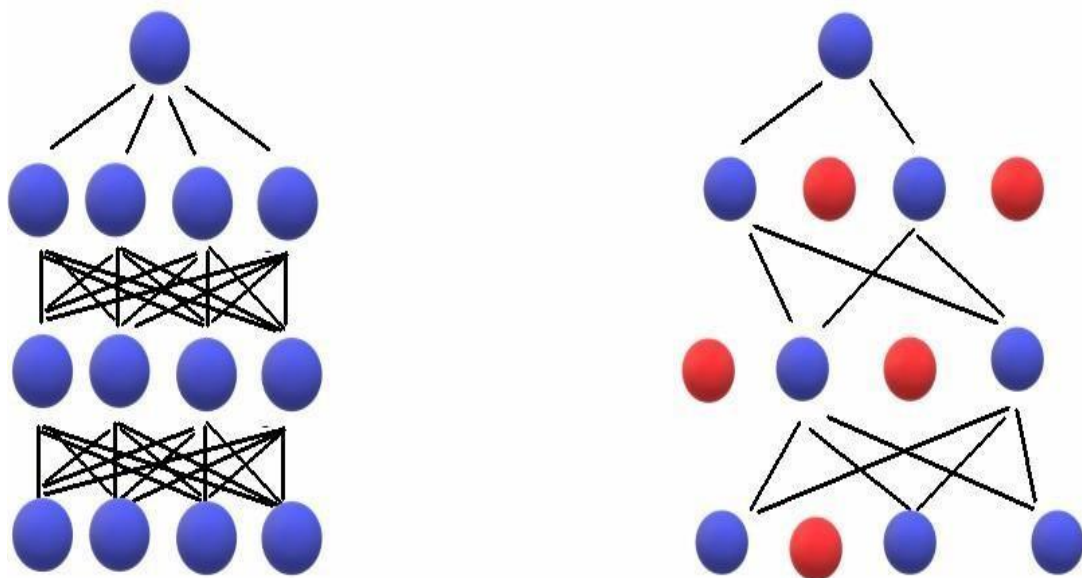


Figure 12:Neural networks depicting dropout

In the figure 12 two diagrams are shown one is network with no dropouts another is a network with dropout, dropouts help in not overfitting [12].

In training VGG16 model we used images of both Shenzhen, and Montgomery datasets and images are augmented so total of 2040 images are used for training the model.

20 epochs are used for training the model we used Adam optimizer for training the model and also used binary cross-entropy loss.

From original VGG16 model [13] which is trained over ImageNet dataset all those weights are imported and used and only last two fully connected layers are changed remaining layers are made non trainable and remains same.

A dropout of 0.6 is used for this model between last layers newly added two layers contain 512 neurons and 2 neurons respectively.

After training the models over training set to check the performance of model we used testing set containing 120 images and used different metrics to display the performance like accuracy, specificity, recall, precision and f1-score using the confusion matrix and for both the model's confusion matrix is displayed below are the formulas for the metrics used in evaluation of model.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{F1score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

r
e
c
a
l

P
r
e
c
i
s
i

o
n
+
r
e
c
a
l
l

TrueNegative

$$\text{Specificity} = \frac{\text{TrueNegative}}{\text{TrueNegative} + \text{FalsePositive}}$$

CHAPTER-4

RESULT AND PERFORMANCE EVALUATION

4.1 preprocessing over images

In this project we used two datasets Shenzhen and Montgomery together a total of 800 images were available out of them 680 were used for training the models and 120 were used for testing and validation, then for better performance of model we do augmentation over training set using keras ImageDataGenerator object and make total image count of 2040 for training set and properties used for ImageDataGenerator object of augmentation are

- Rotation range: -10
- Width shift limit for images: - 0.1
- Height shift limit for images: -0.1
- Zoom: -0.1
- Horizontal flip

We then perform normalization over these images i.e. rescale the ranges of RGB values in images which is between 0-255 to 0-1 we do this by using ImageDataGenerator class and resize method of that class.

Then generators are created from these images and given to model for training the model and then testing the model.

4.2 Performance evaluation over validation set

Validation accuracy is accuracy over validation set that means how accurately model is performing i.e. how many number of predictions are correct compared to total number of predictions made by model and loss is a number which means how bad the model is on single example it is the sum of errors.

In this project we used two datasets Shenzhen and Montgomery together a total of 800 images were available out of them 680 were used for training the models and 120 were used for testing and validation after running both models (Baseline CNN and pretrained VGG16 models their architecture is described in detail in chapter 3) on validation data we got and accuracy and loss for both models as described in table 1 Baseline CNN model architecture is described in chapter 3 and it is run for 100 epochs and to compile the

model we used Adam optimizer with a learning rate of 0.0001 along with loss function as binary cross-entropy and VGG16 is imported and top layer of model is removed and two dense layers are added to the original VGG16 model which is trained over ImageNet we used a dropout of 0.5 between both fully connected layers and all the weights for inner layers are used the same weights imported from model run over ImageNet and they are set for trainable as false and this model is run over training images for 20 epochs and we used Adam optimizer for this model to compile with a learning rate of 0.0001 and a loss function as binary cross-entropy. After running Baseline CNN got a validation accuracy and loss of 0.84 and 0.41 and VGG16 got an accuracy and loss of 0.86 and 0.23.

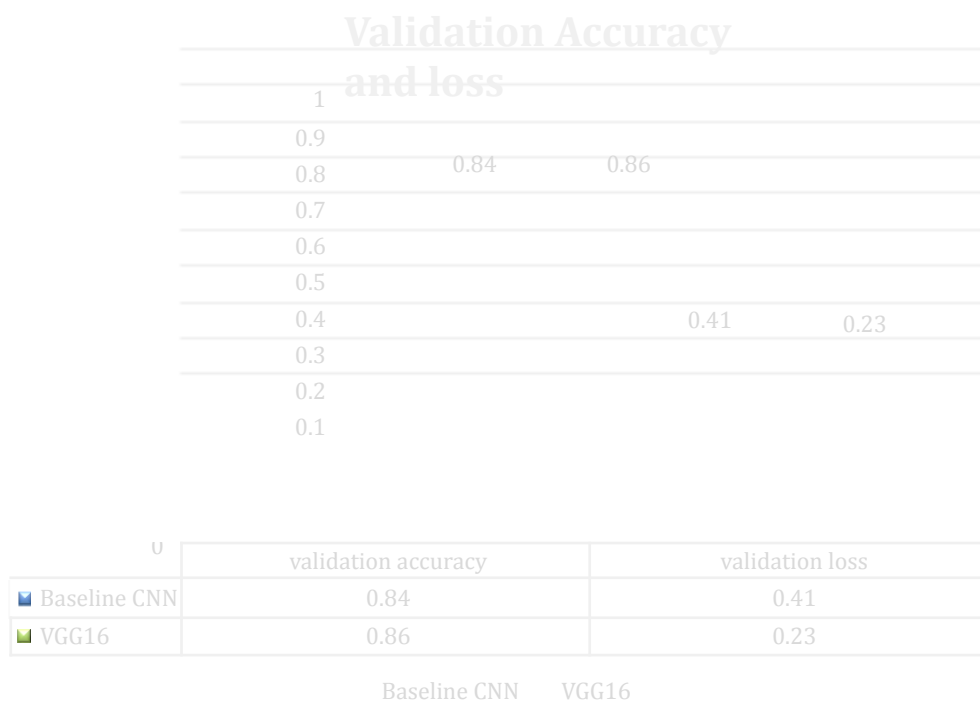


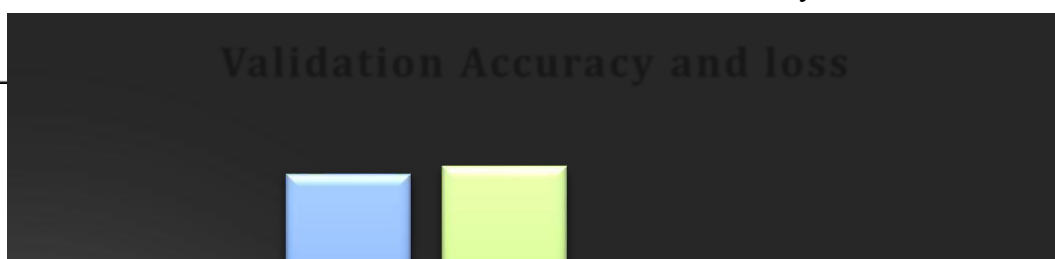
Table 1: Accuracy and loss over validation set

In the table1 both accuracy and loss over validation set for both the models is depicted and validation accuracy and loss obtained by both the models is given below.

4.2.1 Baseline CNN

For this model we trained over augmented images and prediction over validation set got an accuracy of 84%.

For evaluation over validation set we used both validation accuracy and validation loss.



4.2.1 Pretrained VGG16

For this model we trained over augmented images and prediction over validation set got an accuracy of 86%.

For evaluation over validation set we used both validation accuracy and validation loss.

4.3 Training and validation accuracy and loss versus epochs

Baseline CNN model used in this project is run for 100 epochs and Pretrained VGG16 is run for 20 epochs and for each epoch training loss and accuracy and validation loss and accuracy is calculated and is depicted in the form of graphs as shown below in 5.2.1 and

5.2.2. Both the graphs must proceed in same direction i.e. either increasing or decreasing they cannot proceed in opposite ways because that represents overfitting, Number of epochs means the number of passes the model must pass through training dataset and accuracy is the number of correct hits or predictions to total number of predictions and loss is a number how bad model is on single example below for both the models used graphs are plotted for accuracy and loss and is depicted.

4.3.1 Baseline CNN

Figure 13 shows the graph between epochs and validation loss we can see that loss is clearly decreasing as number of epochs increasing and less loss represents that our model predictions are good and our model is performing well.

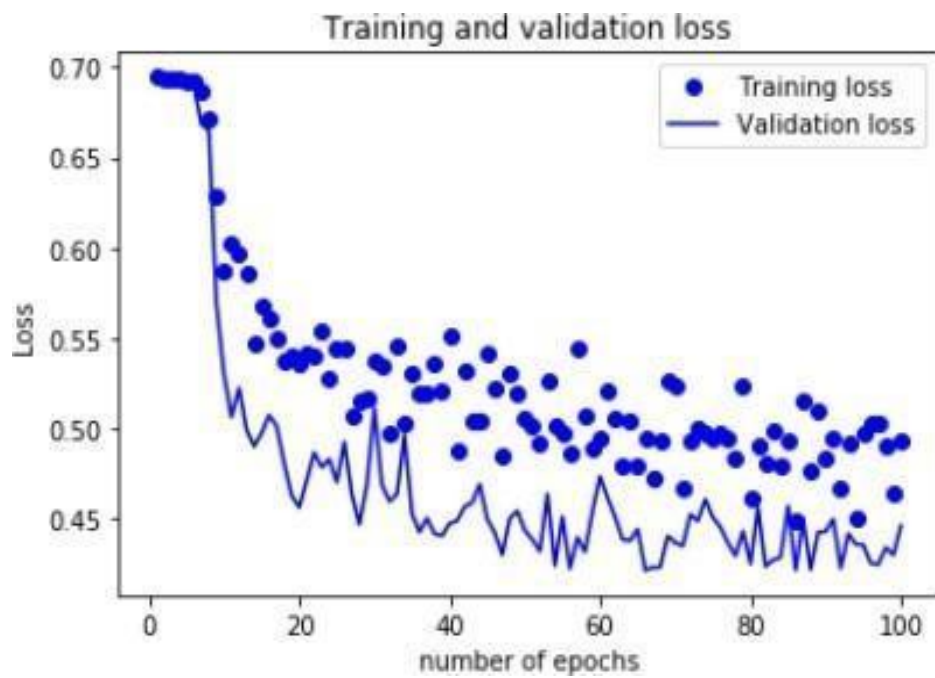


Figure 13: Epochs versus loss for baseline CNN

Figure 14 shows the graph between validation accuracy and number of epochs and we can see that the graph is growing with number of epochs which is a good sign and the accuracy given by the model is around 84%.

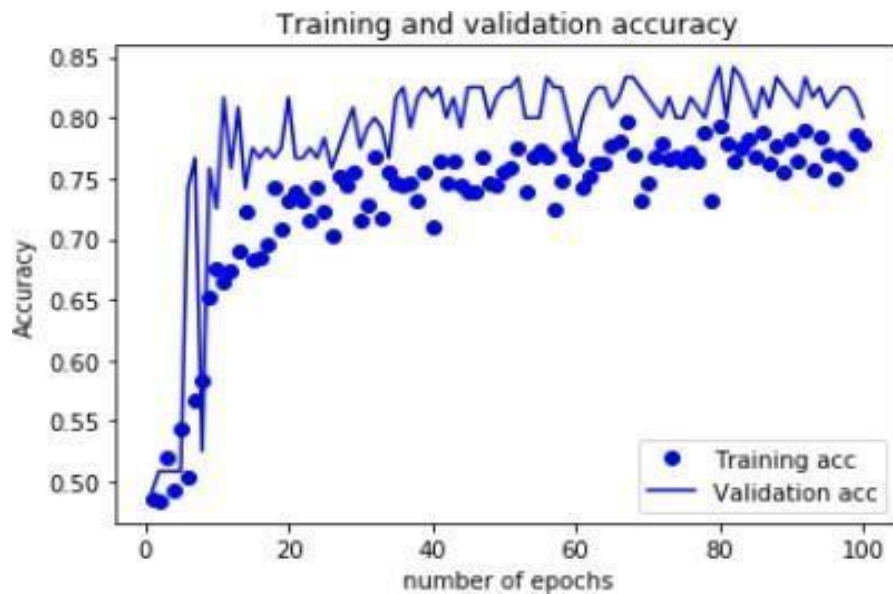


Figure 14: epochs versus accuracy for baseline CNN

4.3.2 Pretrained VGG16

Figure 15 shows the graph between epochs and validation loss we can see that loss is clearly decreasing as number of epochs increasing less loss means that our model is performing good.

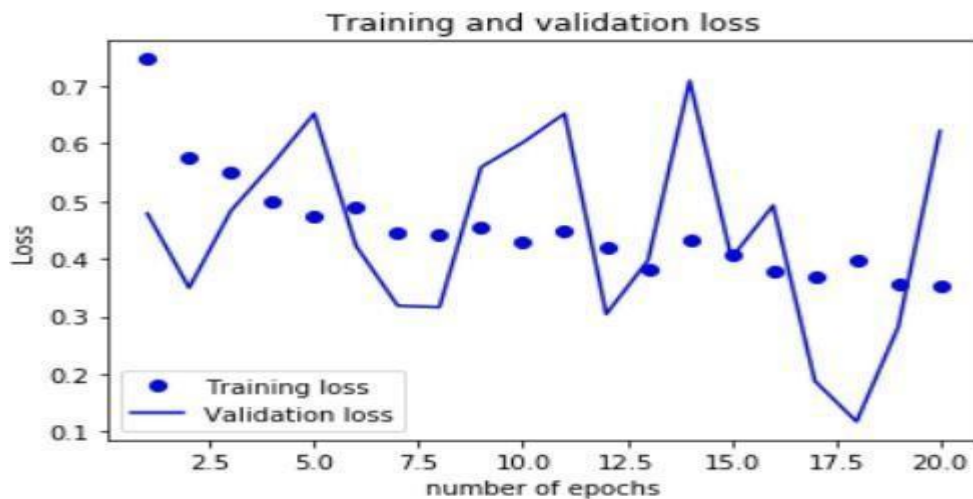


Figure 15:epochs versus loss for VGG16

Figure 15 shows the graph between epochs and loss for training and validation set shown by model VGG16.

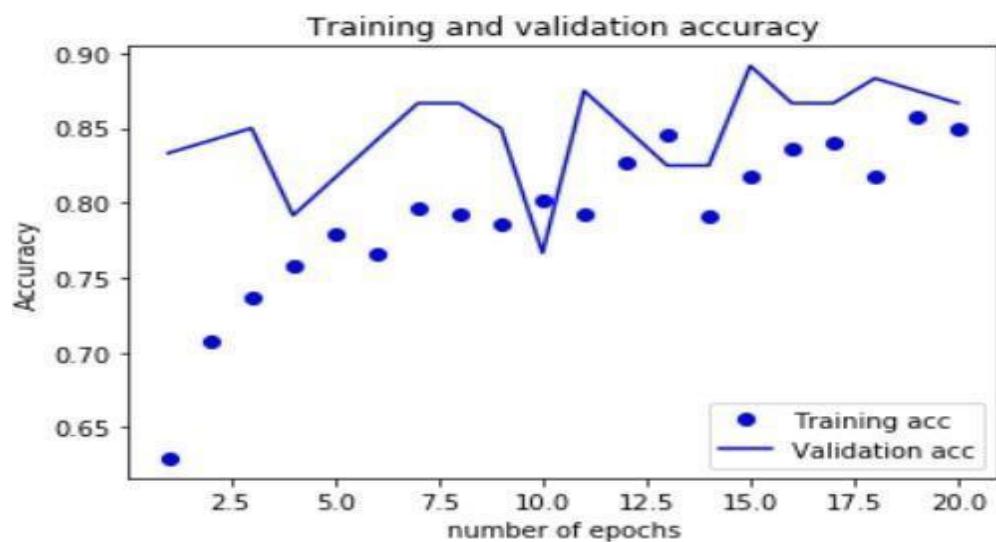


Figure 16:epochs versus accuracy for VGG16

Figure 16 shows the graph between validation accuracy [14] and number of epochs and we can see that the graph is growing with number of epochs which is a good sign and the accuracy given by the model is around 86%.

4.4 Comparison between Models for different performance metrics

Performance metrics are the measure of the performance [15] provided by the image classifiers or any machine or deep learning models some of them are calculated and depicted in the below graphs. We used two models in this project to achieve our goal and their performances are compared their performance are compared over testing set of 120 images we totally had 800 images from both datasets combined (detailed description about dataset used is given in chapter 3.1) out of them 120 are used for testing purpose. Baseline CNN model architecture is described in chapter 3 and it is run for 100 epochs and to compile the model we used Adam optimizer with a learning rate of 0.0001 along with loss function as binary cross-entropy and VGG16 is imported and top layer of model is removed and two dense layers are added to the original VGG16 model which is trained over ImageNet we used a dropout of 0.5 between both fully connected layers and all the weights for inner layers are used the same weights imported from model run over ImageNet and they are set for trainable as false and this model is run over training images for 20 epochs and we used Adam optimizer for this model to compile with a learning rate of 0.0001 and a loss function as binary cross-entropy. After training the model over training set, we run the model over testing images of 120 by using model. Fit-evaluator function then models predicted the image belongs to which classifies and classifies accordingly and they are compared to true values and some performance metrics are calculated and plotted below in the graphs.

4.4.1 Accuracy

Accuracy is the proportion of correct estimates to the total number of estimates. Accuracy is a very important performance evaluator and is used to evaluate many model's performance, accuracy is total number of predictions that are correct to the total number of predictions made by the model.

We used two models in our project and on training the both models Baseline CNN and VGG16 over the testing dataset of 120 images we got accuracy for each models as 81% and 90% VGG16 got better accuracy compared to Baseline CNN and the accuracy values are depicted in the table 2.

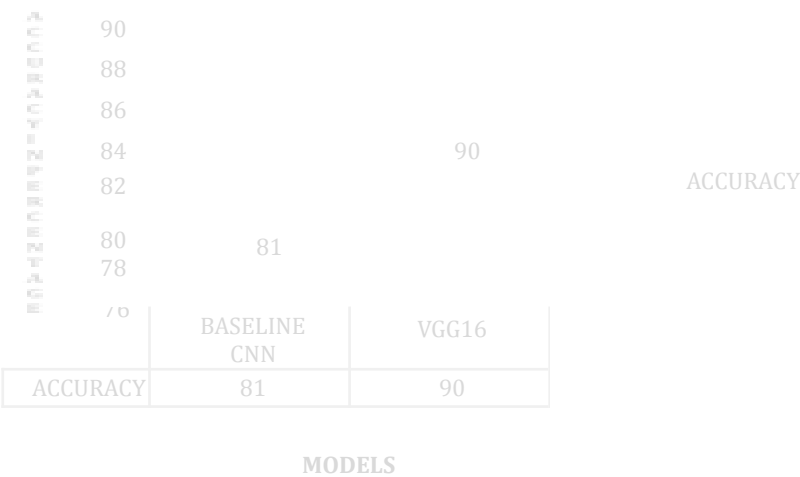


Table 2: Accuracy obtained by Models

Table 2 depicts the accuracy given by the models over the testing images baseline CNN has given an accuracy of 81% whereas VGG16 gives an accuracy of 90%.

4.4.2 Specificity

Specificity is the ratio of true negatives to the sum of true negatives and false positives, specificity is calculated for both models. The confusion matrix which is depicted in the table 3. The specificity values for Baseline CNN and VGG16 are 0.81 and 0.90 respectively.

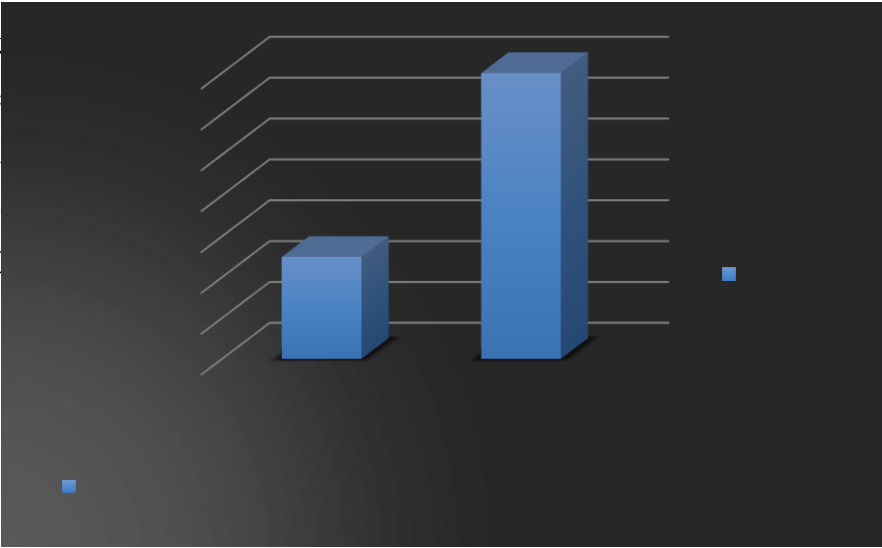




Table 3: Specificity achieved by models

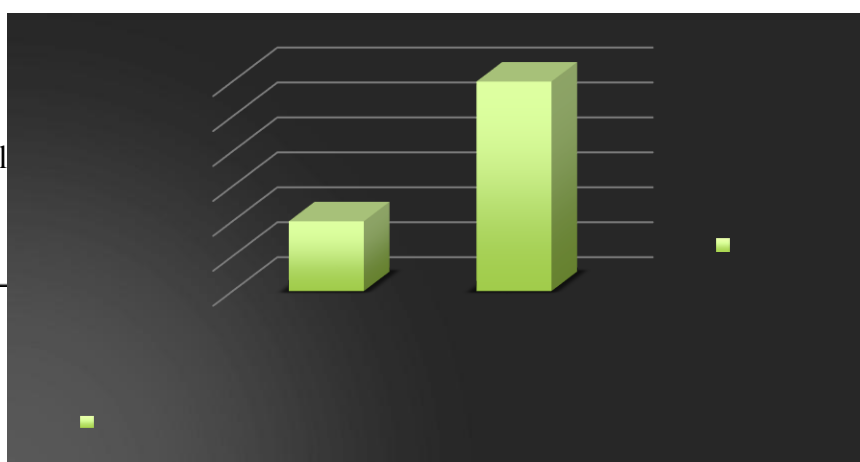
Table 3 depicts the specificity obtained by models

4.4.3 Sensitivity

Sensitivity is the fraction of actual positives which got predicted as the positives, sensitivity is calculated for both the models Baseline CNN and VGG16 from obtained confusion matrix which is explained and depicted for both models in 5.4. Baseline CNN model got a sensitivity of 0.871 and VGG16 model got a specificity of 0.877 which is depicted clearly in table 4.



Table



4.4.4 Precision and F1-score

Precision means what fraction of predicted positives are actually correct and f1-score is HM(harmonic-mean) of recall which is nothing but sensitivity while calculating over confusion matrix and precision for both the models Baseline CNN and VGG16 whose architectures are described in detail in chapter 3 these models when run over training dataset of 120 images we obtained with a precision and f1-score of 0.81 precision and

0.81 f1-score and 0.87 precision and 0.87 f1-score for both models respectively these values are clearly depicted using a bar graph and is shown in table 5.

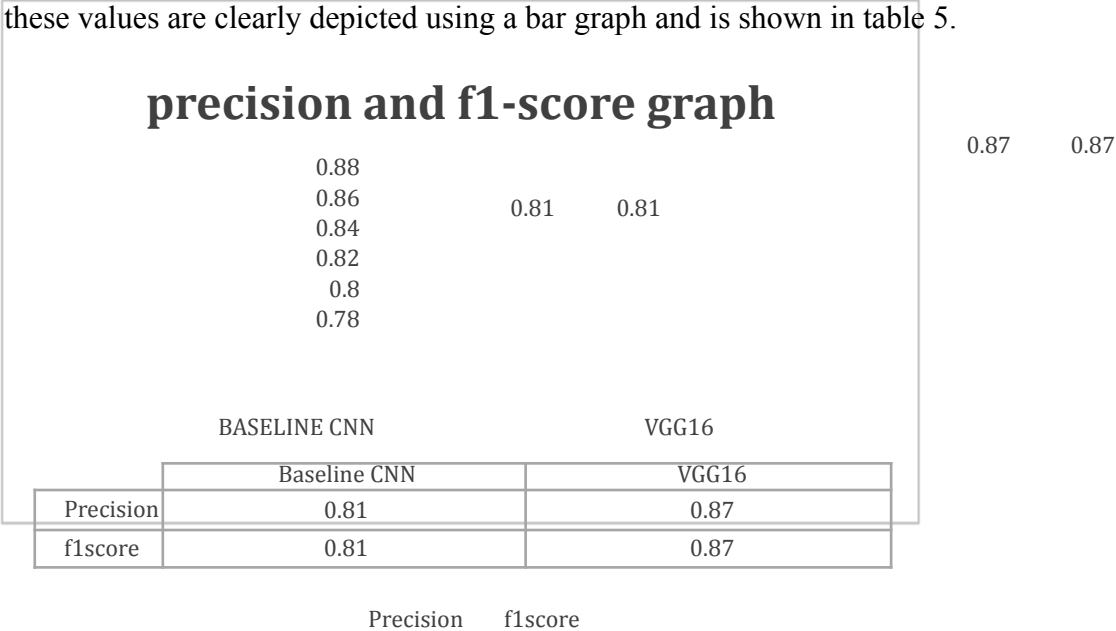


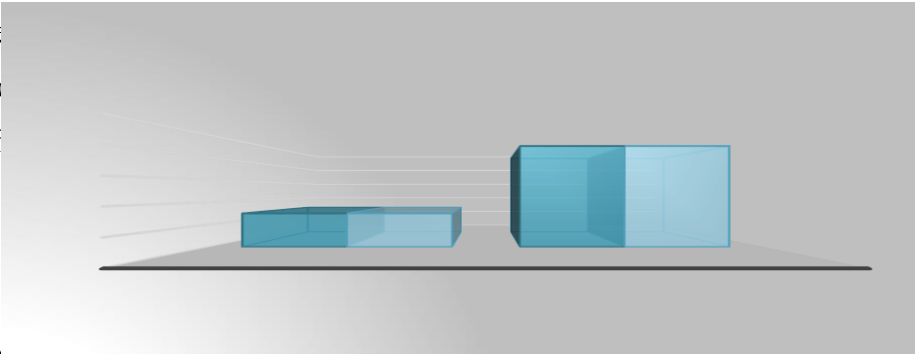
Table 5: Precision and f1-score comparison

Table 5 depicts precision and f1-score for both the models

4.5 Confusion Matrix

Confusion matrix can also be used as a performance evaluator for classifier models this depicts the performance of classification model over testing images when both the models Base predictions are perf conf ues and

4.5.



Training set for our model consists of 120 images and prediction over trained images has given an accuracy of around 81%.

After predicting those 120 images of test set a confusion matrix is drawn and below is a diagram of the confusion matrix predicted by the model over test set.

We can see from table 6 which is confusion matrix for Baseline CNN that true negatives and true positives are more 54 and 47 respectively and false negatives and false positives are very less 7 and 12 respectively out of 120 images which shows our model is performing well.

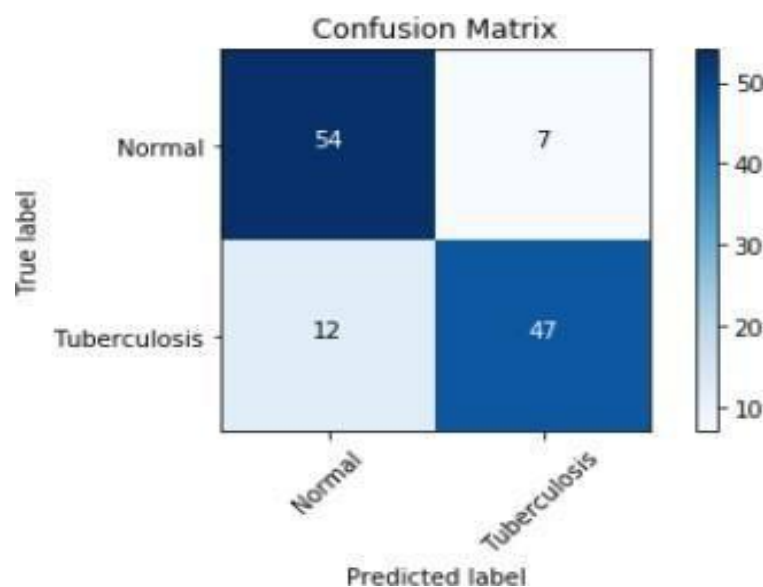


Table 6: Confusion matrix for Baseline CNN model

Table 6 shows the confusion matrix for predictions made by baseline CNN model over test set of 120 testing images.

4.5.2 Confusion matrix for VGG16 model

Training set for our model consists of 120 images and prediction over trained images has given an accuracy of around 87%.

After predicting those 120 images of test set a confusion matrix is drawn and below is a diagram of the confusion matrix predicted by the model over test set.

We can see from table 7 which is confusion matrix for Pretrained VGG16 model that true negatives and true positives are more 54 and 50 respectively and false negatives and false

positives are very less 7 and 9 respectively out of 120 images which shows our model is performing well.

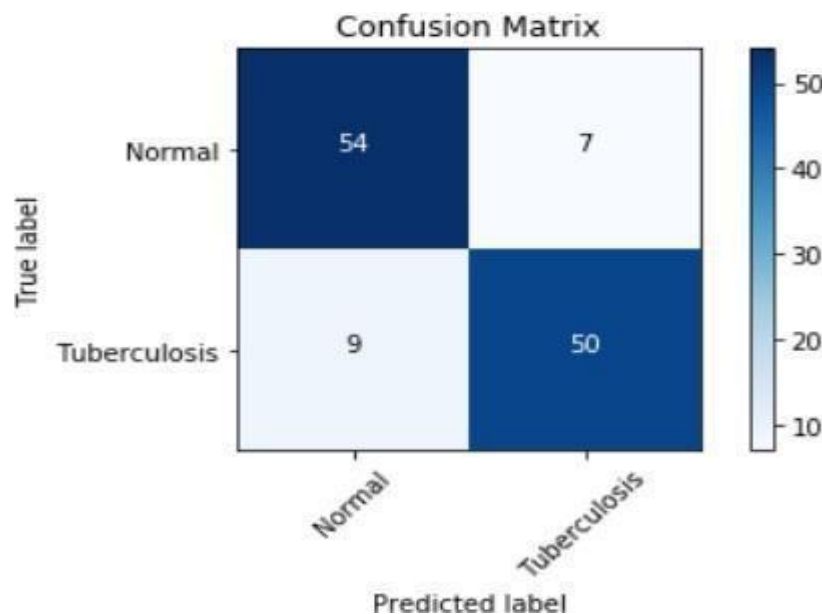


Table 7: Confusion matrix for VGG16 model

Table 7 shows the confusion matrix for predictions made by VGG16 model over 120 testing images

4.6 Overall classification report

It contains precision, recall and f1 score as metrics

Recall = Given a class, will the classifier be able to detect it?

Precision = Given a class prediction from a classifier, how likely is it to be correct?

F1 Score = The HM(harmonic mean) of sensitivity and precision. Essentially, it punishes extreme values.

4.6.1 Classification report for baseline CNN model

In table 8 a detailed report for model Baseline CNN whose architecture is described clearly in chapter 3 recall, precision and f1-score metrics are calculated for this model and depicted in table8 these metrics are calculated by evaluating model over a testing images

of 120 for evaluating a batch size of 10 is taken and model is evaluated over testing set we can see that all the metrics total is 0.81 from table 8.

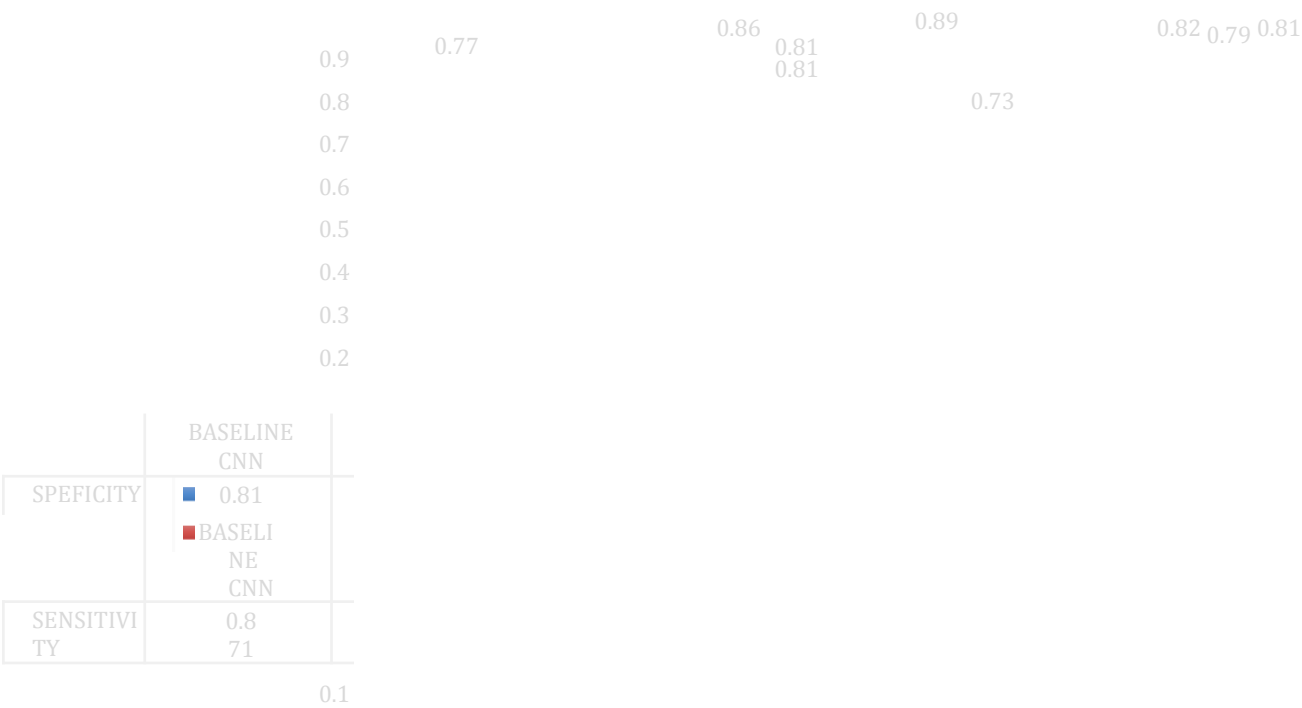
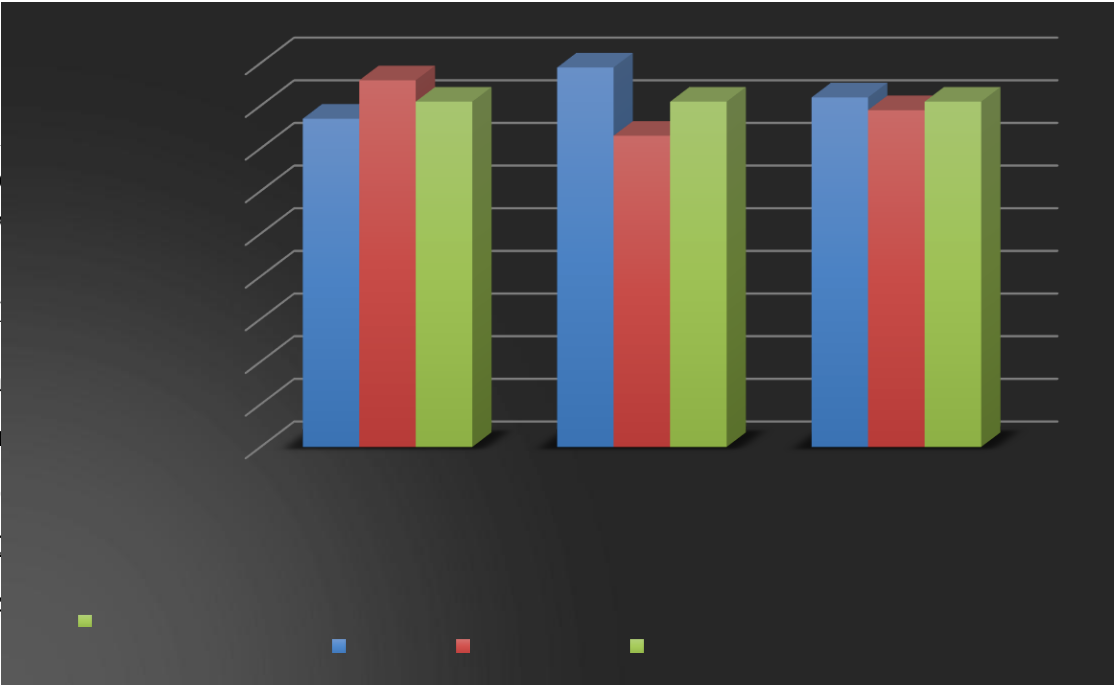


Table 8
set of
show

4.6.

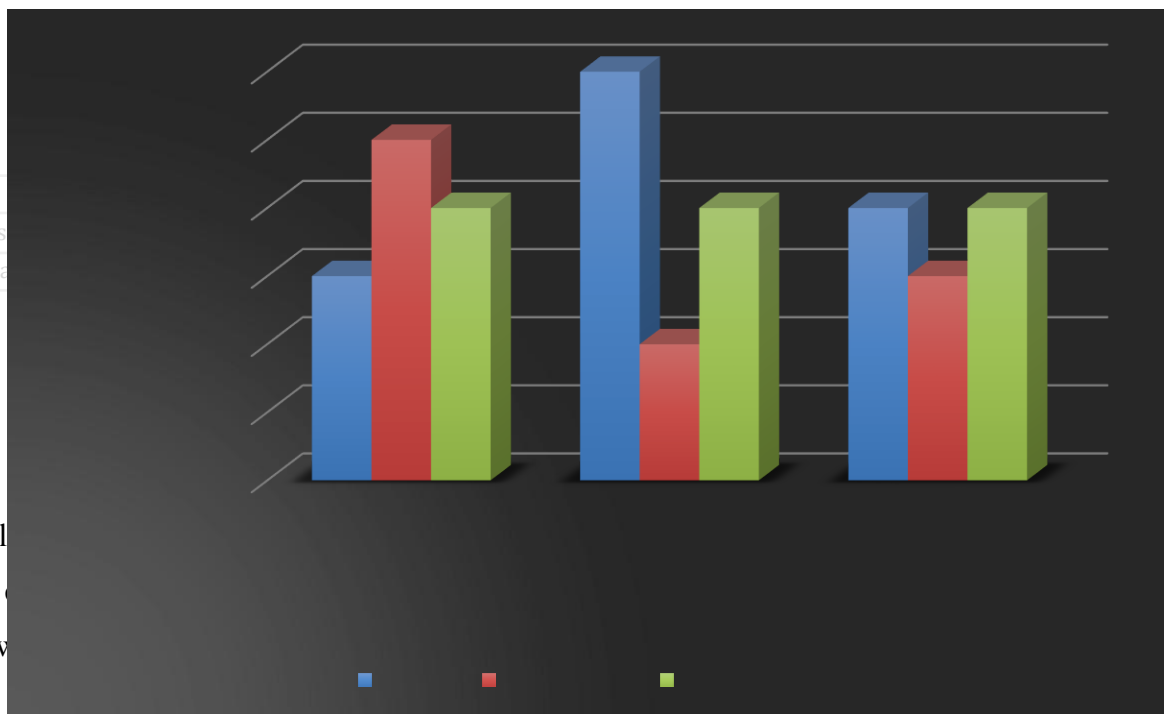
In ta
clear
depi
of 12
we c





Normal
Tuberculosis
Average/total

Table
set c
show



4.7 Why VGG16 performed better compared to baseline CNN

VGG16 is a pretrained model which means it uses transfer learning it has been already trained over a bench mark dataset IMAGENET which contains 14million images belonging to 1000 different classes, large amounts of data is essential for good performance of Neural networks and features extracted from images are similar to all layers irrespective of domain of the image to which it belongs to thus only by finetuning i.e. changing the last layers of the pretrained model gives better accuracy compared to CNN build from scratch which is trained over just 2040 images all the images present in ImageNet dataset are non-medical images but still pre trained model can be used for

medical images classification because features of all the images extracted in starting layers of a CNN are similar so it doesn't cause any problem we need to change the last

layers of the model and train it again over our dataset of images which is chest x rays in our case and it gives better accuracy compared to model build from scratch.

CHAPTER-5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

We have built a model which detects tuberculosis using baseline CNN method. Data is pre-processed for better accuracy along with image data augmentation, as deep learning models perform better with large amounts of data in the CNN model used, we have used 12 convolutional layers along with 2 fully connected layers and an Adam optimizer and binary cross-entropy loss is used another model pretrained VGG16 with Adam optimizer and binary cross-entropy is also used for both the models we trained the model on augmented images.

We designed an automated system which is used for screening CXRs for detection of Tuberculosis. The system is provided with CXR as input, it first performs the segmentation of lung region using Convolutional Neural Networks (CNN). We compute statistical features as input image into either normal or not.

The abnormality detection in the working Chest radiography algorithm takes over higher sensitivity. So, it is used to differentiate between the presence of tuberculosis or not.

System can be implemented further for screening as the accuracy of the proposed algorithm was good. The time for computing chest radiography detection algorithm is more. But it will lead to the reduction of human work by processing all the images.

We have demonstrated the working of our system for automatic detection of Tuberculosis, using a CNN which is trained on 2040 images and is tested over 120 images both the training and testing images are taken from datasets Shenzhen and Montgomery datasets.

The baseline CNN model which is run for 100 epochs with optimizer as Adam and loss as binary cross-entropy achieved 81% accuracy and the model pre trained VGG16 model which is run for 20epochs using optimizer as Adam and loss as binary cross-entropy achieved any accuracy of 90%. From the above results we can see that pretrained model performed better than baseline CNN model.

5.2 Future Scope

Further improvements will include:

- Lung detection methods will be modified further to get good approximation and lung structure should be properly analysed.
- Other techniques can be designed in the system for screening of tuberculosis that include drug susceptibility test and smear microscopy test.
- In order to detect other abnormalities like pneumonia, heart diseases and cancer some other lungs features would also be considered hence making Chest radiography detection system more specific.
- Using most relevant parameters for better results.

References and useful links

1. World Health Organization (2018). Global tuberculosis report 2018. World Health Organization. <http://www.who.int/iris/handle/10665/274453>.
2. Fatima, S. (2018). Automated Tuberculosis Detection and Analysis Using CXR's Images. *Int. J. Comput. Electr. Eng.*
3. Karnkawinpong, T., and Limpiyakorn, Y. (2019). Classification of pulmonary tuberculosis lesion with convolutional neural networks. In *Journal of Physics: Conference Series*.
4. Lakhani, P., and Sundaram, B. (2017). Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology* 284, 574–582.
5. Qin, C., Yao, D., Shi, Y., and Song, Z. (2018). Computer-aided detection in chest radiography based on artificial intelligence: A survey. *Biomed. Eng. Online* 17, 1–23.
6. Sivaramakrishnan, R., Antani, S., Candemir, S., Xue, Z., Thoma, G., Alderson, P., Abuya, J., and Kohli, M. (2018). Comparing deep learning models for population screening using chest radiography. In.
7. “kaggle” (2019). No Title. Available at: www.kaggle.com [Accessed October 26, 2019].
8. Razavian, A.S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.
9. Liu, J., Liu, Y., Wang, C., Li, A., Meng, B., Chai, X., and Zuo, P. (2018). An original neural network for pulmonary tuberculosis diagnosis in radiographs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
10. Perera, P., and Patel, V.M. (2019). Learning Deep Features for One-Class

Classification. IEEE Trans. Image Process.

11. Hakim, B., and Basari (2019). Tuberculosis detection analysis using texture features on CXRs images. In AIP Conference Proceedings.
12. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems.
13. Perera, P., and Patel, V.M. (2019). Learning Deep Features for One-Class Classification. IEEE Trans. Image Process.
14. Lu, D., and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens.
15. Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

