Binary Classification of Pneumothorax in Chest X-Ray Images Using Deep Neural Network

Thesis

Submitted to the

G. B. Pant University of Agriculture & Technology
Pantnagar- 263145, Uttarakhand, India

By

Manjeet Kaur
ID. No. 56669

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

(Mathematics)

August, 2022

ACKNOWLEDGEMENTS

Writing this thesis has been fascinating and extremely rewarding. I would like to thank a number of people who have contributed to the final consequence in many different ways. To commence with, I pay my obeisance to GOD, the Almighty to have bestowed upon me good health, courage, zeal, illumination, strength and patience.

I hereby, express my profound sense of gratitude to my Advisor Dr. Arun Kumar, Professor and chairman of my advisory committee, Department of Mathematics, Statistics and Computer Science, College of Basic Sciences and Humanities, G. B. Pant University of Agriculture and Technology, Pantnagar, for his guidance, motivation and kind help. His expertise, invaluable guidance, constant encouragement, affectionate attitude, understanding, patience and healthy criticism added considerably to my learning.

I wish to express my gratitude to the esteem members of my advisory committee, Dr. Manoj Kumar , Professor and Head, Department of Mathematics, Statistics and Computer Science, College of Basic Sciences and Humanities, G. B. Pant University of Agriculture and Technology,

Dr. Sunjay Kumar, ,Associate Professor, Department of Mathematics, Statistics and Computer Science, College of Basic Sciences and Humanities, G. B. Pant University of Agriculture and Technology,

I also owe great thanks to Dr. Sandeep Arora, Dean, College of Basic Sciences and Humanities for providing necessary facilities during the pursuit of my research work.

It is my privilege to express my heartiest thanks and regards to Dr. Kiran P. Raverkar, Dean, College of Post Graduate Studies and Dr. Brijesh Singh, Dean, Student Welfare for providing necessary facilities throughout the period of my research work.

I wish to express my sincere thanks to Dr. Manoj Kumar, Professor and Head, Department of Mathematics, Statistics and Computer Science, for his encouragement during my degree programme.

It is privilege to express my heartiest regards and sincere thanks to Dr. S. B. Singh, Professor, Dr. Vinod Kumar, Professor , Dr. A.K. Shukla , Professor and Mr. R. K. S. Rajput, Assistant Professor for their cooperation and support throughout the degree programme. I also acknowledge a deep thanks to all the support staff members specially to the program specialist Mr. Puneet Chaubay for their day to day help in research work by providing all essential help during the studies.

I would like to acknowledge all the teachers I learnt from since my childhood, I would not have been here without their guidance, blessing and support.

My vocabulary fails to accentuate my profound reverence and sincere regards to my parents, my source of inspiration, encouragement Mr. Satnam Singh and Mrs. Suman lata, who raised me with love, believe in me, guiding me to achieve success at every step-in life and support me in all my pursuits. I have special words of thanks from the core of my heart to my lovely brothers Sagar and Gaurav and my sister Amandeep Kaur for their unconditional love, affection and care. I owe everything to my family.

I thank very deeply and sincerely all my seniors, Manish Sir, Makrand Sir, Kiran Ma'am, Himani Ma'am, Ankita Ma'am, Yaseen Sir, Shivani Ma' am, Lakshmi Ma'am, Amisha Ma'am, Sunitha Ma'am, Neha Ma'am, Navneet Ma'am, Sangeetha Ma'am, Reshmi Ma'am, Gaurav Sir, Prince Sir, Vaibhav Sir, Mintu Sir, Garima Ma'am and Yunish Sir for their supportive attitude and encouragement throughout the period of my degree programme. I am grateful to my seniors, Shalini Ma'am , Yashika Ma'am, Suman Ma'am and Abhilasha Ma'am for their warm company and love. I express my heartiest thanks to my friends especially Tejasvi, Aashu, Gauri, Mansi and Dikansha for their active help and support. There are no words to pay my regards to many unforgettable persons during my stay in Pantnagar, A very special thanks to my friends Merlin, Tanuja and Anjali whose care, mischief and kind support did not ever let me feel out of my family and made my stay at Pantnagar memorable.

I sincerely admire the contribution of my classmates Sakshi, Merlin and Rithikesh for their pleasant company, co-operation, help and for those memorable moments, which will always be remembered. I am also grateful to all my loving juniors Sudarshan, Dibyanshu, Bharti, Sourav, Tanuja and Pooja for their active help. Words are limited to express the contribution and active help given by them.

Lastly, I shall ever remain thankful to all those who could not have find separate names but had directly or indirectly helped me.

Thanks for all your encouragement!

(Manjeet Kaur)
Author

Pantnagar
August, 2022

CERTIFICATE

This is to certify that the thesis entitled **"Binary Classification of Pneumothorax in Chest X-Ray Images Using Deep Neural Network"**, submitted in partial fulfilment of the requirements for the degree of **Master of Science** with major in **Mathematics** and minor in **Computer Science** of the College of Post-Graduate Studies, G.B. Pant University of Agriculture and Technology, Pantnagar, India, is a record of bonafide research carried out by **Ms. Manjeet Kaur, Id. No. 56669** under my supervision and no part of the thesis has been submitted for any other degree or diploma.

The assistance and help received during the course of this investigation have been duly acknowledged.

Pantnagar
August, 2022

(Arun Kumar)
Chairman
Advisory Committee

CERTIFICATE

We, the undersigned, members of the Advisory Committee of **Ms. Manjeet Kaur**, **Id. No. 56669**, a candidate for the degree of **Master of Science** with major in **Mathematics** and minor in **Computer Science,** agree that the thesis entitled **"Binary Classification of Pneumothorax in Chest X-Ray Images Using Deep Neural Network",** may be submitted in partial fulfilment of the requirements for the degree.

(Arun Kumar)
Chairman
Advisory Committee

22/8/22
(Manoj Kumar)
Member

(Sanjay Kumar)
Member

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
b	:	Bias
BPN	:	Back Propagation Network
CNN	:	Convolutional Neural Network
DL	:	Deep Learning
e.g.	:	For Example
et al.	:	And other
f	:	Activation Function
FFN	:	Feed Forward Network
w	:	Synaptic weight
x	:	Input
y	:	Output
ReLU	:	Rectified Linear Unit
Chest X-Ray or CXR	:	Chest Radiograph
ResNet	:	Residual Neural Network
VGG	:	Visual Geometry Group
n	:	Input size of image
p	:	Padding
s	:	Stride
Conv_block	:	Convolutional Block
SVM	:	Support Vector Machine
L1	:	L1 Regularization
L2	:	L2 Regularization
CPU	:	Central Processing Unit
GPU	:	Graphics Processing Unit
TPU	:	Tensor Processing Unit
PIL	:	Python Imaging Library
TN	:	True Negative

TP	:	True Positive
FN	:	False Negative
FP	:	False Positive
dense	:	Dense Layer
dropout	:	Dropout Layer
AUC	:	Area Under the Curve
ML	:	Machine Learning
TL	:	Transfer Learning
PNX	:	Pneumothorax
0	:	Negative
1	:	Positive

Chapter-1 INTRODUCTION

New advancements in computer vision and machine learning underpin a set of algorithms that can decipher the content of photographs with high precision. These deep learning algorithms are being used to evaluate and assess biological images, and they are revolutionizing imaging data analysis and interpretation. These breakthroughs will make complex analysis routine and allow researchers to conduct previously unthinkable experiments. We examine the relationship between deep learning and cellular image processing, as well as the mathematical mechanics and programming frameworks of deep learning that are relevant to life scientists. Recently this field have grown tremendously as used in picture classification, image segmentation, object tracking, and augmented microscopy to see how far the field has come.

1.1 DEEP LEARNING

The term "Deep learning" was coined by Rina Dechter in the year 1986, while in the year 2000 Igor Aizenberg and colleagues introduced "Artificial neural network" in the context of Boolean threshold neurons.

Deep learning is an AI function that imitates the workings of the human brain in data processing and creating patterns for use in decision making. Deep learning is a subset of machine learning, it is based on artificial neural networks. The word deep is used here because several – several hidden layers are used to extract high-level features from the given input. The concept of Deep learning is very much similar to machine learning but uses different algorithms. Machine learning works with regression algorithms or decision trees, whereas deep learning uses neural networks that function similarly to the biological neural connections of the human brain.

1.1.1 ARTIFICIAL NEURAL NETWORK or ANNs

Artificial Neural Networks are computing systems that are inspired by biological neural networks that may execute a range of tasks employing a large amount of data. To get the most effective results from changing inputs several algorithms are employed to grasp the relationships in a given set of data. Different models are employed to forecast future results based on given information and also the network is taught to deliver the desired outputs. The nodes are joined in such a way that it functions similarly to somebody"s brain. To cluster and classify the data, several correlations and hidden patterns within the information are exploited.

ANN is an attempt to emulate the basic functioning of the human brain to perform complex functions that computer systems are not able to perform every day. Neuron networks are not meant to duplicate the human brain, but to receive information about how the human brain does.

BIOLOGICAL NEURON v/s ARTIFICIAL NEURON

Table 1.1: Difference between Biological and Artificial Neural

BIOLOGICAL NEURAL NETWORK	ARTIFICIAL NEURAL NETWORK
Soma	Node
Dendrites	Input
Synapse	Weights or Interconnections
Axon	Output

Properties of ANN

- Nonlinearity
- Input output mapping
- Adaptivity
- Evidential response
- Context information
- Fault tolerance
- Learn by example

Applications of ANN

- Speech recognition
- Computer vision
- Robot control
- Medical

- Driverless cars or transportation
- Portfolio management
- Time series prediction
- Face alignment

Different types of Learning in neural network

Neural networks are taught and trained in an identical way as a child‟s developing brain is. They cannot be programmed specially for a task. Instead, they are trained in such a way that they will be adapted to changing input. To educate a neural network, there are three approaches or learning paradigms.

- Supervised learning
- Unsupervised learning
- Reinforcement leaning

Supervised Learning- As the name implies, supervised learning takes place in the presence of a tutor or a supervisor. It indicates that for a few data sets a group of labelled data sets already exists with the required output, i.e., the neural network‟s optimal action. The system is then given fresh data sets to evaluate and build the correct prediction by the help of training datasets.

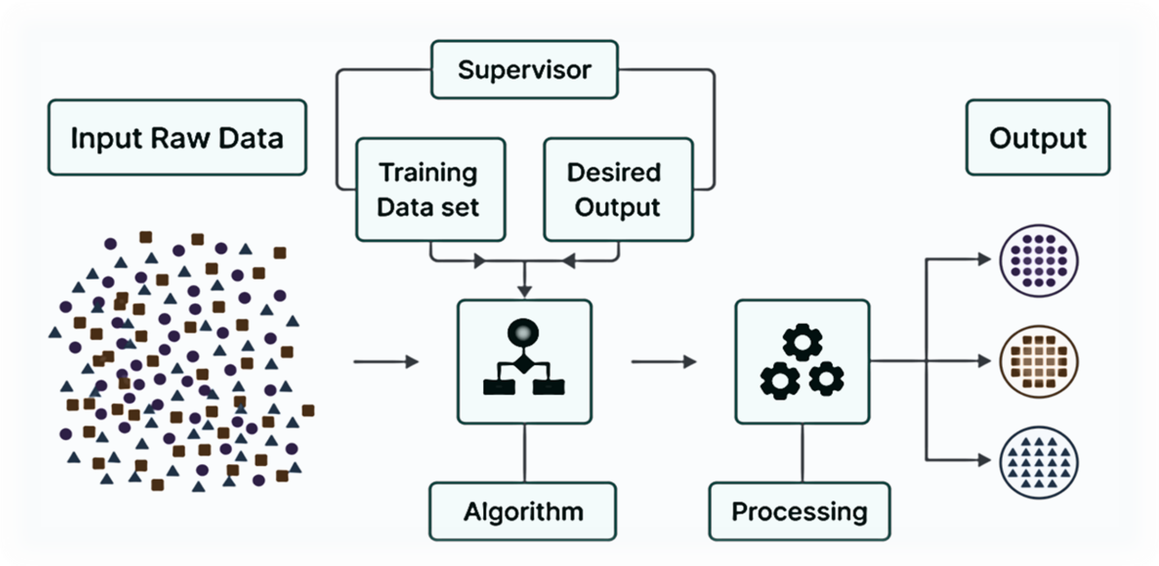


Figure 1.1: Supervised learning procedure is shown.

Unsupervised Learning- There is no teacher or supervisor readily available as the name implies. The data isn‟t categorized or classed during this case, and thus the neural network has no prior instructions. This needs the machine to group the presented datasets supported similarities, differences, and patterns with no prior training.

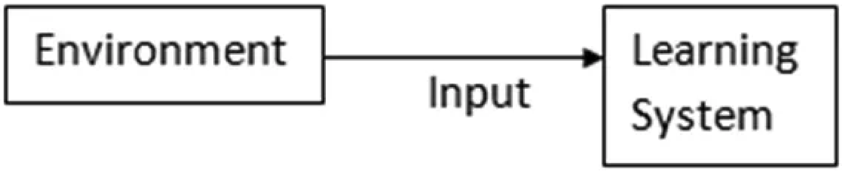


Figure 1.2: Unsupervised learning procedure is shown

Reinforcement Learning- A machine learning training method called reinforcement learning rewards desired behaviours and penalizes undesirable ones. A reinforcement learning agent can typically perceive and interpret its environment, act, and learn by making mistakes.

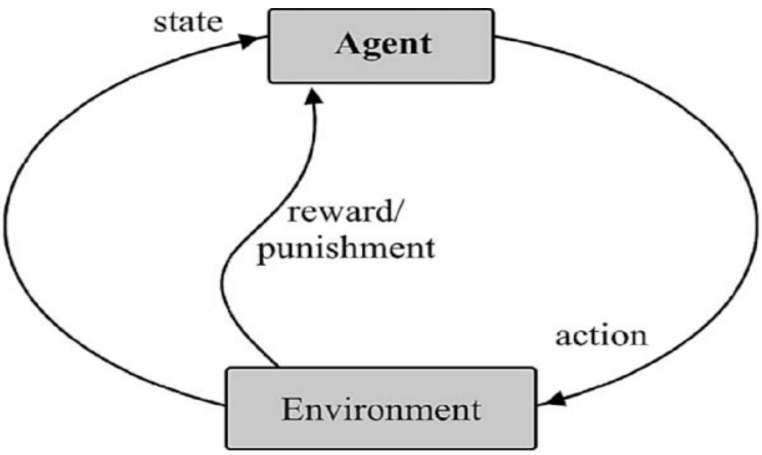


Figure 1.3: Reinforcement Learning procedure

MODEL OF ANN

The neural networks may be weighted by the graph with nodes representing neurons and edges representing connections with weights. It‟s designated by „x‟ and receives (n) input from the outer world. Each input is multiplied by its weights before being joined together, a bias „b‟ is added. The activation function is then passed this weighted sum. The activation function controls the amplitude of the neuron‟s output. The Sigmoid function, Piecewise function, Threshold function, Hyperbolic Tangent function, etc. are examples of activation functions.

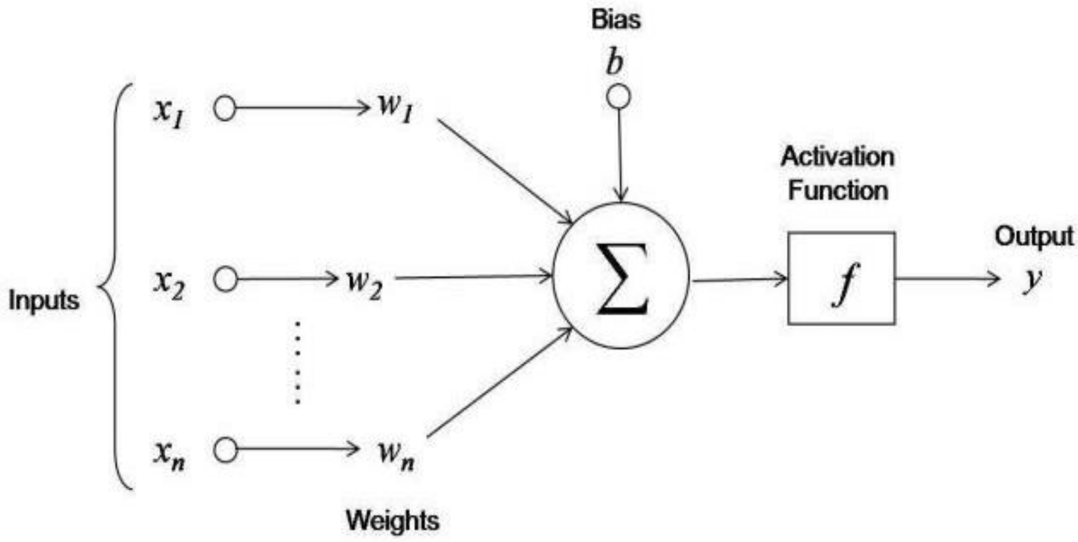


Figure 1.4: Working of neural network

Output y = f (x_1*w_1+ x_2*w_2+x_3*w_3+……………+ x_n*w_n)

Here, x_i = input

w_i = synaptic weights

b= bias

f= activation function;

y= output

ELEMENTS OF NEURAL NETWORK ARE DESCRIBED BELOW

A neural network consists of three layers namely

INPUT LAYER- Input features are accepted by this layer which feeds the network input from the environment. No computation is done in this layer and they just pass the information (features) to the next (i.e. hidden) layer.

HIDDEN LAYER- This layer"s nodes aren"t visible to the surface world, they appear to be a part of the abstraction offered by any neural network. The hidden layer calculates all sorts of the features that enter through the input layer and sends the results to the next (i.e. output) layer.

OUTPUT LAYER- This layer communicates the network"s learned knowledge to the outer world.

SYNAPTIC WEIGHTS: In ANN each node is a set of inputs, weights, and bias values; the weights are the most important factor to convert an input (feature) to impact output. The weights are associated with inputs and provide links between nodes to the next layer. The weights may be positive or negative. The weight shows the effectiveness of a particular input that is the more the weight the more impact it has on the network. Weights are responsible for increasing the steepness of the activation function which means weight decides how fast the function triggers.

ADDER: It is used for computing inputs with their respective weights.

BIAS: It is a constant that allows us to shift that is it increases or decreases the net input value of an activation function. Bias in neural networks plays the same role that a constant in linear function does. Bias is responsible for delaying the triggering of the activation function. At each node, the processing done by each neuron is :-

Let the inputs be x_1, x_2, x_3 and accordingly synaptic weights are w_1, w_2, w_3;

Output = sum (weights * input) + bias

or

$n = (i_1*w_1+ i_2*w_2+i_3*w_3+\ldots\ldots\ldots\ldots\ldots +i_n*w_n) + b\ldots\ldots\ldots.. (1)$

The above equation (1) can be compared to the equation of a straight line-

$Y = m*x + c$

Where

m = weight

c = bias

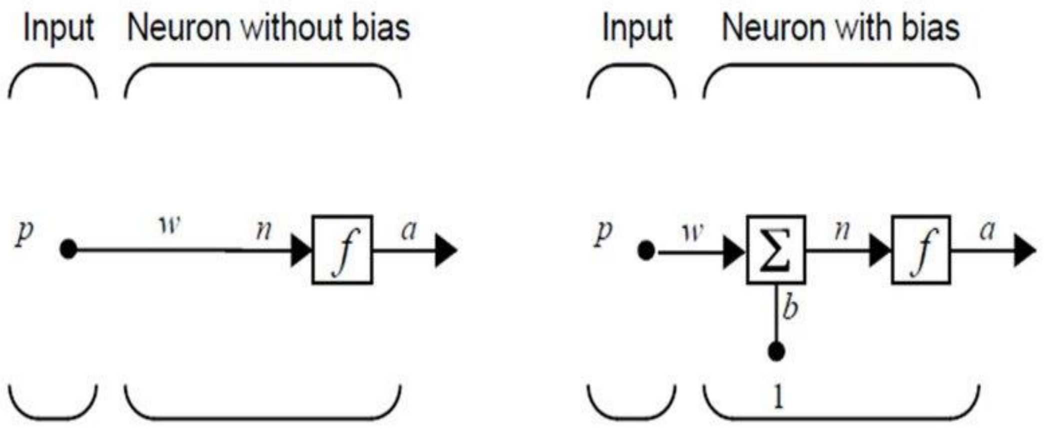


Figure 1.5: neuron with and without bias

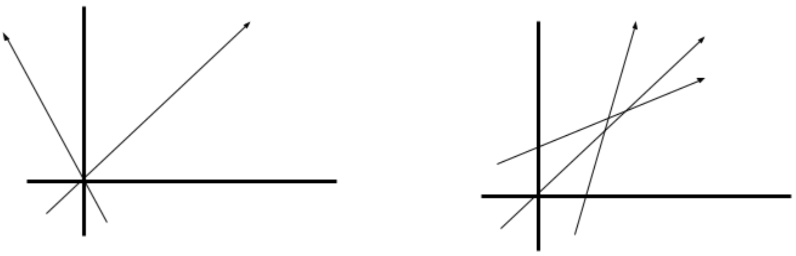


Figure 1.6: The graphs in the absence and presence of bias (c) are shown

If the neuron is without bias the model will be trained over a point passing through origin only. But this does not happen in the real world. Thus, on introducing bias the model becomes more flexible.

1.1.2 ACTIVATION FUNCTION

The activation function is used to "introduce non-linearity" into the output of a neuron. The Activation function decides whether to activate a neuron or not which means that the activation function will decide whether the neuron‟s input is important or not by computing a weighted sum and adding bias to it.

We know that neurons in a neural network work in accordance with their weight, bias, and activation function. In a neural network, we would update the weights and biases of the neurons based on the output error. This is referred to as back-propagation. Because the gradients are supplied along with the error to update the weights and biases, activation functions enable back-propagation.

Need of Non-linear Activation Function- Without an activation function, a neural network is essentially a linear regression model. The activation function transforms the input in a nonlinear manner, allowing it to learn and perform more complex tasks.

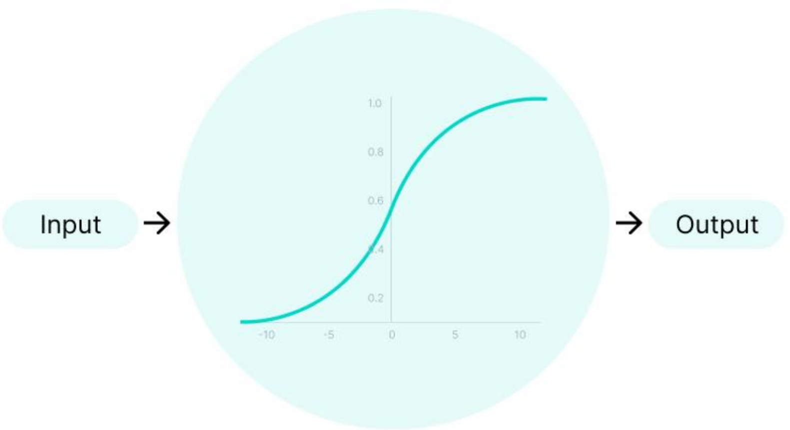
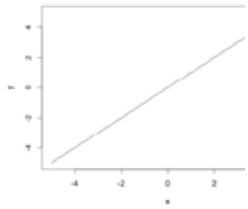
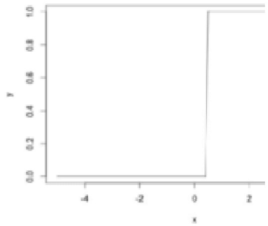
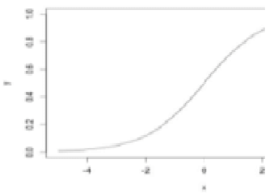
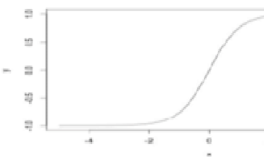
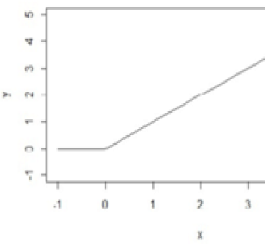


Figure 1.7: Activation Function

Table 1.2: Activation functions with their equation and graphs.

Activation Function	Description	Equation	Implementation
Linear	• The most fundamental activation function is the linear one because it does not change the neuron output at all	$\emptyset(x) = x$	
Step Or threshold	• It is simple. • It return 1 (true) for values that are above the specified threshold •	$\emptyset(x) = \begin{cases} 1, if\ x \geq 0.5 \\ 0, otherwise \end{cases}$	
Sigmoid or Logistic	• It is a popular option for feed-forward NNs that must output only positive numbers. • It ensure that values stay within a relatively small range	$\emptyset(x) = \frac{1}{1+e^{-x}}$	
Hyperbolic Tangent	• It is also a popular activation function for NNs which must output values in the range between -1 and 1. •	$\emptyset(x) = \tanh(x)$	
Rectified Linear Unit (ReLU	• It is a linear, non-saturating function. • ReLU does not saturate to -1, 0, or 1. • A saturating activation function moves towards and eventually attains a value	$\emptyset(x) = \max(0, x)$	

1.1.3 NEURON CONNECTION ARCHITECTURE

SINGLE-LAYER FEED FORWARD NETWORK- Only two layers that are an input layer and an output layer . Now as for the n input layer no computation is performed so we don"t count this layer. When distinct weights are applied on input nodes and the cumulative effect per node is considered, an output layer is formed and then the neurons collaboratively gives the output layer the task of computing the output signals.

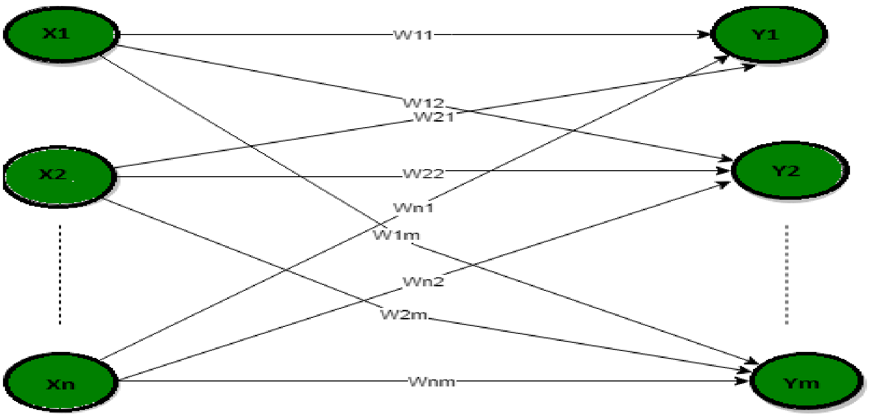


Figure 1.8: Single Layer feedforward Network

MULTILAYER FEED FORWARD NETWORK - There is also a hidden layer within this layer that is internal to a network and has no direct interface with the exterior layer. Because of information through the input function and the intermediate computations necessary to give the output Z, the presence of one or more hidden layers enables the network to be computationally stronger, than the feed-forward network. There are no feedback links, therefore the model's outputs do not feed back into it

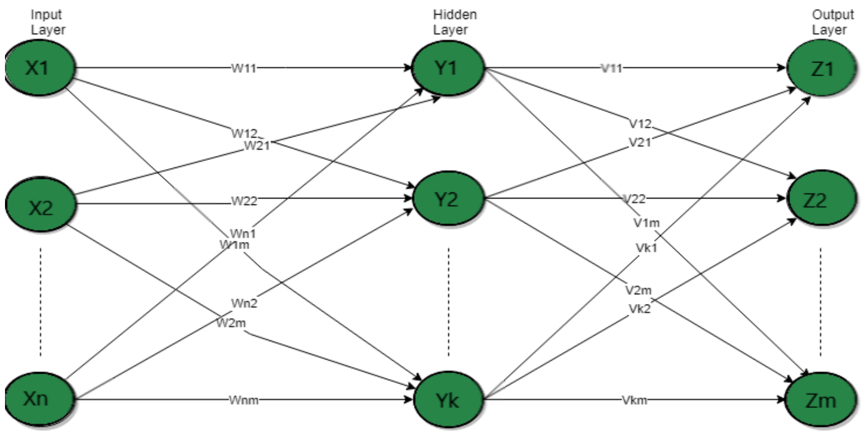


Figure 1.9: Multilayer Feed forward Network

SINGLE NODE WITH ITS FEEDBACK- Feedback networks are formed when outputs can be routed back as inputs to the same layer or preceding layer nodes. Recurrent networks are closed-loop feedback networks. A single recurrent network with a single neuron with feedback to itself is seen in the diagram below.

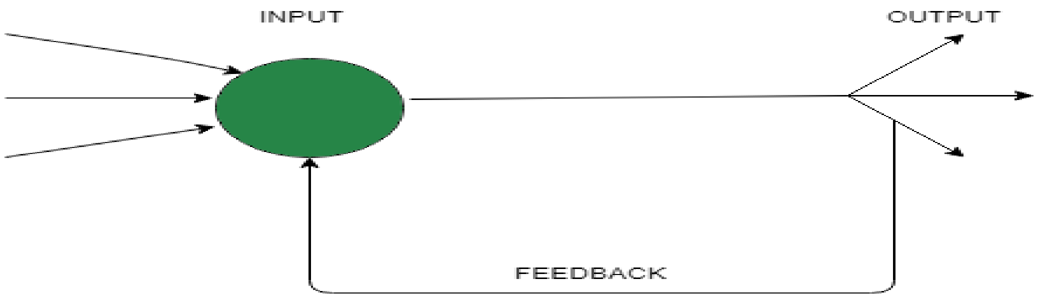


Figure 1.10: Single Node Feed Backward Network

SINGLE LAYER RECURRENT NETWORK- The network shown below is a single-layer network with a feedback connection, in which the output of the processing element can be routed back to itself, to another processing element, or both. A recurrent neural network is a type of artificial neural network in which nodes form a directed graph along a succession of connections. This enables it to display dynamic temporal patterns for a succession of events. RNNs, unlike feedforward neural networks, may process sequences of inputs using their internal state (memory).

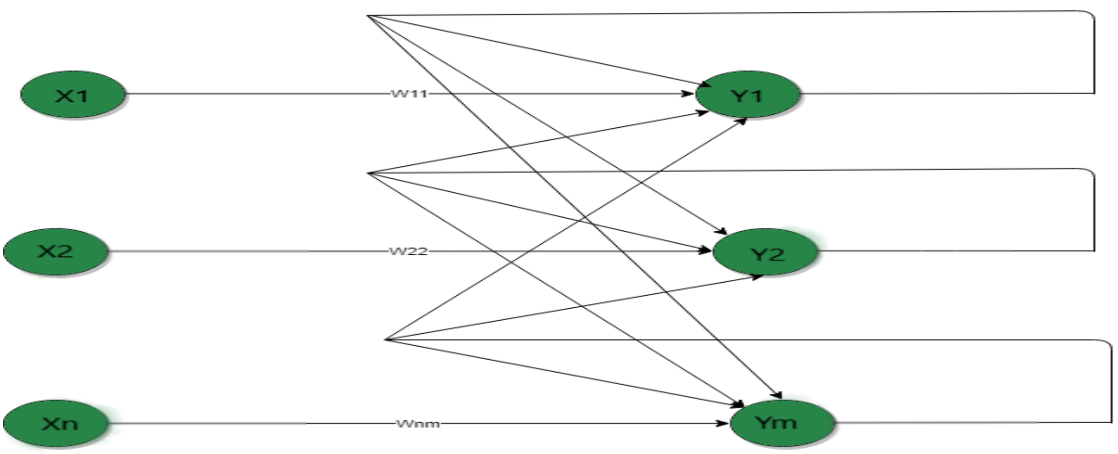


Figure 1.11: Single layer Recurrent Network

MULTILAYER RECURRENT NETWORK- The output of processing units in this sort of network can be directed to processing units in the same and preceding layers, forming a multilayer recurrent network. They do the same thing for each element of a sequence, with the output being determined by the prior calculations. At each time step, no inputs are required. A Recurrent Neural Network's hidden state, which captures some information about a sequence, is its most unique characteristic.

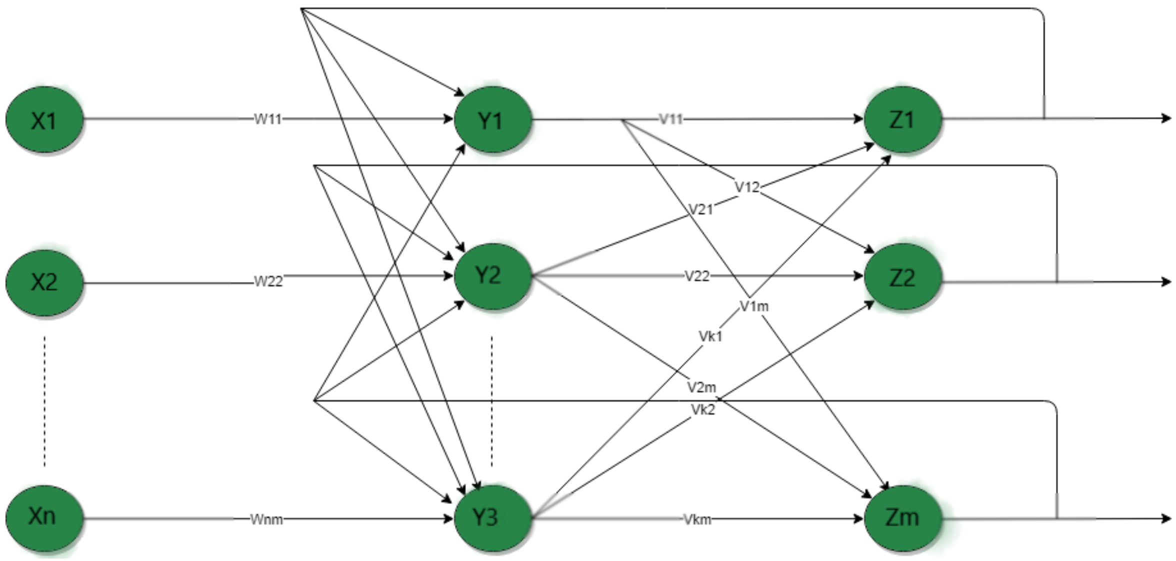


Figure 1.12: Multilayer recurrent Network

1.4 CLASSIFICATION OF NEURAL NETWORK

In today‟s world neural network is a hot researched area in computer science as it is one of the most effective ways to solve real-world problems. For different domains, we have different types of neural networks which are named as.

- Shallow Neural Network
- Deep Neural Network
- Convolutional Neural Network
- Recurrent Neural Network
- Long Short-Term Memory
- Attention Based Network
- Generative Adversarial Network

1.1.5 BACKPROPAGATION

The core of neural network training is backpropagation. It's a technique for fine-tuning the synaptic weights of a neural network based on the previous epoch's error rate (i.e., iteration). By fine-tuning the weights, we can lower error levels and enhance the model's generalization, trying to make it more accurate.

Backpropagation is an abbreviation for "backward propagation of errors" in the neural network. It's a common way to train artificial neural networks. This method is useful for calculating the gradient of a loss function concerning all of the network's weights.

NEED OF BACK-PROPAGATION- Backpropagation is a quick, basic, and straightforward programming method. Apart from the number of inputs, there are no parameters to modify. It is a flexible method because it does not necessarily require any prior knowledge of the network. It is a simple method that works well in certain cases. It doesn't require any special reference of the functions to be trained.

THE WORKING OF BACKPROPAGATION ALGORITHM -The chain rule is being used by the backpropagation method in neural networks to determine the gradient of the loss function for a single weight. Unlike native direct computation, it efficiently computes one layer at a time. It evaluates the gradient but doesn't specify how it'll be used. It generalizes the delta rule's computation. Steps involved in back-propagation are discussed below-

- X inputs enter via a pre-connected path.
- Real weights W are used to model the input. The weights are normally chosen at random.
- From the input layer to the hidden layers to the output layer, calculate the output for each neuron.
- Calculate the output‟s error.
- Actual Output – Desired Output = Error
- Return from the output layer to a hidden layer to change the weights to minimize the error.
- Repeat the procedure until the required result is obtained.

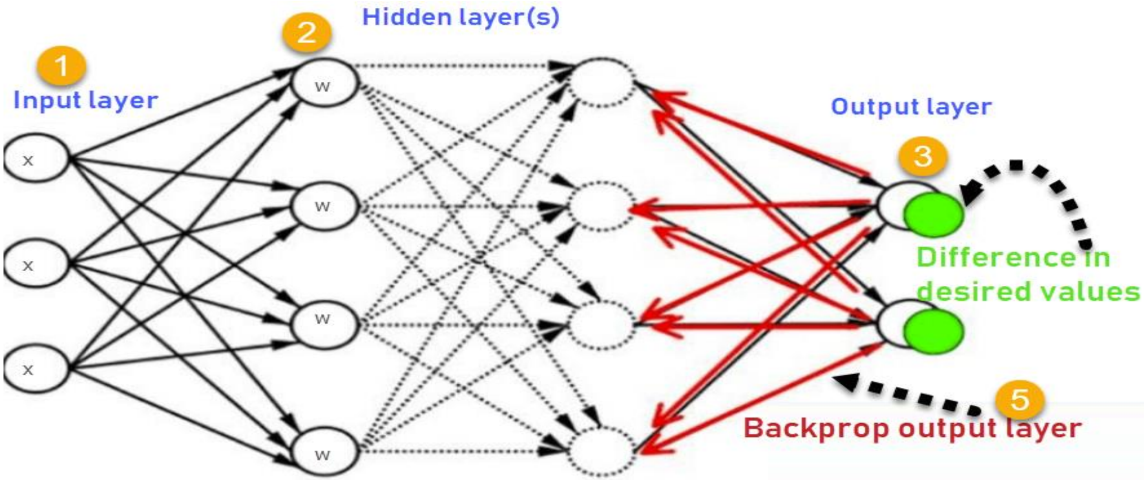


Figure 1.13: Working on the backpropagation algorithm

THERE ARE TWO TYPES OF BACK-PROPAGATION

- **STATIC BACK-PROPAGATION-** Static back-propagation is a type of back propagation network that generates a static input to static output mapping. It helps solve problems like optical character recognition that require static classification.
- **RECURRENT BACKPROPAGATION-** In data mining, recurrent back propagation feeds forward unless mixed values are obtained. The error is then calculated and transferred backward.

The key difference between these two methodologies is that in static back-propagation, the mapping is fast, whereas, in recurrent back-propagation, it is non static.

There are three stages of training the Back propagation algorithm:-

- Feed forward network
- Back propagation of error
- Weight up-dation

1.1.6 CONVOLUTIONAL NEURAL NETWORK (CNN OR ConvNets)

CNNs are ordinary networks that assume the inputs are images or can also be defined as a variety of Deep Neural networks that can acknowledge and classify specific features of an image, and they are extensively employed for image analysis. Picture and video recognition, image classification, medical image analysis, computer vision, and natural language processing are only a few of their uses.

The mathematical function of convolution, which is a special sort of linear operation in which the product of two functions results in a third function that reflects how the form of one function is modified by the other, is denoted by the term "convolution" in CNN. Simply, two matrices are multiplied to provide an output that is used to extract features from the image.

On comparing with other image classification methods, CNNs require very little pre-processing. This means that the network learns to optimize the filters (or kernels) by automatic learning, as opposed to hand-engineered filters in classical techniques. This lack of reliance on previous information or human intervention in feature extraction is a remarkable benefit.

THE MAJOR OPERATIONS IN CNN IMAGE DETECTION/ CLASSIFICATION

- Convolution layer
- Pooling layer
- Padding layer
- Fully connected layer
- Dropout
- Activation Functions

TWO MAIN FUNCTIONS OF CNN ARE

- FEATURE EXTRACTION- In a process known as Feature Extraction, a convolution technique extracts and recognizes the distinct characteristics of an image for analysis.
- CLASSIFICATION- A fully connected layer uses the outcome of the convolution process to determine the image's class using the information acquired in an earlier phase.

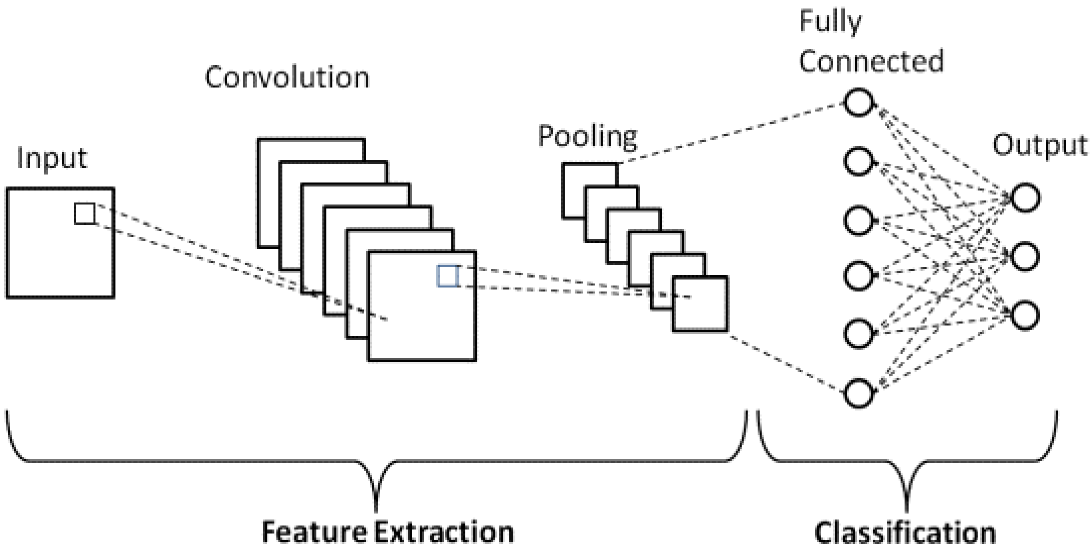


Figure 1.14: Feature extraction and classification process is shown

CONVOLUTIONAL LAYER- In 1D, the convolution operation works with two signals, while in 2D, it works with two pictures. A convolution is a combined integration of two functions that shows how one changes the other mathematically:

$$f(x) * g(y) = \int_{\infty f}^{\infty}(\tau) * g(r_{l-x})\, d\tau$$

The basic goal of a convolutional layer is to detect edges, lines, colour dips, and other visual elements from images. This is a remarkable property since it can recognize an attribute in any region of the image after it has learned it at a specific place in the image.

Filters (also known as kernels or feature detectors) are used by CNNs to recognize characteristics such as edges that are present across an image. A filter is just a set of weights that is a matrix of values that have been taught to detect specific traits. The filter is in motion. A filter is just a set of weighted values (matrix) that have been taught to recognize specific traits. The filter scans each area of the image to see whether the feature it's looking for is present. The filter performs a convolution operation, which is an element-wise product and sum between two matrices, to generate a value expressing how convinced it is that a specific feature is there.

The convolution operation between the filter and that section of the image produces a real number with a high value when the feature is present in that part of the image. But if the feature is absent, it results in a low value.

Feature detection in process

When any filter is applied to the entire image, it produces an output matrix known as feature maps or convolved features, which stores the filter's convolutions over various sections of the image. We now get a 3D output: one 2D feature map per filter, thanks to the numerous filters. To do element-wise multiplication, the filter must have the same number of connections as the input image.

A stride value can also be used to apply a filter across the input image at different intervals. The stride value specifies how far the filter should move each time it is activated.

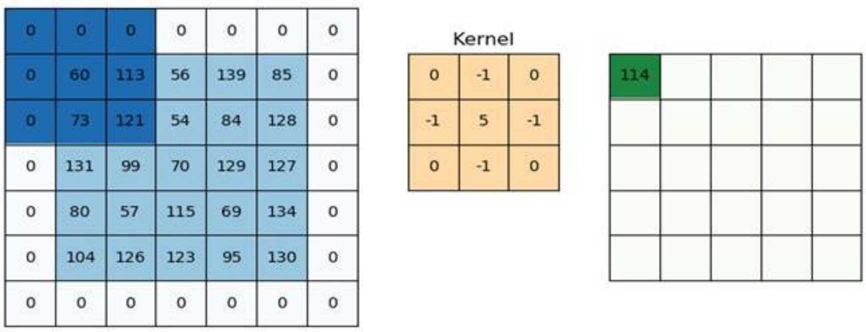


Figure 1.15: Feature detection

The number of output layers of a given convolutional block can be determined as

$$\mathbf{n_{out} = [(n_{in} + 2*p - k) / s] + 1}$$

n_{in} = number of input features

n_{out} = number of output features

k: convolutional kernel size

p: convolutional padding size

s: convolutional stride size

Padding Layer- It refers to the addition of pixels to an image when it‟s been processed by the kernel of the CNN.

It is the best approach, where the number of pixels needed for the convolutional kernel to process the edge then the pixels are added to the outside copying the pixels from the edge of the images or simply pad the image with zeroes (zero-padding) to an image to get an accurate analysis.

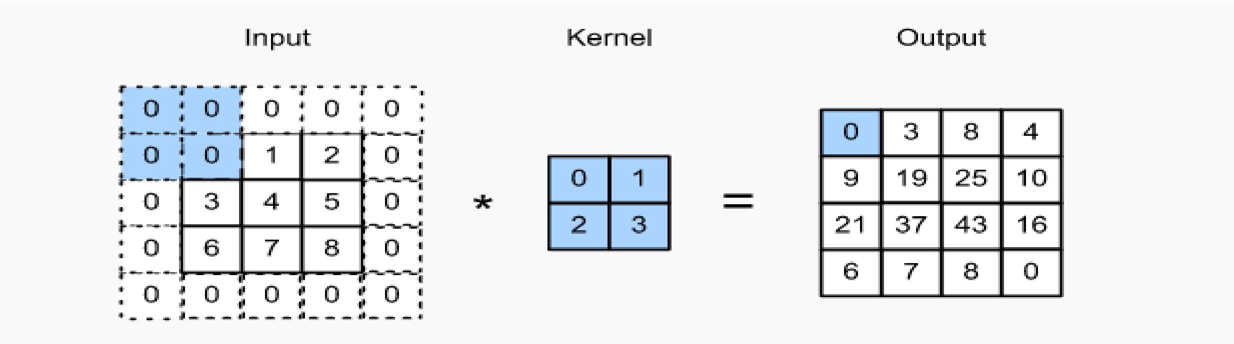


Figure 1.16: Convolutional padding

Pooling layer- A Pooling Layer is usually used to reduce the amount of the convolved feature map to decrease the computational expenses. This is accomplished by reducing the interconnections between layers and operating independently on each feature map. There are numerous sorts of pooling operations, based on the mode utilized.

The three types of pooling operations are described as:

- Max pooling: The batch's maximum pixel value is chosen.
- Min pooling: The batch's minimum pixel value is chosen.
- Average pooling selects the average value of all pixels in the batch.

The term "batch" refers to a group of pixels with a size equal to the filter size, which is determined by the image's size.

Flattening - The 3D depiction of an image is turned into a feature vector that is sent into a multi-layer perceptron to yield probability after numerous convolution layers and down - sampling procedures. To make a prolonged feature vector, individual rows are merged. If there are numerous input layers, their rows are also merged to generate a larger feature vector.

The flattening operation is depicted in the diagram below:

Figure 1.17: The flattening operation is depicted

Fully Connected Layer:- The weights and biases, as well as the neurons, make up the Fully Connected (FC) layer, which is then used to interconnect the neural connections between two layers. The last several layers of a CNN Architecture are usually positioned before the output layer.

The previous layers' input images are flattened and supplied to the FC layer in this step. After that, the flattened vector is sent via a few additional FC layers, in which the mathematical functional operations are normally performed. In ANNs, the fully connected layer is identical to the hidden layer, except that it is fully connected in this case. The anticipated classes are found in the output layer. The data is sent over the network, as well as the error value is determined. To enhance the forecast, the loss would then be back propagated through a system. The classification procedure gets established at this point.

Activation Function- Finally, the activation function is one of the most crucial elements in the CNN model. They're utilized to learn and approximate any form of network variable-to-variable association that's both continuous and complex. In simple terms, it determines which model information should trigger in a forward direction and which should not at the network's end. It provides the network with non-linearity. The ReLU, Softmax, tanH, and Sigmoid functions are some of the most often applied activation functions. Each of these functions has a distinct use. For a binary classification CNN model, sigmoid and softmax functions are favoured, whereas softmax is typically employed for multi-class classification.

1.2 Pneumothorax

An abnormal condition in which air gets collected in the pleural space between the lungs and the chest wall, this air is pushed outside of our lungs and eventually leads to the collapse of the lungs either partially or completely. A blunt or penetrating chest injury, certain medical procedures, or damage from underlying lung disease can all result in a pneumothorax. It could also happen for no apparent cause.

Symptoms typically include sudden onset of sharp, one-sided chest pain and shortness of breath. The severity of the symptoms may indeed be determined by the quantity of lung tissue that has collapsed.

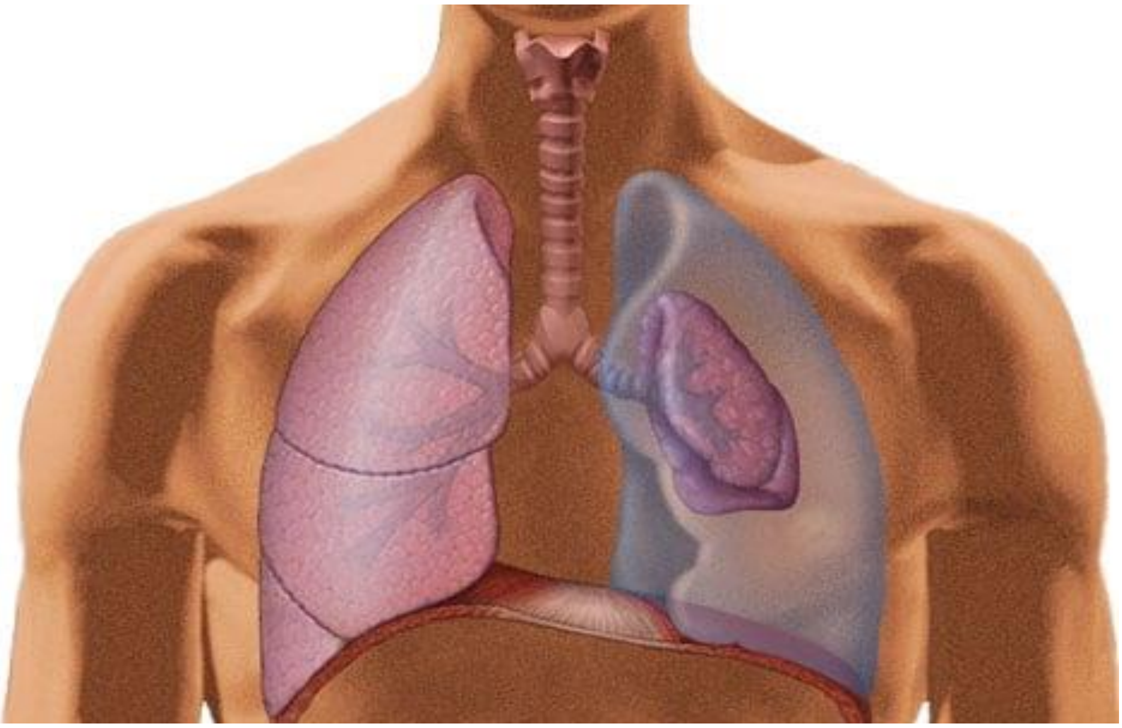


Figure 1.18: Collapsed lung and normal lung

CAUSES OF PNEUMOTHORAX- Pneumothoraxes could be caused by a wide variety of factors, including:

- **Injury to the chest-** Lung collapse can be caused by any blunt or penetrating chest injury. Some injuries occur as a result of physical attacks or vehicle accidents, while others occur as a result of medical procedures that require the introduction of a needle into the chest.
- **Lung illness-** It is a common ailment. Lung tissue that has been damaged is more vulnerable to collapse. Many underlying issues, such as chronic obstructive pulmonary disease (COPD), cystic fibrosis, lung cancer, or pneumonia, may damage lung tissue
- **Air blisters that have ruptured-** Just on top of the lungs, little air blisters (blebs) might be formed. These air blisters can occasionally burst, permitting air to flow into the area around the lungs.
- **Mechanical ventilation-** People requiring mechanical aid to breathe could develop a severe type of pneumothorax. The ventilator might cause an air pressure imbalance in the chest and the lung may entirely collapse.

RISK FACTORS -Generally, men are significantly more prone than women to have a pneumothorax. People between the ages of 20 and 40 are more likely to have a pneumothorax caused by ruptured air blisters, especially if they are exceptionally tall and underweight. A pneumothorax can be caused or exacerbated by underlying lung illness or artificial ventilation. Other aspects to consider are:

Smoking- Even without emphysema, the threat increases with the amount of time and cigarettes smoked.

Genetics- Pneumothorax appears to be transmitted in families in some cases.

Previous pneumothorax- The one who has suffered from pneumothorax early is more likely to suffer again from this.

DIAGNOSIS

Pneumothorax can be diagnosed using several techniques-

- It can be diagnosed using chest X-ray.
- Computed Tomography (CT) scan may be needed to provide more detailed images.
- Ultrasound can also be used to detect pneumothorax.

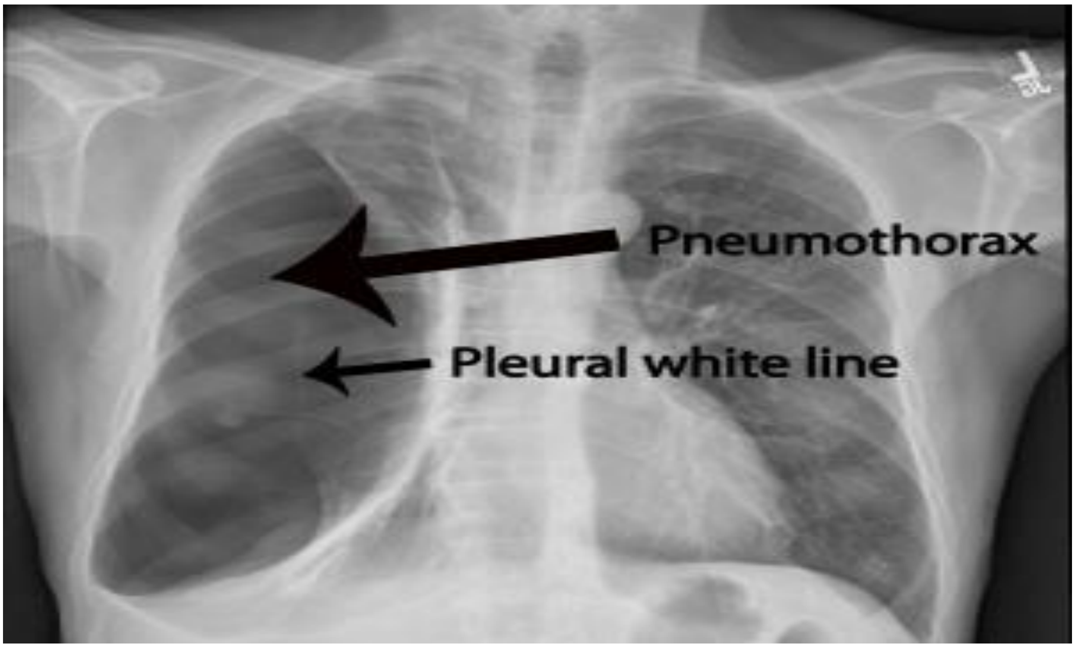


Figure 1.19: Chest X-ray image with pneumothorax

TYPES OF PNEUMOTHORAX

- PRIMARY PNEUMOTHORAX- A spontaneous pneumothorax is also known as a primary spontaneous pneumothorax. It is distinguished by the lack of a definite cause or underlying lung pathology. There may be contributory variables such as cigarette smoking, family history, or the rupture of the bulla (small air-filled sacs in the lung tissue), but none of these will induce pneumothorax.
- SECONDARY PNEUMOTHORAX- A non-spontaneous or complex pneumothorax is another name for secondary pneumothorax. COPD, Asthma, Tuberculosis, Cystic Fibrosis, or Whooping Cough are all examples of underlying lung pathologies.
- TRAUMATIC PNEUMOTHORAX- Trauma to the lungs causes a traumatic pneumothorax. The following have been some of the reasons: A stab wound, a gunshot, an injury from a car accident, or any other type of lung trauma. An iatrogenic pneumothorax is a pneumothorax that arises as a result of a medical operation or inappropriate medical care, such as an inadvertent puncture to the lung during surgery.
- TENSION OR NON-TENSION PNEUMOTHORAX- An excessive pressure builds up around the lungs due to a rupture in the lung surface that enables the air into the pleural cavity during inspiration but does not let it pass out on expiration, resulting in a tension pneumothorax. A one-way valve is created by the rupture. Lung collapse occurs as a result of this.

  The air is removed by placing an underwater drip into the pleural cavity through a surgical intervention. This high pressure can also hinder the heart from pumping properly, resulting in shock. A non-tension pneumothorax is not as serious as a tension pneumothorax since there is no continuous accumulation of air and hence no increased pressure on the organs and chest.

TREATMENT- The goal of the treatment of pneumothorax is to release pressure on your lung so it can expand again. A secondary goal may be to avoid recurrences, depending on the origin of the pneumothorax. The methods used to achieve these objectives vary depending on the severity of the lung collapse and, in some cases, for one‟s overall health. Observation, needle aspiration, chest tube insertion, nonsurgical repair, and surgery are some of the treatment options available. To help with air reabsorption and lung expansion, you may need supplemental oxygen therapy.

1.3 MOTIVATION

CNN"s became popular tools for computer vision or image recognition tasks likewise as image classification or segmentation with the launch of AlexNet in the year 2012 and are now thought to be the state-of-the-art method for handling these machine vision challenges. Activities like image classification and segmentation are successful enough and have the ability to outperform humans. Also, CNN"s are extensively used in medical image analysis. When the cases are of sophisticated medical situations, accurate models might aid clinicians in image lesion segmentation and can improve decision making and thus, can solve the matter of human error in diagnosis.

In recent years, the research on this subject has focused on strengthening mathematical structures as well as CNN architectures to boost performance on these problems. However, combining the predictions of multiple existing CNN architectures to provide more accurate models using ensemble machine learning approaches remains a very unexplored topic of research. instead of constructing new CNN architectures to enhance picture segmentation results, we can use layered generalisation approaches to merge multiple existing CNN designs to form more accurate models. These ensemble machine learning techniques have the potential to boost CNN generalisation on increasingly complicated medical image segmentation issues, where even experienced physicians have difficulty delivering consistent and accurate results.

1.4 OBJECTIVE

In the proposed study we aim to build a model for the binary classification of pneumothorax and recall its presence or absence. The proper diagnosis is of great importance for improving care coordination among inter professional team members to improve outcomes for patients affected by pneumothorax. The main objective of proposed study is to develop several frameworks supported computing techniques for automated identification of pneumothorax from chest radiographs. We will also develop some models for pneumothorax automated diagnosis using deep neural network. Some learning algorithms can also be developed in this proposed study. We will compare the results as well as the models for comparative study of different developed models.

Chapter-2 REVIEW OF LITERATURE

Pneumothorax is a condition in which air builds up in the pleural space of the thorax, which is located between the chest and the lungs. The air that is present in the lungs during breathing to escape gets trapped between the chest and the lungs, in the pleural space of the chest, causing chest pains, breathing difficulty, and, in the case of tension pneumothorax, the lung may also collapse completely. As a result, this should be diagnosed early so that treatment can begin as soon as possible. Neural networks are effective in image classification tasks, and many researchers have demonstrated that chest anomalies such as pneumonia, pneumothorax, and others, which are diagnosed by radiologists, can be detected by training a neural network model.

Aozasa *et al.* (1993) Pneumothorax is a medical condition that results from an air leak between the chest wall and the lungs. Radiologists diagnose it, and confirmation is a time-consuming process. The Attention UNet and Attention UNet + ResNet34 models are used in semantic image segmentation to detect pneumothorax in radiological images, providing non-radiologists with confident results. Furthermore, the performance difference between traditional methods of training the models and modern methodologies such as snapshot callbacks and stochastic weighting average is analysed and compared. The motivation for an approach is that incorporating the modern approach gives the model an advantage when learning, allowing it to outperform the conventional approach and provide an early diagnosis of problems. Patients who received artificial pneumothorax had a significantly higher risk of developing pleural lymphoma (relative risk = 4.92, P0.05). There was no report that described the development of pleural neoplasias in patients with chronic empyema who underwent surgical resection of the pleural pyogenic membrane. Based on these findings, it is hypothesised that artificial pneumothorax caused chronic non-healing inflammation in the pleural cavity, leading to the development of pleural lymphoma.

Geva *et al.* (2015) We present a novel framework for automatically detecting pneumothorax abnormalities in chest radiographs. The proposed method employs a texture analysis approach in conjunction with supervised learning techniques. The proposed framework is divided into two major steps: first, a texture analysis process is used to detect local abnormalities. In the texture analysis procedure, labelled image patches are extracted, and local analysis values are incorporated into a novel global image representation. The global representation is used for image-level training and detection of abnormalities. The presented global representation is based on the distinctive shape of the lung and takes into account the features of typical pneumothorax abnormalities. A supervised learning process was used on both local and global data, which resulted in a trained detection system. The system was put through its paces on a set of 108 upright chest radiographs. Several cutting-edge texture feature sets were tried out (Local Binary Patterns, Maximum Response filters). The best configuration had a sensitivity of 81% and a specificity of 87%. The evaluation results are promising, establishing the current framework as a foundation for future improvements and extensions.

Melendez and Cordel (2015) This paper describes a method for classifying chest abnormal radiograph patterns such as pleural effusion, pnuemothorax, cardiomegaly, and hyperaeration using texture-based feature extraction and Support Vector Machines (SVM). A previous attempt used histogram values to achieve 97% and 87.5% accuracy for pleural effusion and pneumothorax, respectively, while image processing schemes achieved 70% and 73.33% accuracy for cardiomegaly and hyperaeration. They aimed to improve performance in classifying the aforementioned lung patterns, specifically cardiomegaly and hyperaeration, in this study. The developed system achieved accuracies of 96% and 99%, with sensitivities of 97% and 100%, for the cardiomegaly and hyperaeration cases, respectively, using texture-based features.

Wang *et al.* (2017) aim to create a "machine-human annotated" comprehensive chest X-ray database that depicts the realistic clinical and methodological challenges of dealing with tens of thousands of patients (similar to "ImageNet" in natural images). They have also used the ChestX-ray8 database to perform extensive quantitative performance benchmarking on eight common thoracic pathology classifications and weakly-supervised localization. The primary goal is to spur future efforts by promoting public datasets in this critical domain. Developing truly large-scale, fully automated, high-precision medical diagnosis systems remains a difficult task. ChestX-ray8 can help data-hungry deep neural network paradigms create clinically relevant applications such as common disease pattern mining, disease correlation analysis, and automated radiological report generation.

Blumenfeld *et al.* (2018) This study presents a computer-assisted diagnosis system for detecting pneumothorax (PTX) in chest radiographs that is based on pixel classification using a convolutional neural network (CNN). To overcome the large variability in the input images from various imaging sources and protocols, the proposed system employs a pre-processing step of lung field segmentation. Within each lung segment, suspected pixel candidates are extracted using a CNN classification. For the presence of PTX in each lung, the overall percentage of suspected PTX area was used as a robust global decision. The system was trained on 117 chest x-ray images with ground truth PTX region segmentations. The system was tested on 86 images and achieved AUC=0.95 in diagnosis accuracy. Overall, preliminary results are promising, indicating that CAD-based systems are improving their ability to detect findings in medical imaging with clinical-level accuracy.

Chan *et al.* (2018) Diseases are diagnosed more precisely by employing automatic image segmentation and attribute analysis. Due to varied image quality among medical devices, automatic medical image segmentation is problematic. To overcome this challenge, in this research work the automatic technique is used for image multiscale intensity texture analysis and segmentation. First, SVM is used to identify frequent pneumothorax in this paper. With the LBP, features are retrieved from lung pictures (local binary pattern). The SVM is then used to classify the pneumothorax. Second, for segmenting anomalous lung regions, the suggested automatic pneumothorax detection method is based on multiscale intensity texture segmentation by removing background and noises in chest images. The texture transformed from computing many overlapping blocks is based on the segmentation of anomalous regions. Sobel edge detection is used to identify the rib borders. Finally, the rib border is filled up and situated between the aberrant sections to obtain a full illness region. Accuracy, precision, and sensitivity for different patch sizes:- 5*5 is 85.8, 83.6, 87.4; for 7*7 is 84.1, 84.4, 85.5; for 9*9 is 83.2, 85.1, 82.3 and for 11*11 is 81.1, 86.2 , 81.6 respectively.

Taylor *et al.* (2018) When tested on an internal test set, built automated models with a high AUC, were sensitive to big and intermediate pneumothorax, and had a good specificity. The high specificity model (specificity 0.97) yielded a PPV of 12.5 percent for the scenario of 1% pneumothorax prevalence (containing mild, moderate, and big pneumothorax). This evaluation is on what is needed to prioritize low-prevalence discoveries.

Yao *et al.* (2018) The study, present a method for dealing with all these practical constraints by employing a novel architecture that learnt at multiple resolutions while generating saliency maps with minimal supervision. Furthermore, they parameterize the Log-Sum-Exp pooling function with a learnable lower-bounded adaptation (LSE-LBA) to incorporate a sharpness prior and to improve the handling of localising abnormalities of varying sizes used only image-level labels. Using this method for interpreting chest x-rays, they established the state of the art on 9 abnormalities in the NIH's CXR14 dataset while producing the highest resolution with AUC of 80.5 percent that is saliency maps to date.

Baltruschat *et al.* (2019) ResNet-50-large-meta is a patient's-data adapted model architecture. The ResNet-50 model serves as the foundation for the architecture. Because the input size was increased, they added a max-polling layer after the first three Residual Blocks. Furthermore, at the end of the model, a fused image and patient features to incorporate patient information. 70% of the data is used for training, 10% for validation, and 20% for testing. While working with deep learning, working without a validation set or cross-validation can easily result in over-fitting. The results are computed with each fold on the test set and then averaged, as a result AUC obtained in this study was 84.0 percent.

GooBen *et al.* (2019) suggested a method for detection and localization of pneumothorax in chest x-ray images, they have studied the characteristics of various deep learning techniques:

- Convolutional neural network
- Multiple instance learning
- Fully convolutional network

After performing five-fold cross-validation on a dataset having 1003 chest X-ray images. The three techniques had AUCs of 0.96, 0.93, and 0.92, respectively, according to ROC analysis. They looked at how these approaches perform in terms of classification and localization, as well as a combination of the three strategies. Using the average AUC as a performance criterion, they were able to achieve fairly consistent results across all techniques, with values ranging from 0.92 to 0.96. These data show that the algorithms have a very strong overall performance. They have achieved the best performance in terms of AUC using CNN but MIL and FCN gave higher confidence for localization, which could guide radiologists by looking at the image region responsible for the network's decision and increasing the trust in the proposed deep learning architecture.

Hooda *et al.* (2019) This research work provides a shallow learning-based strategy for segmenting lung fields from chest radiographs that use 40 radio mic variables. The output is refined using the distance regularised level set evolution (DRLSE) approach and additional post-processing procedures. The suggested approach is trained and evaluated on the JSRT dataset, which is freely available. The testing results show that the suggested method outperforms other lung field segmentation (LFS) methods and is similar to state-of-the-art deep learning-based LFS methods. The proposed method has an average accuracy of 98.12 percent and an average overlap score of 94.50 percent, according to the testing data. The experimental results show that the suggested method outperforms the previous shallow-learning-based LFS algorithms in terms of performance. They have chosen the Shallow learning-based model over the deep learning model because deep learning has some limitations like it needs more storage, more computational power, and GPUs.

Irvin *et al.* (2019) have presented chest expert, a huge dataset that contains 224316 chest radiographs of 65240 patients. They have created a labeler that can automatically discover 14 observations in radiology reports, incorporating the inherent uncertainty in radiograph interpretation. They have a look into various methods for applying uncertainty labels to train convolutional neural networks that output the probability of these observations given the available frontal and lateral radiographs. They discovered that different uncertainty techniques are beneficial for different pathologies based on a validation set of 200 chest radiography scans manually annotated by three board-certified radiologists. The performance of their top model is then compared to that of three additional radiologists in the detection of five pathologies using a test set of 500 chest radiography studies annotated by a consensus of 5 board-certified radiologists. The model ROC and PR curves for Cardiomegaly and Pleural Effusion were above the three radiologist operational points. They make the dataset available to the public as a standard benchmark for evaluating chest radiograph interpretation models' performance.

Sze -to and Wang (2019) suggested a pneumothorax classification model that evaluated the power of transfer learning. A 122-layered deep neural network called tCheXNet was proposed, it was named so as it used the pre-trained model CheXNet to initialise the model weights. The CheXpert dataset was chosen for this study and 94,948 CXRs were used for training this model, only 13911 of which belonged to the pneumothorax class; this class-imbalance issue was resolved using the weighted binary cross entropy loss function. The proposed framework performed well on a test set consisting of 7 pneumothorax images and 195 normal CXRs, with an AUC of 70.8 percent.

Tang *et al*. (2019) The proposed architecture was formed by deep neural networks that learned by competing and collaborating with one another to model on the content structure of normal chest X-rays. If a chest X-ray image is normal in the testing phase, the architecture model learnt well and reconstruct the content well; if it is abnormal, the model performs poorly in its reconstruction because the content is unseen in the training phase. As a result, abnormal chest X-rays can be distinguished from normal ones. Quantitative and qualitative experiments show that their approach was effective and efficient, with an AUC of 0.841 achieved on the challenging NIH Chest X-ray dataset in a one-class learning setting, with the potential to reduce radiologists' workload.

Bharati *et al.* (2020) By merging VGG, data augmentation, and spatial transformer network (STN) with CNN, a new hybrid deep learning framework is introduced. VGG Data STN with CNN is the name given to this novel hybrid technique (VDSNet). Jupyter Notebook, Tensorflow, and Keras are utilized as implementation tools. A dataset of NIH chest X-ray images obtained from the Kaggle repository is used to test the proposed model. The dataset is taken into account in both its full and sample forms. VDSNet beats existing techniques for both whole and sample datasets in terms of measures like precision, recall, F0.5 score, and validation accuracy. Vanilla grey, vanilla RGB, hybrid CNN and VGG, and modified capsule network have accuracy scores of 67.8 percent, 69 percent, and 73 percent respectively for the case of the whole dataset.

Shreyas *et al.* (2020) Pneumothorax is a medical condition that results from an air leak between the chest wall and the lungs. Radiologists diagnose it, and confirmation is a time-consuming process. The Attention UNet and Attention UNet + ResNet34 models are used in semantic image segmentation to detect pneumothorax in radiological images, providing non-radiologists with confident results. Furthermore, the performance difference between traditional methods of training the models and modern methodologies such as snapshot callbacks and stochastic weighting average is analysed and compared. The motivation for an approach is that incorporating the modern approach gives the model an advantage when learning, allowing it to outperform the conventional approach and provide an early diagnosis of problems.

Sze-to *et al.* (2020) The application of image search based on deep features obtained by DNNs to detect pneumothorax in chest X-ray pictures was investigated in this study. All chest X-ray images were first tagged with a feature vector using image search as a classifier. The majority voting of the top k retrieved chest X-ray images was then employed as a classifier given a query chest X-ray image. In this research work, they have looked through a database of over 550,000 chest X-ray scans. This research shows that image search using Auto Thorax -Net features can achieve high identification performance, paving the way for real-world application. In 194,608 photos (pneumothorax and normal), they have achieved 92 percent AUC accuracy for a semi-automatic search, and 82 percent AUC accuracy for a fully automated search in 551,383 images (normal, pneumothorax, and many other chest disorders).

Sze-to and Tizhoosh (2020) The use of image search based on deep pre-trained features in classifying pneumothorax among more than half a million chest X-rays were investigated in this work. Experiments revealed that a content-based medical picture retrieval system, such as image search, could be a practical and cost-effective solution. It is the first work to show that deep pre-trained features can be employed for CBIR of pneumothorax in half a million chest X-ray pictures. In the profession of diagnostic radiology, image search can be used as a semi-automatic or fully automated method. Image search results, when compared to traditional classification, may be more clinically useful because they are backed up by reports and histories of clearly diagnosed cases, effectively serving as a "second opinion" for diagnostic purposes, a factor that may provide more confidence in a reliable diagnosis.

Wang *et al.* (2020) presented “Automated segmentation and diagnosis of pneumothorax on chest X-rays with fully convolutional multi-scale ScSE Densenet”. They aimed to form a large chest X-ray dataset for pneumothorax with the assistance of pixel-level annotation and then train an automatic segmentation then finally diagnose pneumothorax on time and appropriately. The dataset consists of 11,051 front CXR out of which 5,566 were cases of PTX and 5,485 were cases of Non-PTX. The proposed algorithm outperforms the state-of-the-art segmentation algorithms in terms of mean pixel-wise accuracy (MPA) of 0.93 0.13 and dice similarity coefficient (DSC) of 0.92 0.14, and achieves competitive performance on the diagnostic accuracy of 93.45 percent and F1-score of 92.97 percent, in step with the experiment.

Cha *et al.* (2021) The purpose of this research is to increase the accuracy of a computer-aided diagnostic approach that medical professionals can employ as a backup tool. They have developed an attention-based transfer learning framework for detecting thoracic disease in chest X-ray images in this research. As a feature extractor, they collected features from three types of pre-trained models: ResNet152, DenseNet121, and ResNet18. For a new job, they rewrote the classifier and used the attention method as a feature selector. As a result, the suggested method achieved 96.63 percent accuracy, 0.973 percent F-score, 96.03 percent AUC, 96.23 percent precision, and 98.46 percent recall, respectively.

Cho *et al.* (2021) The focus of this research was to see how well fully connected tiny artificial neural networks (ANNs) using a simple training technique, the Kim-Monte Carlo algorithm, could determine the location of a pneumothorax in chest X-rays. The training and test sets were made up of 1,000 chest X-ray images with pneumothorax that were randomly selected from the NIH (National Institutes of Health) public image collection. For pneumothorax localization, each X-ray image having pneumothorax was segmented into 49 boxes. The area under the receiver operating characteristic (ROC) curve (AUC) for each of the boxes in the test set's chest X-ray pictures was 0.882, with sensitivity and specificity of 80.6 percent and 83.0 percent, respectively.

Choudhary and Hazra (2021) They used a deep learning artifice in this paper to bridge the semantic gap that exists between low-level information that is captured by imaging devices as well as high-level information preserved by humans. They have proposed a convolutional neural network architecture comprised of five layers: a convolutional layer, an activation layer, a pooling layer, and a fully connected layer, followed by a Softmax activation that provides the probability of output for each genre. The proposed method was tested on a well-known publicly available NIH dataset called Chest X-Ray 14 and have produced state-of-the-art results of 83.671 percent (scratch CNN) in 2021, which was significantly higher than the other methods.

Folke *et al.* (2021) proposed a method to diagnose pneumothorax with the help of Bayesian teaching a computational model and an integrative framework as well as a formal account of explanation rooted in the cognitive science of human learning. It is found that the medical experts exposed to explanations created by Bayesian teaching successfully predicted the AI's diagnostic decisions and more likely agreed with it. On trail 6 out of 8 on average were predicted correctly.

Iqbal *et al.* (2021) first evaluation of multiple solutions for class imbalance (CI) problems using a medical picture dataset before providing such a system for automatic identification of pneumothorax utilizing highly unbalanced data. They presented a VDV model which outperforms a single data-level-under-bagging based ensemble with a single CNN architecture as a fixed feature extractor, according to their findings. For the SIIM pneumothorax dataset, the proposed VDV system achieved 85.17 percent recall and 86.0 percent AUC. For the RS-NIH dataset, the random split of data achieves 90.9 percent recall and 95.0 percent AUC. The patient-wise split of data yielded 85.45 percent recall and 77.06 percent AUC. The findings they have obtained on both datasets, SIIM Pneumothorax and RS-NIH, are better, confirming the effectiveness of their proposed framework, for any unbalanced dataset, with a few tweaks such as utilizing a different CNN architecture for feature extraction and a different input picture resolution. They have also suggested that, using this paradigm for larger datasets in the future, such as the entire NIH Chest X-ray-14 dataset. A segmentation model based on the SIIM dataset can also be constructed, which will aid radiologists in correctly diagnosing diseases.

Tolkachev *et al.* (2021) In recent years, deep learning has significantly improved the computer-aided diagnosis of pneumothorax. innovations and the first public competition for pneumothorax diagnosis with 19 team members manually annotating 15257 chest X-rays radiologists. One of the best frameworks is described in this study.

The structure looks into The Unit convolutional neural network's advantages when combined with ResNet34, SE-ResNext50, SEResNext101, and DenseNet121 are a few examples of different backbones. The publication provides detailed data and instructions for the framework application different pre-and post-processing processes, as well as augmentation. The framework performed 0.8574 assessed in terms of the dice factor.

Wang *et al.* (2021) Although computer-aided diagnosis of pneumothorax on chest X-ray has been extensively studied as a prerequisite for timely cure, it is still not satisfactory to achieve highly accurate results. An image classification algorithm based on deep convolutional neural network (DCNN) is proposed in this paper for high-resolution medical image analysis of pneumothorax X-rays, which includes a Network In Network (NIN) for data cleaning, random histogram equalisation data augmentation processing, and a DCNN. The experimental results show that the proposed method can effectively increase the correct diagnosis rate of pneumothorax, with an Area under Curve (AUC) of 0.9844 on ZJU-2 test data and 0.9906 on ChestX-ray14, respectively.

Yu *et al.* (2021) Background and Goal: Chest X-Ray (CXR) imaging is a critical first-line diagnostic tool for pneumothorax, a potentially fatal condition. The main treatment for pneumothorax is the insertion of a chest tube to drain air. The goal of this study was to create an artificial intelligence (AI) model for detecting pneumothorax on CXR. In addition, we developed an auxiliary AI model for detecting an inserted chest tube. Although a pneumothorax can be fatal, the presence of a chest tube indicates that the pneumothorax is being treated. A fresh pneumothorax is emergent in the clinical scenario, but a pneumothorax after chest tube treatment is not. As a result, They proposed a combined AI model to differentiate between the newly onset emergent condition and the condition after treatment. They trained a neural network on MobileNet V2 to do binary classification of whether or not a pneumothorax occurred. Three radiologic technologists labelled the dataset, which was then reviewed by a radiologist. The same methods were used to create the AI model for the chest tubes. On the test dataset, the accuracy for pneumothorax and chest tube identification was 92.19% and 98.22%, respectively. The areas under the receiver operating characteristic curve for pneumothorax were 0.9638 and 0.9968, respectively (chest tube). Each image's inference time ranged from 0.21 to 0.6 second.

Luo *et al.* (2022) Pneumothorax is a potentially fatal disease that necessitates prompt diagnosis and treatment. A chest X-ray examination is the first choice for diagnosing pneumothorax in the clinic. However, when the lesion area is only composed of a small amount of air, it is difficult to diagnose pneumothorax using only frontal chest X-ray imaging. As a result, a method is proposed for pneumothorax diagnosis neural network based on feature fusion, which combines frontal and lateral X-ray information. There are two inputs and three outputs in this network. The frontal chest X-ray image and the lateral chest X-ray image are the two inputs. The classification results of the frontal chest X-ray image, the classification results of the lateral chest X-ray image, and the classification results integrating the characteristics of the fused frontal chest X-ray image and lateral chest X-ray image are the three outputs. In the pneumothorax recognition model, the algorithm considers the vanishing gradient problem and introduces the residual block to address it. They used channel attention mechanisms to improve the model's performance due to the large number of channels in this model. Their comparative experiments show that neural network fusion of frontal and lateral chest image features can outperform the single task model in terms of accuracy. Our pneumothorax model can achieve high recognition accuracy using only image-level annotation.

Malhotra *et al.* (2022) This research work proposes a deep learning neural network model for detecting pneumothorax areas in chest X-ray images. Mask Regional Convolutional Neural Network (Mask RCNN) framework and transfer learning are used in the model, with ResNet101 serving as the backbone feature pyramid network (FPN). The suggested model was trained on a pneumothorax dataset created in collaboration with the American College of Radiology by the Society for Imaging Informatics in Medicine (SIIM-ACR). The suggested MRCNN model based on ResNet101 as an FPN is compared to the traditional model based on ResNet50 as an FPN in this paper. In comparison to the conventional model based on ResNet50 as an FPN, the proposed model has a lower class loss, bounding box loss, and mask loss. With 10 and 12 epochs, respectively, both models were simulated with a learning rate of 0.0004 and 0.0006.

Wang *et al.* (2022) have proposed a two-stage deep learning method for automatic pneumothorax segmentation of chest X-ray images. Firstly, the screening X-ray is performed to detect the presence or absence of pneumothorax. If the pneumothorax is detected in the first stage then. In the second stage the accurate segmentation of pneumothorax is performed. Through comparisons, it is seen that this proposed method is more stable and more accurate than single-stage methods and state-of-other methods. Due to higher accuracy in image segmentation, the proposed method can be used by radiologists or physicians for prompt and precise diagnosis of pneumothorax and good treatment planning. The average dice score of 0.9827, accuracy of 0.9925, sensitivity of 0.9802, and specificity of 0.9960 was obtained in this research work'.

Chapter-3 MATERIALS AND METHODS

In the medical imaging domain, a computer-assisted evaluation is enabling higher image interpretations which have always been challenging. In the image understanding arena, recent breakthroughs in machine learning, especially deep learning, have made significant progress in assisting with identifying, classifying, and quantifying structures in medical data. The gains are supported using hierarchical visual features learned directly from data instead of handwritten features largely generated and supported by domain-specific expertise. As a result, deep learning has now been fast establishing itself as a state-of-art platform, generating improved leads for a range of medical applications.

In this section, we will discuss the approaches, strategies, and tools that were selected for the work, including concepts and algorithms that are essential for carrying out this study and have also been examined on the datasets to get effective results. Pneumothorax can be caused by a blunt chest injury, lung disease, or, most terrifyingly, it can occur for no apparent reason at all. A collapsed lung can be life-threatening in some cases. Pneumothorax is usually diagnosed on a chest x-ray by a radiologist, and it can be difficult to confirm at times. A reliable AI algorithm for detecting pneumothorax would be beneficial in a variety of clinical scenarios. AI could be used to prioritize chest radiograph interpretation or to provide non-radiologists with a more confident diagnosis.

3.1 Material

3.1.1 Dataset

The data used to validate our proposed model in the context of medical diagnosis and on which the experiment is conducted is publicly available on Kaggle with the link: https://www.kaggle.com/code/gpiosenka/efficientnet-b2-f1-score-85

The total input size is 812.32 MB. The data is in the form of a CSV file which here is named train_data.csv and has three columns containing the image id, image name (file name), and the target value „0" and „1", where „0" denotes the absence of pneumothorax and „1" denotes the presence of pneumothorax in respective images. The chest x-ray images are present in a folder named small_train_data_set and 2027 are total images in a png file format with the size (1024, 1024). The training dataset consists of 1824 total images, 1437 images with the presence of pneumothorax, and 387 images with the absence of pneumothorax. The validation and testing dataset comprised 101 images in each set and the testing data contains 102 images in total out of which 80 images are with the presence of pneumothorax and 22 images with no pneumothorax.

The dataset is first of all uploaded in Drive, saved in My Drive in the folder named as archive (2) and further the subdivision is done. The images are in small_train_data_set and its detail like image id and file name are in a folder named as train_data.csv.

Dataset location in the drive

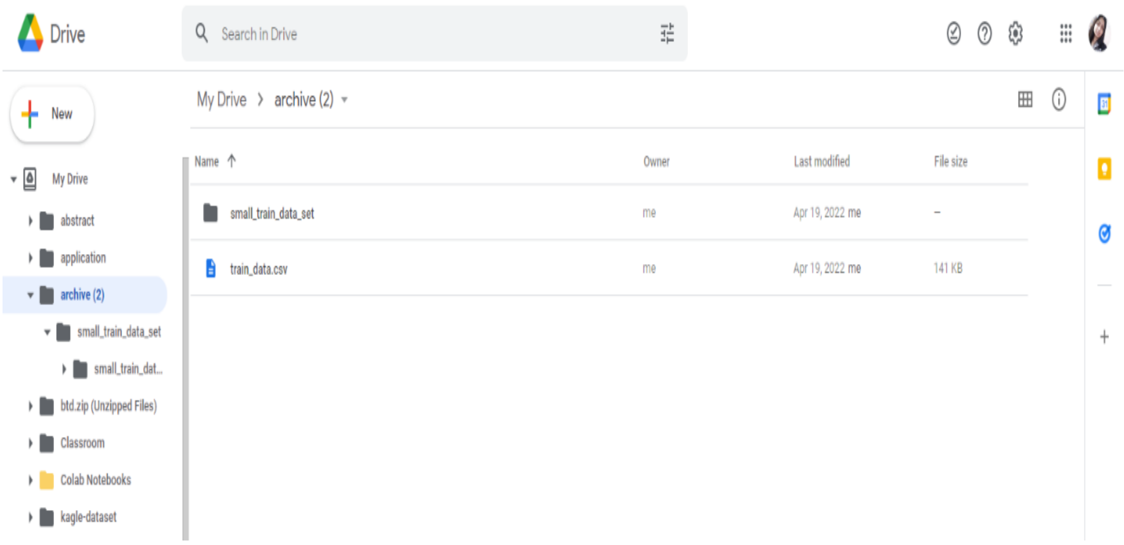


Figure 3.1: Dataset location in the drive

Visualization of image data

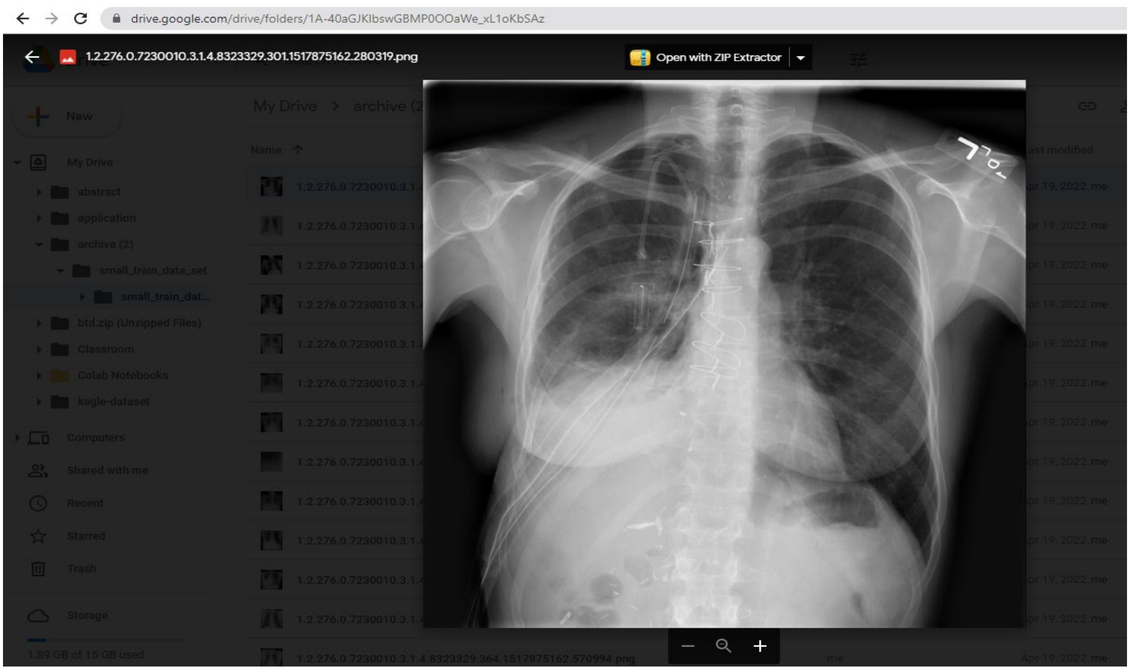
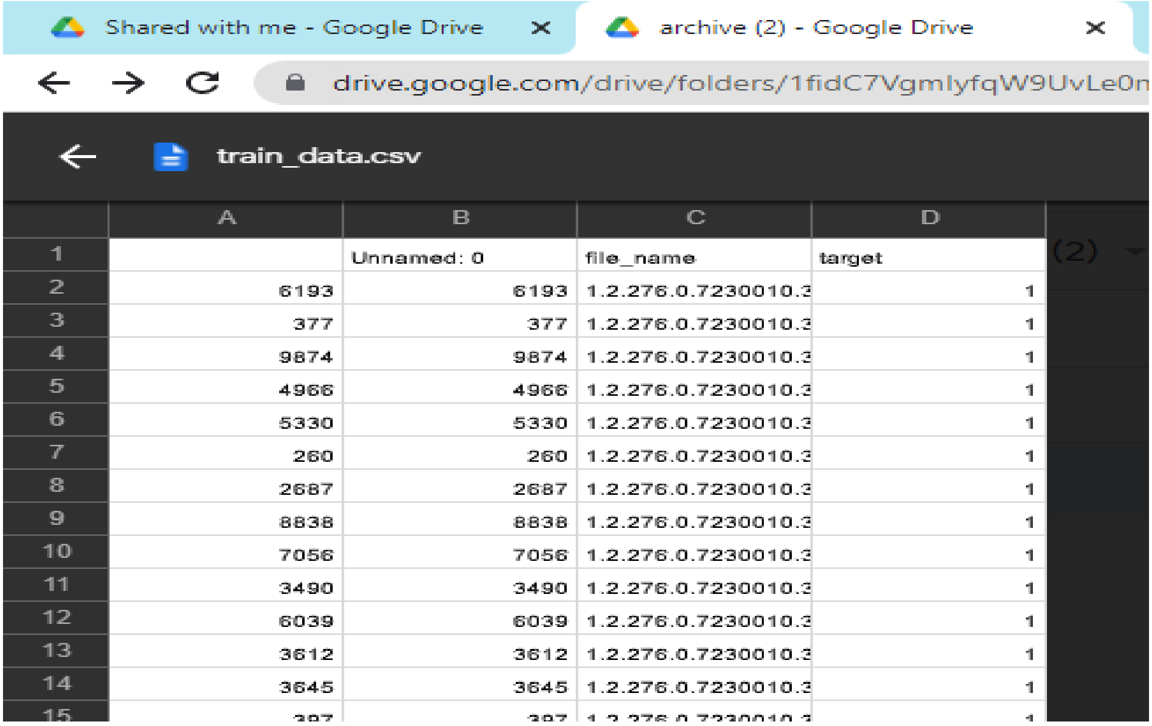


Figure 3.2: Visualization of image data

Visualization of a csv file



	A	B	C	D
1		Unnamed: 0	file_name	target
2	6193	6193	1.2.276.0.7230010.3	1
3	377	377	1.2.276.0.7230010.3	1
4	9874	9874	1.2.276.0.7230010.3	1
5	4966	4966	1.2.276.0.7230010.3	1
6	5330	5330	1.2.276.0.7230010.3	1
7	260	260	1.2.276.0.7230010.3	1
8	2687	2687	1.2.276.0.7230010.3	1
9	8838	8838	1.2.276.0.7230010.3	1
10	7056	7056	1.2.276.0.7230010.3	1
11	3490	3490	1.2.276.0.7230010.3	1
12	6039	6039	1.2.276.0.7230010.3	1
13	3612	3612	1.2.276.0.7230010.3	1
14	3645	3645	1.2.276.0.7230010.3	1

Figure 3.3: Visualization of a csv file

3.1.2 Feature Extraction

Image data feature extraction represents the interesting parts of an image as a compact feature vector. Previously, this was accomplished using specialized feature detection, extraction, and matching algorithms. Deep learning is now widely used in image and video analysis, and it is well known for its ability to take raw image data as input and skip the feature extraction step. Computer vision applications like image registration, object detection, classification, and content-based image retrieval all require the effective representation of image features - either implicitly by the first layers of a deep network, or explicitly by employing some of the long-standing image feature extraction techniques. In our proposed study we are going to use pre-trained model that is something known as „Transfer Learning".

3.1.2.1 Transfer Learning

Transfer learning in deep learning is the use of a previously trained model on a recent feature. In transfer learning, a machine uses the information gained from previous work to improve its prediction about a new task. Through transfer learning, the expertise of a deep learning model that has already been trained is applied to an unrelated but closely related problem. Some of the best models on ImageNet's image recognition tasks namely VGG, Inception, and ResNet, are easily accessible using Keras.

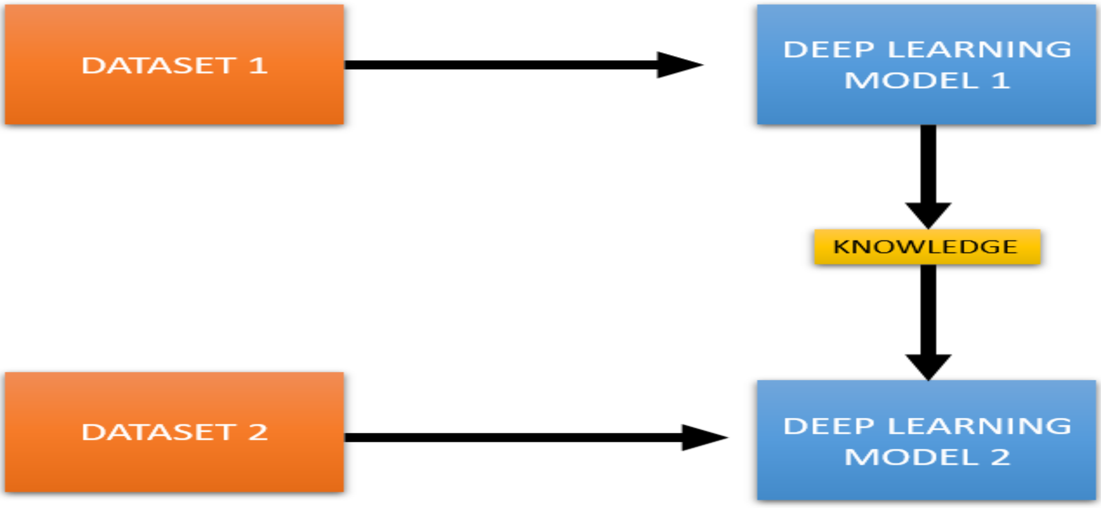


Figure 3.4: Flow chat for transfer learning

The following workflow is the most standard application of transfer learning within the context of deep learning:

- Add layers to a model that has already been trained.
- Freeze them to forestall losing any of the info they contain during upcoming training sessions.
- On top of the frozen layers, add some fresh, trainable layers. On a brand-new dataset, they will discover a way to forecast using the previous features.
- On your dataset, train the new layers.

Computer Vision Pre-Trained Models are a saved network that was previously trained on a large dataset, typically on a large-scale image classification task, and is referred to as a pre-trained model. A number of pre-trained networks are available that can be used for a variety of computer vision applications, including image production, neural style transfer, image classification, image captioning, anomaly detection, and more. For example- ResNet50, VGG19, Inceptionv3, and EfficientNet. But for our proposed research work ResNet-50 a pretrained model will be used for feature extraction.

ResNet-50

A convolutional neural network with 50 layers is called ResNet-50. A common neural network that serves as the foundation for many computer vision applications is called ResNet, short for Residual Networks. The main innovation with ResNet was that it enabled us to train very deep neural networks with more than 50 layers

Skip Connections

All other algorithms like AlexNet, VGG, etc. are trained on the output „Y" but, ResNet is trained on F(X). In simpler words, ResNet tries to make F(X)=0, so that Y=X.

A direct link known as a skip connection bypasses some model layers. This skip connection causes the output to differ. Input X is multiplied by the layer weights in the absence of the skip connection, and then a bias term is added.

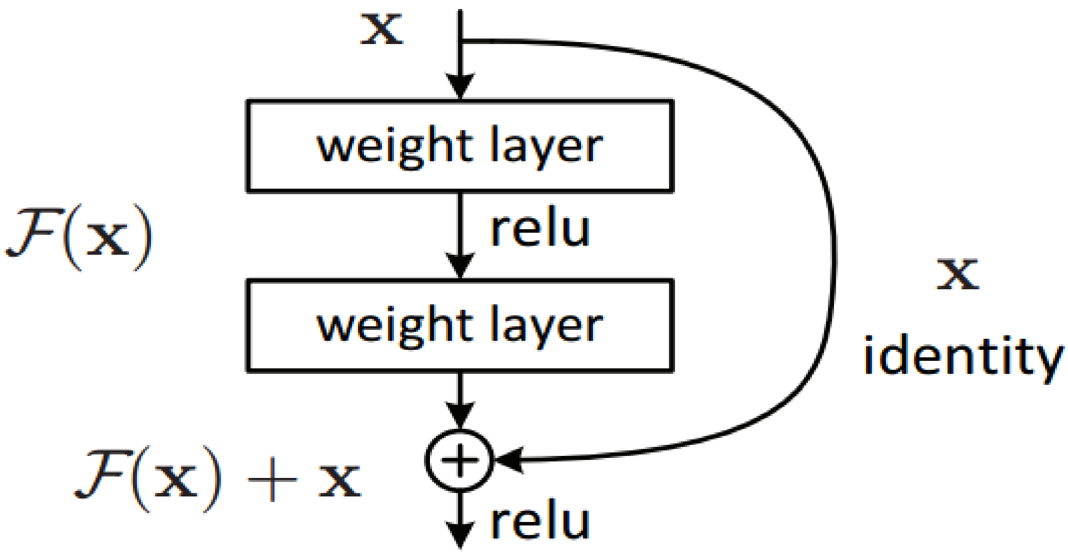


Figure 3.5: the residual block of ResNet-50

Need for skip connection: It is the addition of original input to the output of the convolutional block. Due to some major drawbacks in the CNN training process "Skip Connection" came into existence-

- They diminish the matter of fading (vanishing) gradient during backpropagation for weight optimization, by allowing this alternate shortcut track to flow through.
- They permit the model to find out an identity function that ensures that the upper layer will perform a minimum of pretty much as good as a lower layer, and not inferior to it.

There are two kinds of blocks in ResNet-50:

- **Identity block**- If there is no change in the input or output dimensions, Identity Block is adopted.

input size == output size

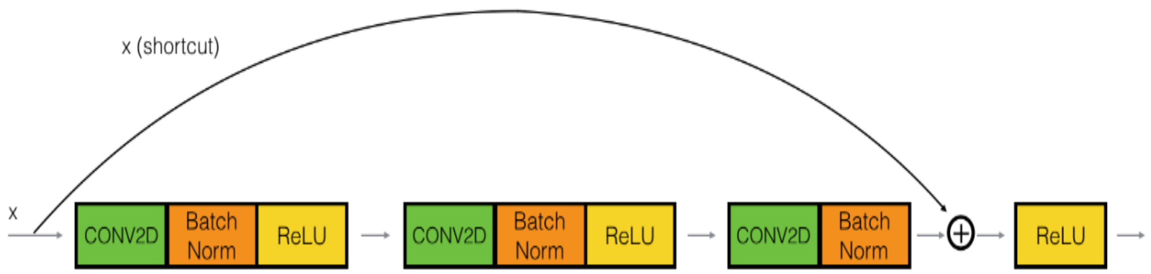


Figure 3.6: Identity Block

- **Convolutional Block-** Convolutional blocks are virtually identical to identity blocks, but a convolutional layer is used in a shortcut path to simply adjust the size so that the input and output sizes match.

  input size != output size

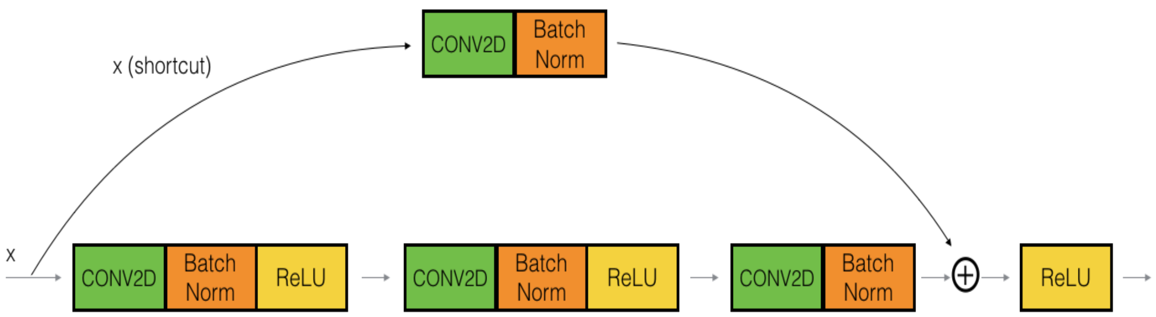


Figure 3.7: Convolutional Block

The value of „x" is added to the output layer only if,

Input size == Output size, that is the dimensions are same

There are two approaches to equalizing the input and output sizes:

- Performing 1*1 convolutions
- Padding the input volume.

Now the size of the output layer can be calculated by using the formula;

$[\ \{(n + 2*p – f) / s\} + 1]^2$, here n = input size of image

p = padding, s = stride

f = number of filters

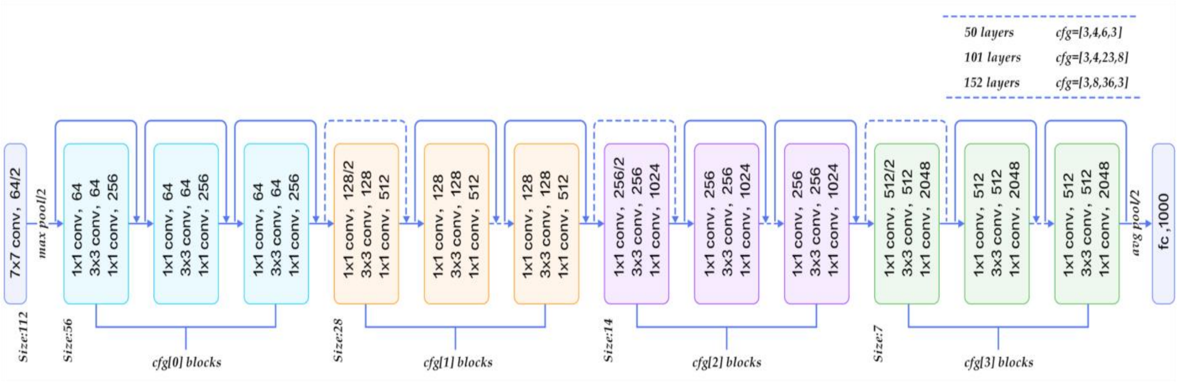


Figure 3.8: ResNet-50 Architecture

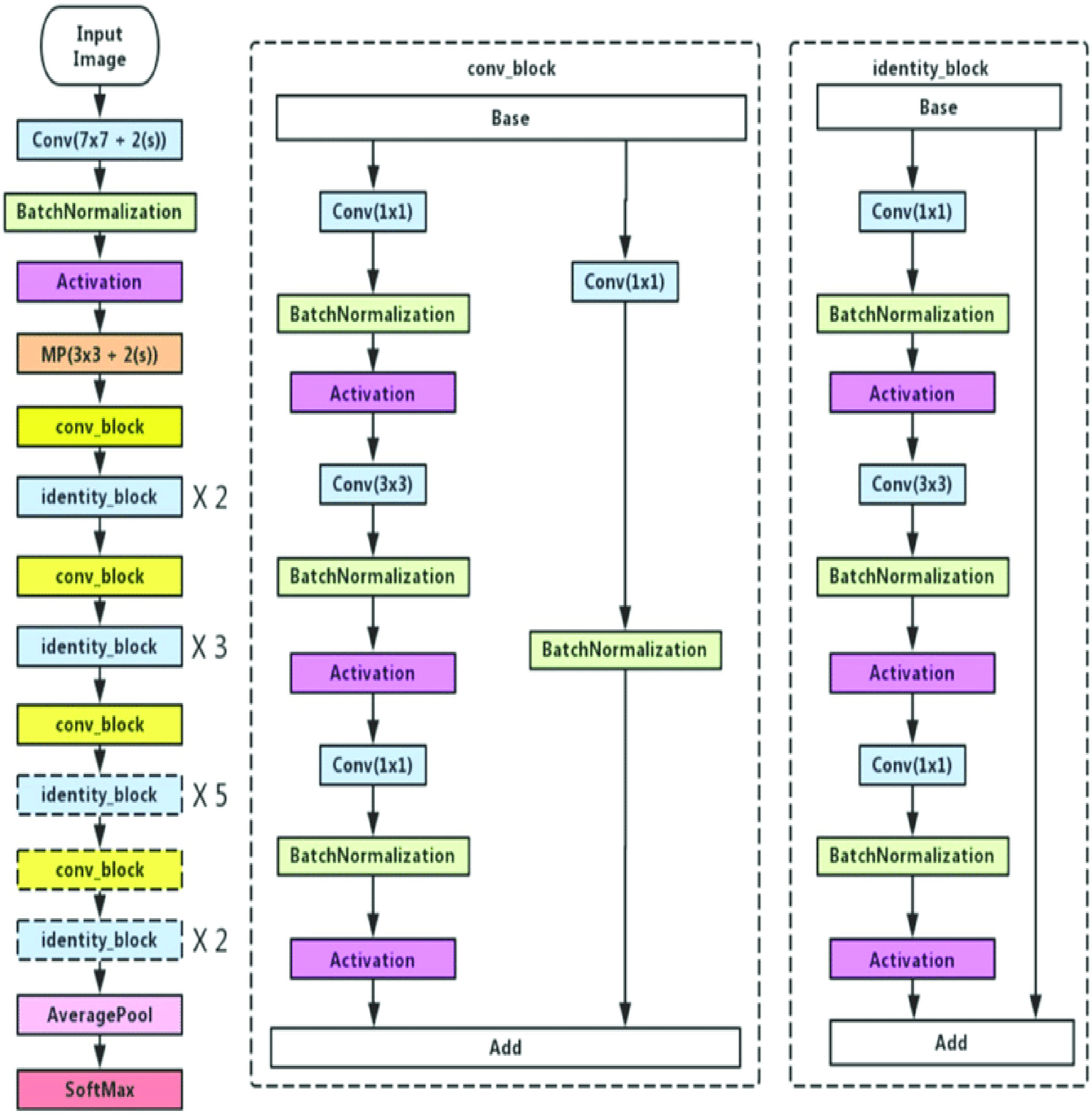


Figure 3.9: Convolutional and identity blocks of ResNet-50

Different types of ResNet architectures summary:

layer name	34-layer	50-layer	101-layer
conv1	$7\times7,64$, stride 2		
	3×3 max pool, stride 2		
conv2_x	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
conv3_x	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$
conv4_x	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$
conv5_x	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
	average pool, 2048-d fc		

Figure 3.10: different ResNet architectures

The very next thing to be done after extracting features from an image is to classify whether an image belongs to class „0" or „1". So, for this purpose all we need is a classifier.

3.1.3 Classifier

One of the most popular applications of AI technology is the use of machine learning classifiers to automatically evaluate data, expedite procedures, and obtain insightful information. An algorithm known as a classifier in machine learning automatically arranges or categorizes data into one or more of a set of "classes." One of the most prevalent instances is an email classifier, which examines emails and filters them according to whether they are spam or not. Machine learning (ML) which is also a subset of artificial intelligence (AI) that enables software applications to predict outcomes more accurately at without explicitly programming them to do so. Machine learning algorithms predict new output values by using historical data as input. Supervised machine learning is used for binary classification as supervised learning algorithms perform these tasks excellently.

Various types of Classification algorithms depending on our needs and our data are listed below:

- Decision Tree
- Naive Bayes Classifier
- K-Nearest Neighbors

- Support Vector Machines
- Artificial Neural Networks

The proposed study is related to binary classification of chest x-ray images and SVM gives state-of-the-art performance on such problems. Thus, SVM will be used by us.

3.1.3.1 Support Vector Machines (SVM)

One of the most well-liked supervised learning algorithms, Support Vector Machine, or SVM, is used to solve Classification and Regression problems. However, it is largely employed in Machine Learning Classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name of this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method. The diagram below shows two different categories that are classified using hyperplane or decision boundary:

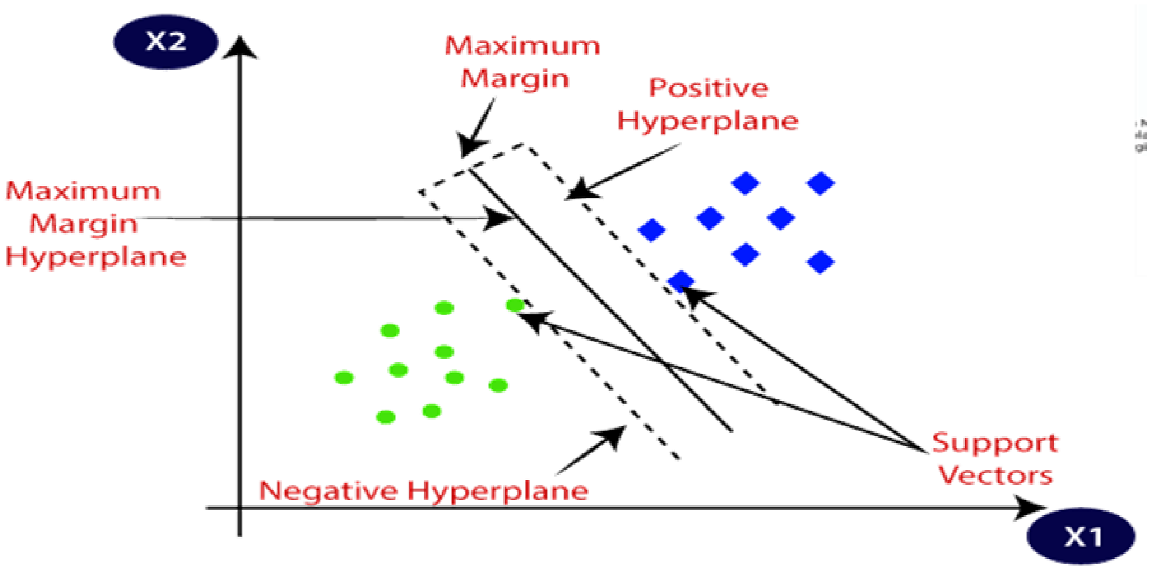


Figure 3.11: Classification visualization of SVM

The SVM algorithm is used in face identification, image classification, text categorization, etc.

Hyperplane and Support Vectors in SVM algorithm:

Hyperplane: In n-dimensional space, there may be several lines or decision boundaries used to divide classes; however, the optimal decision boundary for classifying the data points must be identified. The hyperplane of SVM is a name for this ideal boundary. The dataset's features determine the hyperplane's dimensions, therefore if there are just two features (as in the example image), the hyperplane will be a straight line. Additionally, if there are three features, the hyperplane will only have two dimensions. The largest margin, or the space between the data points, is always used to form a hyperplane.

Support Vectors: Support vectors are the data points or vectors that are closest to the hyperplane and have the greatest influence on where the hyperplane is located. These vectors are called support vectors because they support the hyperplane.

The SVM is of two types

Linear SVM: Linear SVM is used for data that can be divided into two classes using a single straight line. This type of data is termed linearly separable data, and the classifier employed is known as a Linear SVM classifier.

Non-Linear SVM: Non-Linear SVM is used for non-linearly separated data. If a dataset cannot be classified using a straight line, then it is considered non-linear data, and the classifier employed is known as a Non-linear SVM classifier.

3.1.4 Optimizers

Optimizers are algorithms or methods that change the characteristics of the neural network, such as weights and learning rates, to reduce losses. Adam is the best optimizer as it outperforms on dataset.

3.1.4.1 Adam

An algorithm for gradient descent optimization is called Adaptive moment estimation. When dealing with complex problems involving a lot of data or factors, the strategy is incredibly effective. It is effective and uses little memory. It combines the "gradient descent with momentum" algorithm and the "RMSP" algorithm, intuitively.

The Adam optimizer combines the following two methods of gradient descent.

Momentum

By using the "exponentially weighted average" of the gradients, this approach is used to speed up the gradient descent algorithm. The technique converges quicker towards the minima when averages are used.

$W_{t+1} = w_t - \alpha m_t$

where,

$m_t = \beta\, m_{t-1} + (1-\beta)\, [\delta L / \delta w_t]$

m_t = the sum of slopes at time t [at the moment] (Initially, m_t is taken as zero)

m_{t-1} = the sum of the gradients at time t-1.

w_t = weights at time t.

w_{t+1} = weights at time t+1.

α_t = the rate of learning at time t.

δL = the loss function's derivative.

δw_t = weights at time t's derivative.

β = moving average parameter is (constant, 0.9).

RMSP (Root Mean Square Propagation)

An adaptive learning system called Root Mean Square Prop, or RMSprop, aims to enhance AdaGrad. It uses the "exponential moving average" as opposed to AdaGrad's method of computing the cumulative sum of squared gradients.

$W_{t+1} = w_t - \alpha_t / (v_t + \varepsilon)^{1/2} * [\delta L / \delta w_t]$

where,

$v_t = \beta\, v_{t-1} + (1-\beta) * [\delta L / \delta w_t]^2$

W_t = weights at time t

W_{t+1} = weights at time t+1

α_t = learning rate at time t

∂L = derivative of Loss Function

∂W_t = derivative of weights at time t

V_t = sum of square of past gradients. [i.e. sum($\partial L/\partial Wt-1$)] (initially, $V_t = 0$)

β = Moving average parameter (constant, 0.9)

ε = A small positive constant (10^{-8})

An aspect of Adam Optimizer's mathematically

Using the formulas from the two methods mentioned above, we obtain is

$m_t = \beta_1 m_{t-1} + (1-\beta_1) * [\delta L / \delta w_t] * v_t$

$= \beta_2 v_{t-1} + (1-\beta_2) * [\delta L / \delta w_t]^2$

ϵ = a small +ve constant to avoid 'division by 0' error when (v_t -> 0), (10^{-8}).

3.1.5 Batch Normalization

Batch normalization is a technique used to scale the input data passed between layers. For each batch passed through a neural network, batch normalization fixes the means and variances of each layer‟s inputs. The main benefit of batch normalization is that it increases the stability of a neural network by limiting the range of values in each layer, thus also limiting the internal covariate shift or change in the distribution of the network activations due to weight updates. As a result, batch normalization allows for the use of higher learning rates and can allow the optimization algorithm to converge faster.

3.1.6 Technique to avoid overfitting

Overfitting is a common issue that arises not only when training deep learning models, but also when training machine learning models in general. Overfitting happens when a model basically memorizes training examples and appears to perform well on training data. However, it is unable to apply what it has learned to previously unseen testing or validation data. In practice, for example, The original dataset will be divided into sections for any machine learning experiment or research investigation. There are three sets: a training set, a validation set, and a testing set that is not touched until the end.

- **Weight decay-** is a technique in which the loss function is modified to control the magnitudes of the weights and discourage the network from assigning extremely high values to specific weights. In practice, this technique can help to reduce overfitting by generating simpler models. Weight decay is classified into two types, which are commonly referred to as L2 and L1 regularization.

- **Regularization technique-** L2 regularization modifies the original loss function by adding a term with the sum of the squares of the weights multiplied by a constant. In weight decay λ is the hyper-parameter that controls the regularization strength. Larger values of , will result in simpler models, while smaller values will result in more complex models that may be prone to underfitting. This is a hyperparameter that must be tuned through trial and error for various deep learning tasks. L1 regularization is similar to L2 regularization, except that when defining the new loss function, the sum of the squares of the weights is replaced with the sum of the absolute values of the weights.

- **Data Augmentation-** The process of massively raising the quantity of data by generating new data points from available data is known as data augmentation. This includes making minor changes to data or by using machine learning models to develop new data points in the original data's latent space to amplify the dataset. This improves the performance and outcomes of deep learning models by generating fresh and varied training dataset instances.

3.1.7 GOOGLE COALBORATORY

Google Colaboratory software is used for the training portion in our study. The model's training was carried out on a personal laptop model - HP. Google Colaboratory, or "Google Colab" for short, is a Google Research product. It is a notebook (similar to a Jupyter Notebook) in our Google Drive where Python code can be run. We can write text, and code, run that code, and see the results all in the same notebook.

3.2 Method

For proper implementation and to get good results certain steps need to be followed which are discussed below.

Step 1: Import all the libraries and modules required.

Various libraries along with their work are discussed. Image processing tools are used for a variety of machine learning and deep learning tasks. Python, on the other hand, is a very popular and trending programming language with a large number of Python libraries. As a result, it is important to use Python libraries for image processing in order to complete all machine and deep learning tasks.

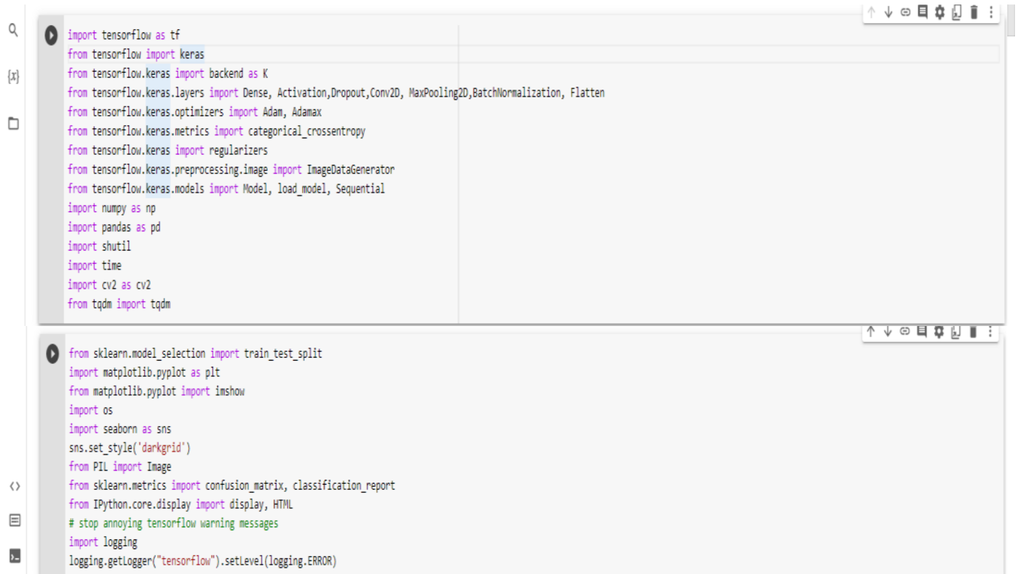


Figure 3.12 & 3.13: Shows all the library imported

The optimizer tunes the model during training to minimize the loss of training examples. After at least one epoch has been run, the loss and metrics plot in the Evaluation view will display the average value of the loss across all training and validation examples.

Step 2: Import the dataset.

We need to import datasets for data analysis while running Python codes. Python has several numbers of modules that assist us in importing external data in various file formats into a Python program.

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

Figure 3.14: Code for mounting drive

Step 3: Split the dataset into training, validation, and testing.

To avoid a model from over-learn from training data and then performing poorly in production. We must have a mechanism in place to determine how well the model generalizes. As a result, the input data must be divided into training, validation, and testing subsets to order to prevent overfitting and effectively evaluating our model.

Training Dataset is the part of the dataset used for learning by a model that is by fitting the parameters to the machine learning model.

The Validation Dataset is the dataset used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. Other types of model preparation, such as feature selection and threshold cut-off selection, are also involved.

Testing Dataset is the dataset used to provide an honest evaluation of a final model fitted on the training dataset.

```
[ ] def preprocess (csvpath, sdir, trsplit, vsplit):
        df=pd.read_csv(csvpath)
        df=df.copy()
        df=df.drop(['Unnamed: 0',  'Unnamed: 0.1'], axis=1)
        df.columns=['filepaths', 'labels']
        df['filepaths']= df['filepaths'].apply(lambda x: os.path.join(sdir,x))
        df['labels']=df['labels'].astype(str)
        # split df into train_df, test_df and valid_df
        strat=df['labels']
        dsplit=vsplit/(1-trsplit)
        train_df, dummy_df=train_test_split(df, train_size=trsplit, shuffle=True, random_state=32, stratify=strat)
        strat=dummy_df['labels']
        valid_df, test_df=train_test_split(dummy_df, train_size=dsplit, shuffle=True, random_state=32, stratify=strat)
        print('train_df length: ', len(train_df), '  test_df length: ',len(test_df), '  valid_df length: ', len(valid_df))
        print(train_df['labels'].value_counts())
        return train_df, test_df, valid_df
```

Figure 3.15: code for splitting dataset into training, testing, and validation set

Data Supplementation

First of all the data is cleaned and then split into training, validation, and testing. The total size of the input is 812.32 MB. The data is in the form of a CSV file which here is named train_data.csv and has three columns containing the image id, image name (file name), and the target value „0" and „1", where „0" denotes the absence of pneumothorax and „1" denotes the presence of pneumothorax in respective images. The chest x-ray images are present in a folder named small_train_data_set and a number of images 2027 are in png file format with the size 1024 X 1024. The training dataset consists of 1824 total images, 1437 images with the presence of pneumothorax, and 387 images with the absence of pneumothorax. The validation and testing dataset comprised 101 images in each set. Now as the training dataset for the images with the absence of pneumothorax is less in number so we make a trim and balance function in order to increase the number of images by augmentation technique and balancing training data.

Data augmentation is a technique that involves modifying existing training examples in order to create more diverse training examples from which a neural network can learn. Because images can be altered in a variety of ways to produce a wide variety of training examples for a network to learn from, data augmentation is a common technique used when training CNNs on image data.

To generate additional images from an existing set of training images, a variety of techniques can be used. These techniques include, but are not limited to:

- Horizontal and vertical image flips.
- Random image rescaling and cropping.
- Changing the image brightness and contrast.
- Modifying the hue, saturation, and value of the images.

When all of these augmentations are randomized and combined, data augmentation can occur batch by batch to produce a virtually infinite number of unique training images that differ slightly from the original images. Data augmentation enables the network to learn invariance to image properties that are unrelated to the expected output that it must produce. An image of an object, such as a car, remains an image of the car even if the brightness is changed or it is flipped horizontally. Instead of finding incorrect patterns and overfitting, a neural network could learn to recognize the features that truly define an image of a car through data augmentation.

After the data is split into training, testing, and validation datasets, the train_df is unbalanced as it has more images of one class than of another. In order to balance the dataset by first limiting the number of samples in a class, then creating and storing augmented images, generating a df of the augmented images, and merging it with train df to create a composite training set, df function is created.

```
[ ] max_samples= 600
    min_samples=0
    column='labels'
    working_dir = r'./'
    img_size=(500,500)
    ndf=balance(train_df,max_samples, min_samples, column, working_dir, img_size)

    [600, 387]
    Found 387 validated image filenames.
    Total Augmented images created= 213
    [600, 600]

[ ] channels=3
    batch_size=20
```

Figure 3.16: Code for data supplementation

Step 4: Visualize the dataset

```
csvpath=r'/content/drive/MyDrive/Colab Notebooks/5. MANJEET msc./train_data.csv'
sdir=r'/content/drive/MyDrive/Colab Notebooks/5. MANJEET msc./small_train_data_set/small_train_data_set'
train_df, test_df, valid_df= preprocess(csvpath,sdir, .9, .05)

train_df length:  1824   test_df length:  102   valid_df length:  101
1    1437
0     387
Name: labels, dtype: int64
```

Figure 3.17: Code for data visualization

```
img_path=r'/content/drive/MyDrive/Colab Notebooks/5. MANJEET msc./small_train_data_set/small_train_data_set/1.2.276.0.7230010.3.1.4.8323329.10013.1517875220.968509
img=plt.imread(img_path)
print (img.shape)
imshow(img)

(1024, 1024)
<matplotlib.image.AxesImage at 0x7fedb208de90>
```

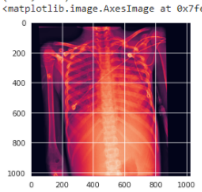


Figure 3.18: Data visualization

Step 5: Import the pre-trained model that is ResNet-50

We use multiple forward and backward iterations to find the best weights for the network. We can directly use the weights and architecture obtained by using pre-trained models that have previously been trained on large datasets and apply the learning to our problem statement by using pre-trained models that have previously been trained on large datasets. This is referred to as transfer learning. The pre-trained model's "learning" is "transferred" to our specific problem statement.

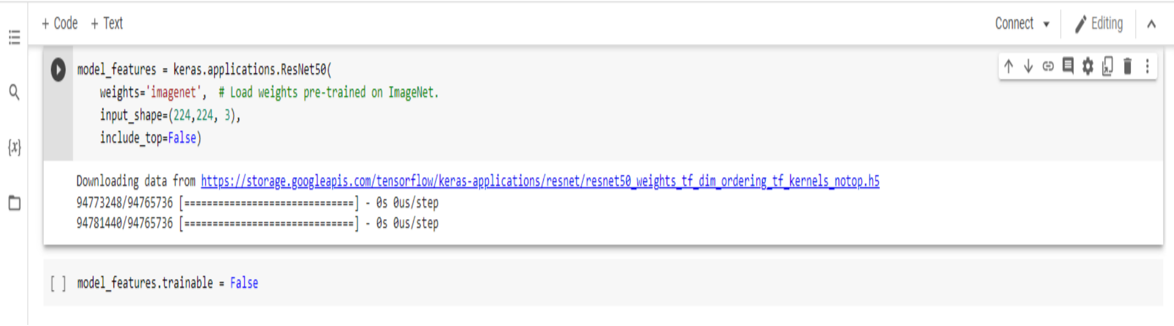


Figure 3.19: Code for importing ResNet-50

The summary of the ResNet-50 is as shown below

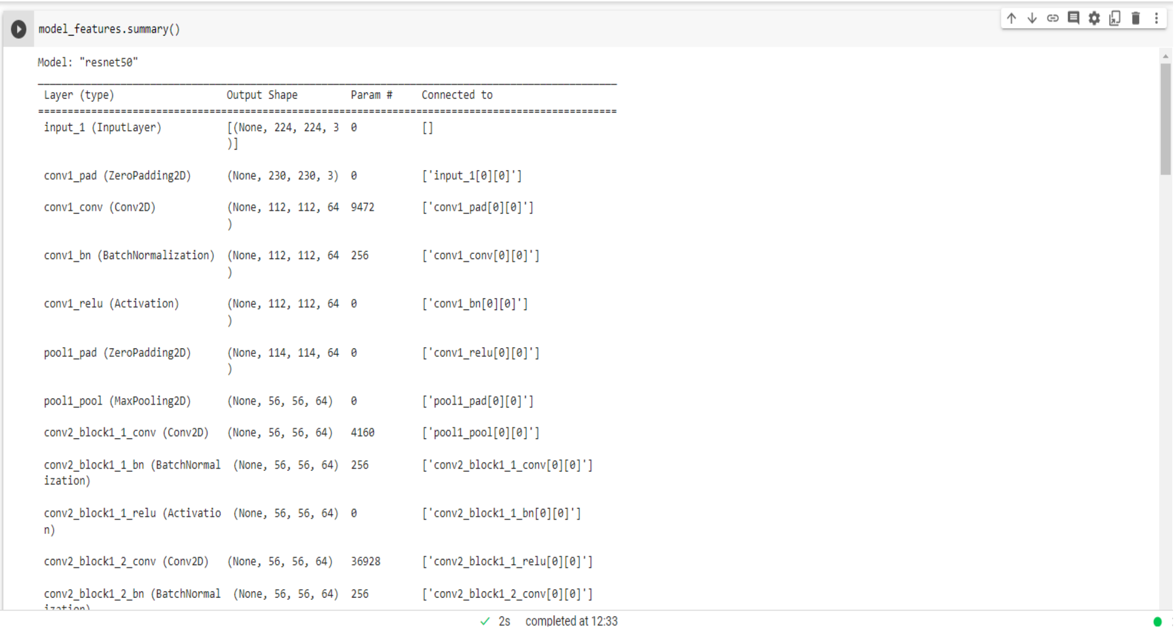


Figure 3.20: The summary of the ResNet-50 is as shown

Certain things to keep in mind while using ResNet-50 as a pretrained model in our proposed model are:

- We specify to include top=False when importing the ResNet50 class. This makes it possible for us to add input and output layers that are specific to our data.
- The weights= 'imagenet' is mentioned. This indicates that the weights the Resnet50 model learned while being trained on the imagenet data will be used.
- We conclude by mentioning the layer. In the pre-trained model, trainable=False. We will save a great deal of time and space complexity by preventing the model from having to learn the weights again.
- We will also build a fully connected output layer where actual learning may occur now that we have imported a pre-trained model.

Proposed model

The fully connected layer of pretrained model is removed and then global max pooling layer is added which is followed by dense layer with 512 neurons and the activation used here is relu. Further dropout technique is used with drop out value of 0.3 which means 30 percent of the preceding layer's activations will be set to zero at random. Finally the output layer is added which is actually SVM having two classes of classification. The optimizer Adam is used with learning rate of 0.001, loss function „hinge loss‟ is used as it better works with SVM , L2 regularization technique is used with activation function „Linear‟ as we have only two classes for classification.

```
model_features.trainable = False
```

SVM

```
inputs = keras.Input(shape=(224, 224, 3))
# We make sure that the base_model is running in inference mode here,
# by passing `training=False`. This is important for fine-tuning, as you will
# learn in a few paragraphs.
x = model_features(inputs, training=False)
x = keras.layers.GlobalAveragePooling2D()(x)
x = Dense(512,activation = 'relu')(x)
x = keras.layers.Dropout(0.3)(x)
x = Dense(128,activation = 'relu')(x)
outputs = keras.layers.Dense(2,kernel_regularizer=tf.keras.regularizers.l2(0.001), activation = 'linear')(x)
model = keras.Model(inputs, outputs)

model.compile(optimizer = tf.keras.optimizers.Adam(learning_rate=0.001), loss='hinge', metrics = ['accuracy'])
```

Figure 3.21: The summary of the ResNet-50 is as shown.

We have aggregated the unique and computational attributes of Deep Learning and Machine Learning. We have used the Deep Transfer Learning technique which is a pre-trained model that is a Residual Network for image feature extraction and the Support Vector Machine (SVM) algorithm which creates the best decision boundary known as "Hyperplane" to classify data points, is a Machine Learning model used for pneumothorax binary classification.

For image classification, a pre-trained Imagenet model is used. We use include top = False means in the arguments, in order to remove the fully connected layers in the end.

Summary of tools

Dropout is a technique that involves randomly turning off the activations of specific neurons in a layer to prevent the network from overly relying on specific features when making predictions. Dropout is synonymous with "dropping out" or removing neurons at random. A parameter in a dropout layer will indicate the percentage of neuron activations that should be set to zero. A dropout value of 0.3, is used in our model which indicates that 30% of the preceding layer's activations will be set to zero at random. The activation Rectified Linear Unit (ReLU) function with one of the few benchmarks in the deep learning revolution is the rectified linear activation unit, or ReLU. It's straightforward but far superior to various other activation functions such as sigmoid or tanh.

The ReLU formula is as follows: $f(x) = \max(0,x)$

The ReLU function and its derivative are both monotonic functions. If the function is given a negative value, it returns 0; however, if it is given a positive value, it returns that value. As a result, the output ranges from 0 to infinity. For implementation of SVM Linear Activation Function comes into action here. The linear activation function (multiplied x1.0), also known as "no activation" or "identity function," is one in which the activation is proportional to the input. The function makes no changes to the weighted sum of the input; it simply returns the value it was given. This function is used in last layer for SVM model. Learning Rate plays an important role in weight updation. The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward the minimum of a loss function in machine learning and statistics. In our proposed model learning rate ss taken to be 0.001. The Hinge loss is a loss function in machine learning that is used to train classifiers. The hinge loss is commonly used in "maximum-margin" classification, most notably in support vector machines (SVMs). The hinge loss of the prediction y is defined for an intended output t = 1 and a classifier score y.

$$y = \max(0, 1 - t \cdot y)$$

y should be the "raw" output of the decision function of the classifier, not the predicted class label. For example, in linear SVMs, $y = w.x + b$, where (w, b) are hyperplane parameters and x is the input variable (s). When t and y have the same sign (meaning y predicts the correct class) and $|y| >= 1$, the hinge loss l(y) is equal to zero. When they have opposite signs, l(y) increases linearly with y, and if $|y| < 1$ has the same sign, l(y) increases linearly with y. (correct prediction, but not by enough margin).

Step 6: Model Evaluation

There are numerous performance metrics that can be used to assess classification models. AUC is preferred over other performance measures in the case of class imbalance datasets. Furthermore, most authors in the literature have reported the superiority of their proposed methods in terms of AUC. Few researchers, however, have calculated other performance metrics such as accuracy, sensitivity, precision, and specificity. The formulas for the calculation of accuracy, recall/ sensitivity, specificity, and precision are as written below.

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

$$\text{Recall or Sensetivity} = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F1 score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$\text{Sorensen Dice Coefficient} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}$$

Chapter-4 RESULTS AND DISCUSSION

Pneumothorax is an abnormal condition in which air builds up in the pleural space of the thorax, which is located between the chest and the lungs and when this air is forcefully thrown outside leads to collapse of the lungs either partially or fully. As a result, this should be diagnosed early so that treatment can begin as soon as possible. Neural networks are effective in image classification tasks, and many researchers have demonstrated that chest anomalies such as pneumonia, pneumothorax, and others, which are diagnosed by radiologists, can be detected by training a neural network model. In this chapter the results of, proposed model that is ResNet-50 a deep neural network along with machine learning model that is SVM, that was trained on a dataset of over 2,027 chest x-ray images to predicts the presence or absence of pneumothorax from chest x-ray images are discussed.

After applying ResNet-50 to predict the presence or absence of pneumothorax the success rate of the method used that gives the results of the study is shown in this session.

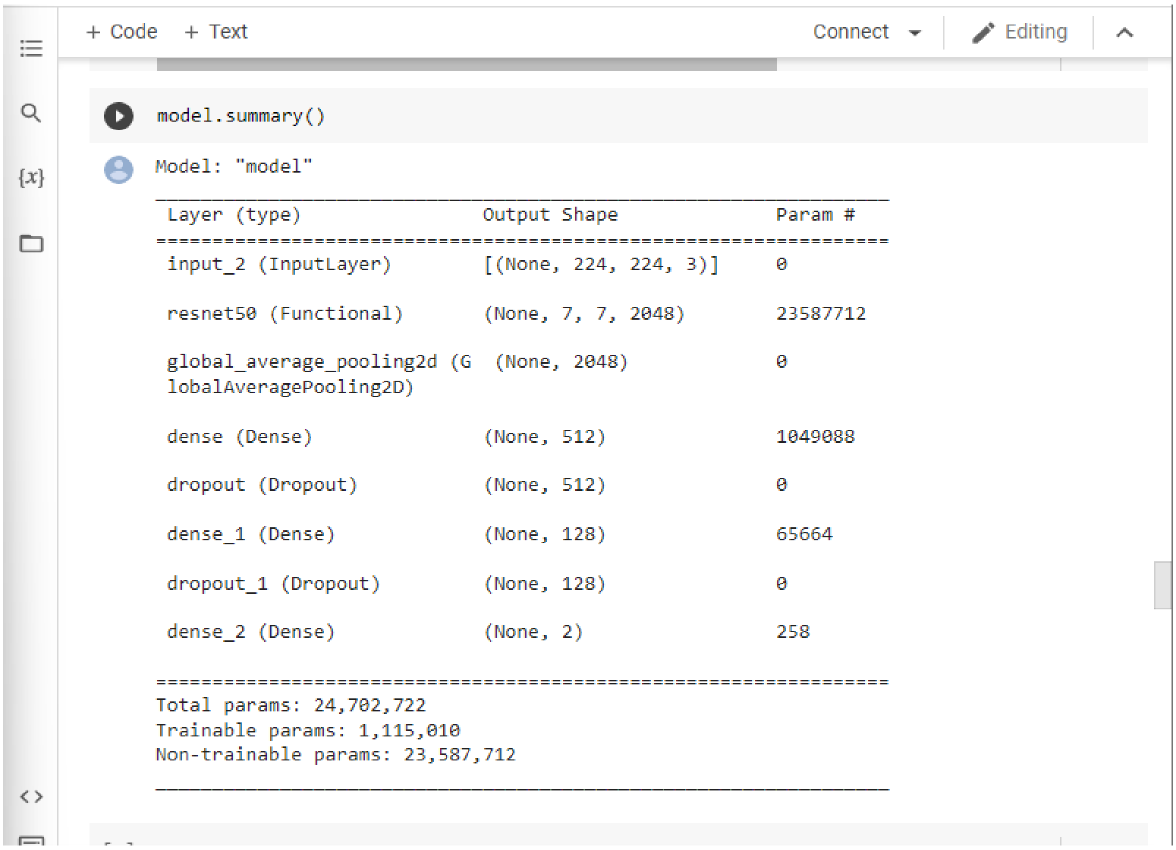


Figure 4.1: Summary of proposed model

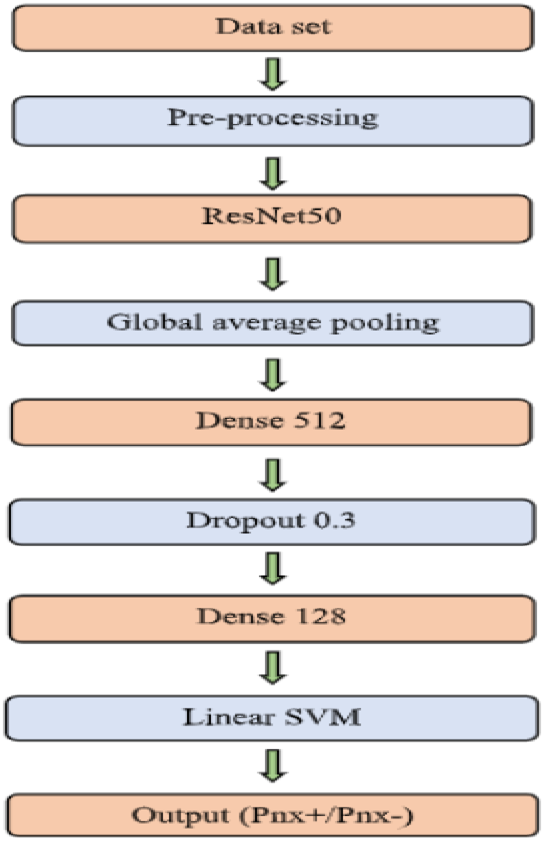


Figure 4.2: Flowchart of proposed model

The summary of the proposed model as well as flow chat is shown in the above figure. We initially have an image of size (1024,1024) but as we are using a pre-trained model ResNet-50, it accepts input images of a certain size. So, we resize our image to (224, 224), and is given as input to pre-trained model that is ResNet-50 which is followed by global average pooling which further is followed by three dense and one dropout layers.

Our model combines Keras callbacks like- Reduce Learning Rate on Plateau, Early Stopping, and Model Checkpoint for training and validating data. Furthermore, it provides a more readable summary of the model's performance after the conclusion of each epoch. It also has a useful feature that allows us to specify the number of epochs to train for until a message is received if we want to stop training on the current epoch, enter an integer indicating how many more epochs should be run before the message reappears This is extremely useful while training a model and determines whether the metrics are satisfactory and wherever we want to end. The callback always returns to the model with the weights set to those of the highest performing epoch depending on the metric being tracked (AUC or validation AUC). The callback will initially monitor the training AUC and adjust the learning rate until the AUC reaches a user-specified threshold level. When that level of training AUC is reached, the callback switches are activated to monitor validation loss and adjust the learning rate accordingly.

The callback takes the following form:

callbacks = [tensorboard, checkpoint, reduce_lr], epochs = 40)

```
Epoch 38/40
60/60 [==============================] - ETA: 0s - loss: 0.3527 - auc: 0.9238
Epoch 38: val_loss did not improve from 0.38934

Epoch 38: ReduceLROnPlateau reducing learning rate to 5.904900035602622e-09.
60/60 [==============================] - 25s 408ms/step - loss: 0.3527 - auc: 0.9238 - val_loss: 0.4375 - val_auc: 0.8831 - lr: 1.9683e-08
Epoch 39/40
60/60 [==============================] - ETA: 0s - loss: 0.3395 - auc: 0.9296
Epoch 39: val_loss did not improve from 0.38934
60/60 [==============================] - 25s 407ms/step - loss: 0.3395 - auc: 0.9296 - val_loss: 0.4375 - val_auc: 0.8831 - lr: 5.9049e-09
Epoch 40/40
60/60 [==============================] - ETA: 0s - loss: 0.3418 - auc: 0.9291
Epoch 40: val_loss did not improve from 0.38934
60/60 [==============================] - 25s 407ms/step - loss: 0.3418 - auc: 0.9291 - val_loss: 0.4375 - val_auc: 0.8831 - lr: 5.9049e-09
```

Figure 4.3: Loss and AUC obtained by our model

The batch size of training data is taken to be 20 so for 1200 images in 60 batches the total data will be trained for each epoch. Epochs is an integer value that represents the number of epochs to train and the number of epochs is set to be 40. Patience is an integer that specifies the number of consecutive epochs that can occur before the learning rate is adjusted. The float factor determines the new learning rate via the equation lr=lr*factor. The random state is set to 32 and the data is shuffled for both training and validation datasets. Stop patience is an integer that specifies how many successive epochs the learning rate was adjusted for but no improvement there in monitored metric occurred before training was terminated for which Early Stopping callback is used.

As result, for 40 epochs the training set obtains an AUC of 92.91 and training loss as .3418 as well as the validation AUC is obtained to be 88.31 and validation loss is .4371 for learning rate 5.9049e^09. After training, the model was tested on test images, and the results were captured in an array, with an average accuracy of 79.261.

Confusion Matrix

A confusion matrix is a table used in classification problems to determine where errors in the model did occur. Where the rows represent the actual classes for which the outcomes should have been calculated. Whereas the columns represent our predictions. This table makes it simple to see which predictions are incorrect.

Here TN, TP, FN, and FP is True Negative, True Positive, False Negative, and False
The Confusion Matrix created has four different quadrants:
True Negative (Top-Left Quadrant)
False Negative (Top-Right Quadrant)
False Positive (Bottom-Left Quadrant)
True Positive (Bottom-Right Quadrant)

Table 4.1- Confusion matrix

Predicted Class		
Actual Class	Negative (normal)	Positive (pathology)
Negative (normal)	TN	FN
Positive (pathology)	FP	TP

For the proposed model, the required confusion matrix is shown below with the prediction and actual values. True indicates that the values were correctly predicted, while False indicates that there was an error or incorrect prediction,

```
[ ]  confusion_matrix(y_true_cat, y_pred_cat)

     array([[17,  5],
            [15, 65]])
```

Figure 4.4: Confusion matrix obtained

Table 4.2- Confusion matrix in tabular form

N = 102	Negative (normal)	Positive (pathology)
Negative (normal)	17	5
Positive (pathology)	15	65

The actual results of the images from the excel sheet were in the form of 0 and 1, where 0 indicates that the x-ray does not have pneumothorax and 1 indicates that the image has pneumothorax. After training we get results and finally the confusion matrix is then formed using the sklearn library's confusion matrix, and the values from the confusion matrix are used to calculate Sensitivity and Specificity.

There are numerous performance metrics that can be used to assess classification models. AUC is preferred over other performance measures in the case of class imbalance datasets. Furthermore, most authors in the literature have reported the superiority of their proposed methods in terms of AUC. Few researchers, however, have calculated other performance metrics such as accuracy, sensitivity, precision, and specificity. Performance is calculated manually and the results obtained are shown below in the table.

Table 4.3: All the performance metrics along with formula and value obtained are shown

Metrics	Formulae	Performance
Accuracy	$\frac{TN + TP}{TN + TP + FN + FP}$	0 .833
Recall or Sensitivity	$\frac{TP}{FN + TP}$	0.812
Specificity	$\frac{TN}{TN + FP}$	0.773
Precision	$\frac{TP}{TP + FP}$	0.929
F1-score	$\frac{2 * Recall * Precision}{Recall + Precision}$	0.867
Dice coefficient	$\frac{2 * TP}{2 * TP + FP + FN}$	0.866

The confusion matrix as well as classification report with all the metric performance computed by model are shown below:

Classification Report

precision	recall	f1-score	support	
0	0.531	0.773	0.630	22
1	0.929	0.812	0.867	80
accuracy			0.804	102
macro avg	0.730	0.793	0.748	102
weighted avg	0.843	0.804	0.816	102

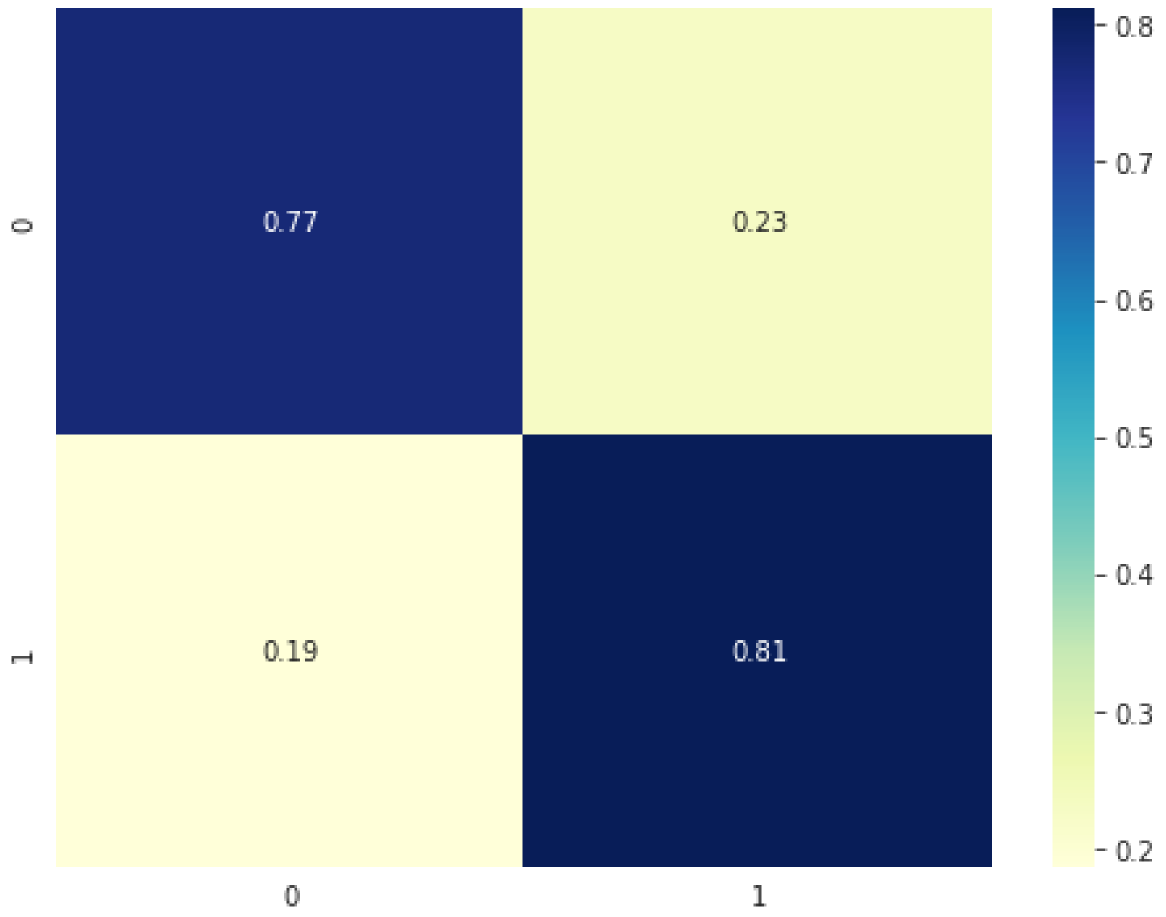


Figure 4.5: Confusion matrix

Model loss and AUC

In this section, we show graphs of the training and validation loss values, as well as the validation auc and training auc for ResNet-50, for our proposed model. The small size of the validation set in comparison to the training set explains the volatility of the loss curves. If performance improvements on the validation data occurred after each epoch, the best weight configuration for each network was saved to storage media. We measured validation data performance using the auc value, which is the area under the curve, rather than the validation loss. As a result, we also included graphs displaying the auc values of both the training and validation during training.

This graph demonstrates the loss value against epoch. The term 'loss' in the figures denotes loss during training and validation, while the orange colour curve represents validation loss and the blue colour curve represents training loss

Figure 4.6: Loss graph

This graph demonstrates the AUC value against epoch. The term 'AUC' in the figures denotes area under the curve during training and validation, while the orange colour curve represents validation AUC value and the blue colour curve represents training AUC value.

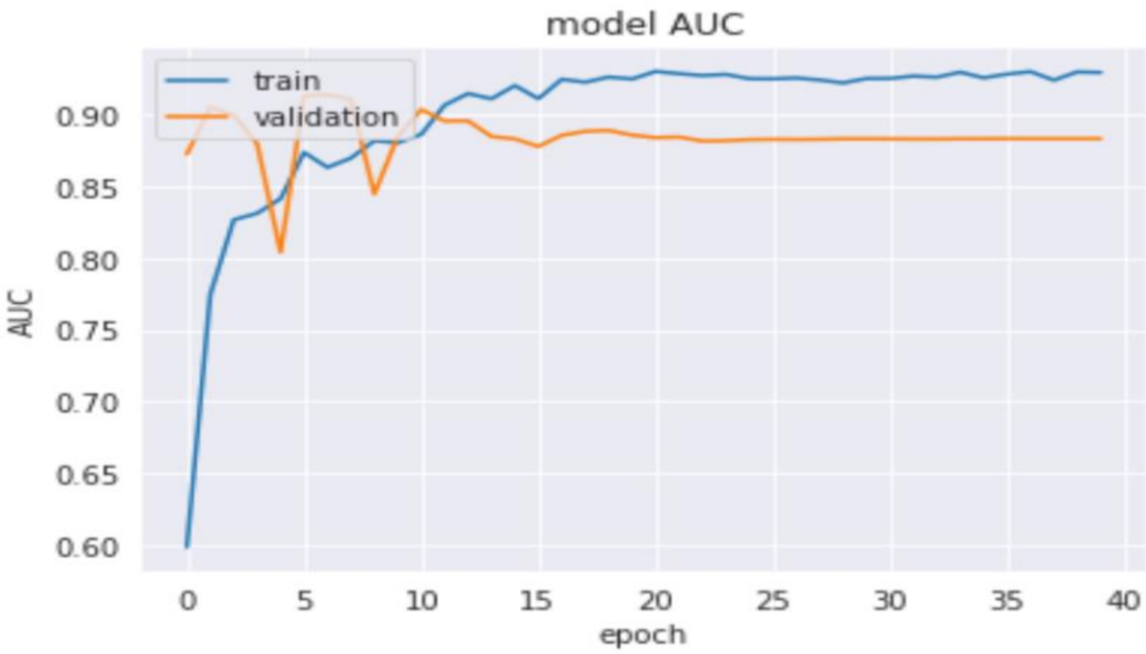


Figure 4.7: AUC graph

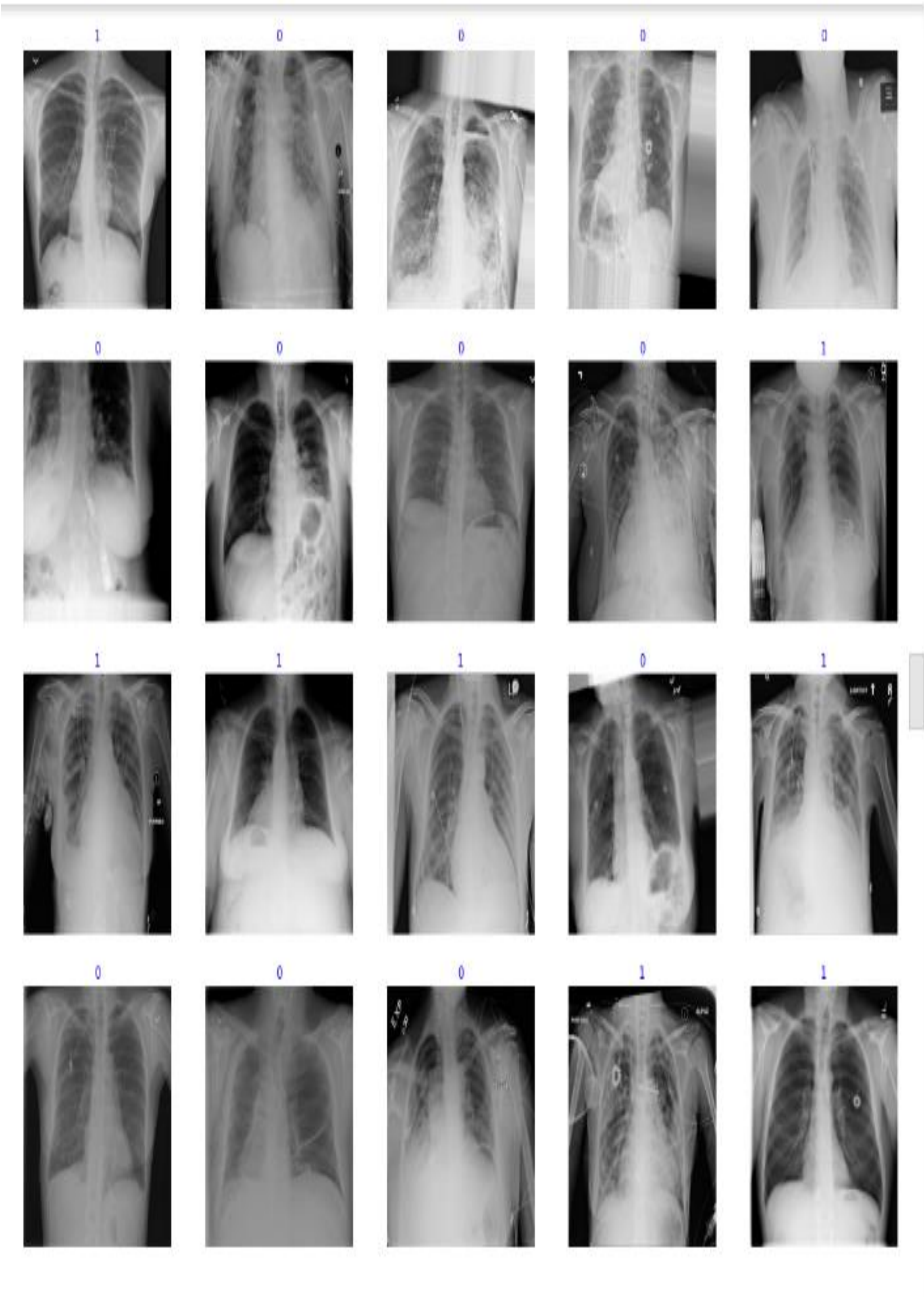


Figure 4.8: The testing images correctly predicted by our model are shown above

Table 4.4: Comparison of proposed model with other models

Author	Year	Tech	Methodology Description	Class Imbalance Solution	Result (%)
Wang *et al*.	2017	DL	NIH Dataset Collected, DCNNs were applied and bounding box were generated for localization	Proposed Weighted Cross Entropy Loss Function	Classification: AUC=79.93
Li *et al*.	2018	DL	Multi-Resolution and Multi Instance Learning	No Proper Explanation for class Imbalance	Classification: AUC=80.5
Baltruschat *et al*.	2019	DL	Transfer Learning with and without fine tuning was explored Using ResNet-50	Class-Averaged binary Cross Entropy	Random Split: AUC=87.0 Official Split: AUC=84.0
Sze *et al*.	2019	DL	Transfer Learning Using CheXNet	Weighted Binary Cross Entropy Function	AUC=70.8
Tang *et al*.	2019	DL	Generative Adversarial Network to distinguish between normal and abnormal CXRs	Balanced Dataset	AUC=84.1
Bharati *et al*.	2020	DL	VGG16, data Augmentation, STN	No Proper Explanation for class Imbalance	Accuracy=70.8 Val. Accuray = 73
Choudhary and Hazra	2021	DL	Own CNN architecture and transfer learning Using VGG16	No proper Explanation for Class Imbalance.	Accuracy=83.7
Proposed Work	**2022**	**DL and ML**	**ResNet-50 Along with Support Vector Machine**	**Dataset is balanced by augmentation.**	**AUC=88.31 Accuracy = 83.3 Recall = 0.812 Specificity = 0.773 Precision = 0.929 F1-score = 0.867 Dice coefficient = 0.866**

5. Discussion

Drawing inspiration from previously successful implementations, this study looks into ML and DL implementations and thus an integrated models of these two combines their benefits and provides a fresh look at the medical field. In this study, a ResNet-50 is used for the deep feature extraction and SVM as a classifier is used to build an integrated model. In the medical field a single classification AUC value is insufficient so that a complete picture of model's performance can be visualized. The following parameters were also assessed: accuracy, sensitivity, precision specificity, and f1 score. This integrated model is similar to an automated system that extracts and classifies deep features from chest X-ray images in a single click and detects the presence or absence of pneumothorax. Furthermore, while DL and ML techniques are ideal for classification tasks with large datasets, they can also be used in fields with limited data. Although the dataset taken was small for deep feature extraction and classification, but various effective techniques such as data augmentation, drop-out, callbacks, global average pooling, and optimizer improved detection capability.

In previous research, Wang al et., the data was balanced by weight cross-entropy loss function but still accuracy for feature extraction was so low. Yao Li al et., in this study no proper explanation for class imbalance was provided and in addition to this comparison with related works is missing. Baltruschat, the class imbalance issue was solved by applying averaged binary cross entropy but on the random split, training was performed, performance needs some improvement in this case. Sze, the dataset was balanced but it was extremely small in addition to which performance also needs improvement. Tang the dataset taken for this research was balanced, but the criteria for selection was not mentioned also a specific pathology should be main focus. Bharati, the reseach work was conducted without providing any proper explanation for class imbalance. The comparison with related works was unfair also the study lacks providing any experimental detail about data splitting, and performance in other metrics for example in terms of AUC was missing. Choudhary and Hazra, no proper explanation for class imbalance as well as data split were given and in addition to this performance in terms of other metrics was missing.

Because DL methods can make intelligent decisions without human intervention, they are preferred over ML in the existing literature for pneumothorax detection. Because the size of the training data affects the performance of a deep learning model. All of the CXRs datasets available to date have very few pneumothorax samples compared to the total dataset size, so researchers must contribute to collecting larger datasets with a greater number of pneumothorax samples. It is well known that most medical image datasets are imbalanced in nature, containing fewer pathology samples than normal samples. This could lead to biassed results in favour of majority class samples. Most researchers have attempted to solve this problem using algorithm level approaches, such as adding weights to the loss function to give more priority to fewer occurring samples, as shown in related works. It is undeniable that the trend has shifted from Machine Learning (ML) to Deep Learning techniques over the last decade (DL). For classification and localization, the proposed frameworks mostly used pre-trained Convolutional Neural Networks (e.g., VGG16, ResNet, and DenseNet) . As a result, there are lot of opportunities to work in this field. But even so, it is observed that most research teams have used DenseNet or ResNet for classification purposes, so it could be an interesting thought experiment to try some other and new CNN architectures like EfficientNet in automating the field of automatic detection of thoracic pathologies, particularly pneumothorax.

There are numerous methods for extracting features and machine learning techniques available. Pretrained models are what, deep feature extractors. Building an optimal model for a specific task from scratch takes a significant amount of time and effort, but transfer learning provides an efficient way to accomplish this goal. It applies the pre-trained model's learning in terms of weights to our specific problem, resulting in higher accuracy and less training time. Deep learning, extraction of features practices and machine learning classification methods are a good fit for this task. Since machine learning and deep learning methods individually resulted in an impressive performance, the current study investigated whether the performance of previous DL/ML classifiers could be improved by aggregating them.

At last, we can conclude that our research outperforms previous works in the field of pneumothorax because by using publicly available datasets on kaggle we allow other researchers to study, comprehend, and provide feedback. Our results show that the proposed framework ResNet-50, a feature extractor, and Support Vector Machine, a classifier that is a fusion of deep transfer learning and machine learning, outperforms the existing approaches and can be applied to any pneumothorax dataset.

Chapter-5 SUMMARY AND CONCLUSION

With a summary, this chapter determines the opinion. A novel model has been proposed for pneumothorax detection using deep transfer learning and machine learning. Generally, the clinical diagnosis of pneumothorax is based on both images as well as other information sources. Chest pain and shortness of breath are symptoms of primary spontaneous pneumothorax, chronic lung diseases are responsible for secondary pneumothorax while traumatic pneumothorax is caused due to any injury, a complication after penetrative interventions like lung tissue biopsy or stabbing. When X-ray information is insufficient for a confident diagnosis, medical image analysis is frequently the deciding factor. Here, the procedure is planned for pneumothorax detection by aggregating deep and machine learning.

Chapter 1- The ‘Introduction’ is purely introductory and discusses all of the fundamental aspects of this study. The chapter consists of all the definitions and basic concepts of ANN, CNN, SVM, back propagation, and all other terms used for the detection and binary classification of pneumothorax.

Chapter 2- The ‘Review of Literature’ is the study of the literature of all the latest research papers associated with the use of Deep Learning and Machine Learning on Chest X-ray images for pneumothorax detection. This chapter of the thesis also reflects on how, over time, research in this field has grown and continues to be a hot field of research in the present and future.

Chapter 3- With the help of python codes on google collaboratory a model is developed using deep transfer learning that is a pre-trained model named as ResNet-50 aggregated with machine learning SVM which is a classifier to set a new record with a good performance by achieving state-of-the-art results. Back propagation along with data balancing is used to train the model.

Chapter 4- The results obtained by the method proposed in the chapter- 3 are discussed in detail. The model developed is good enough as it performs on the testing dataset is very good that is the predicted outputs resemble the actual outputs. It can be concluded that a combination of DL and ML models are promising models in medical image analysis.

CONCLUSION

In this research, a novel model which is obtained by the integration of DL for deep feature extraction and ML for classification for the detection and binary classification of pneumothorax on Chest X-ray images and its evaluation. A pretrained model, ResNet-50 with 50 layers along with SVM is trained on a publically available dataset on Kaggle which improves the performance for binary classification. We worked hard to solve several problems that are still a bit of a puzzle to the research community. To improve results back propagation along with adam optimizer, callbacks like 'ReduceLROnPlateaue', 'Early Stopping', and 'checkpoint' are used. In order to avoid overfitting techniques like regularization l2, augmentation and dropout layer with 0.3 is used. In terms of metrics such as accuracy, recall (Sensitivity), f-1 score, precision, loss, and AUC, our proposed method model outperforms (Area Under the Curve). This study set a new high standard for performance, achieving state-of-the-art results of 0.8831 in terms of AUC and 0.4375 in terms of loss. By obtaining these performances we conclude that DL and ML are powerful tools as they give results of high accuracy to medical images. Comparison with various other models is done and our model performs best among all of them. It's simple to understand and easy to implement over dataset.

Aozasa, K., Ohsawa, M., Iuchi, K., Tajima, K., Komatsu, H., and Shimoyama, M. 1993. Artificial pneumothorax as a risk factor for development of pleural lymphoma. *Jpn J Cancer Res.*, 84(1), 55-57.

Baltruschat, I. M., Nickisch, H., Grass, M., Knopp, T. and Saalbach, A. 2019. Comparison of Deep Learning Approaches for Multi-Lebel Chest X-Ray Classification. *Sci. Rep.*, 9(1): 1-10.

Bharati, S., Podder, P. and Mondal, M.R.H. 2021. Hybrid deep learning for detecting lung dieases from X-ray images. *Inform. Med. Unlocked*, 20: 100391.

Blumenfeld, A., Konen, E. and Greenspan, H. 2018. Pneumothorax detection in chest radiographs using convolutional neural networks. *SPIE Med. Imag.,*:10575(10): 110-119.

Cha, S.M., Lee, S.S. and Ko, B. 2021. Attention-Based transfer learning for efficient pneumothorax detection in chest X-ray images. *Appl. Sci.*, 11(3): 1-15.

Chan, Y.H., Zeng, Y.Z., Wu, H.C., Wu, M.C. and Sun, H.M. 2018. Effective pneumothorax detection for chest X-ray images using local binary pattern and support vector machine. *J. Healthc. Eng.*, 2018: 1-11.

Choudhary, P., and Hazra, A. 2021. Chest disease radiography in twofold: using convolutional neural networks and transfer learning. *Evol. Syst.*, *12*(2), 567-579.

Cho, Y., Kim, J.S., Lim, T,H., Lee, I. and Choi, J. 2021. Detection of the location of pneumothorax in chest X-rays using small artificial neural networks and a simple traing process. *Sci. Rep.*,11(1): 1-8.

Folke, T., Yang, S.C., Anderson, S. and Shafto, P. 2021. Explainable AI for medical imaging: Explaining pneumothorax diagnoses with Bayesian Teaching, *arXiv.,:* 2106.04684.

Geva, O., Zimmerman-Moreno, G., Lieberman, S., Konen, E., and Greenspan, H. 2015. Pneumothorax detection in chest radiographs using local and global texture signatures. *SPIE Med. Imag.,* 9414: 448-454.

GooBen, A., Deshpande, H., Harder, T., Schwab, E., Baltruschat, I., Mabotuwana, T., Cross, N. and Saalbach, A. 2019. Chexpert: Deep Learning For Pneumothorax Detection and Localization in Chest Radiographs. *arXiv*: 1901.07324.

He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep residual learning for image recognition. *Proc. IEEE Int. Conf. Comput. Vis.*, 17(2): 770-778.

Hooda, R., Mittal, A. and Sofat, S. 2019. Segmentation of lung fields from chest radiographs-a radiomic feature-based approach. *Biomed. Eng. Lett.*, 9(1): 109-117.

Iqbal, T., Shaukat, A., Akram,M. U., Mustansar, Z. and Khan, A. 2021. Automatic diagnosis of pneumothorax from chest radiographs: a systematic literature review. *IEEE Access*: 1-23.

Irvin, J., Rajkumar, P., Ko, M., Yu, Y., Ilcus, S.C., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D.A., Halabi, S.S., Sandberg, J.K., Jones, R., Larson, D.B., Langlotz, C.P., Patel, B.N., Lungren, M. P. and Ng, A.Y. 2019. Chexpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparrison. *Proc. Conf. AAAI Artif. Intell.*, 33(1): 590-597.

Luo, J.X., Liu, W.F. and Liang, Y. 2022. Pneumothorax Recognition Neural Network Based on Feature Fusion of Frontal and Lateral Chest X-Ray Images *IEEE Access.*, 10: 53175-53187.

Malhotra, P., Gupta, S., Koundal, D., Zaguia, A., Kaur, M. and Lee, H.N. 2022. Deep Learning -Based Computer-Aided Pneumothorax Detection Using Chest X-ray Images. *Sensors.*, 22(6): 2278.

Melendez, G. P. and Cordel, M. 2015. Texture-based detection of lung pathology in chest radiographs using local binary patterns. *Int Conf Bioinform Biomed.*, 1109(10): 1-5.

Shreyas, M. S., Bhat, M. A. R. and Rao, V. S. 2021. Use Neural Networks to Detect Pneumothorax on X-ray Images. *IEEE INOCON.*, 1109(10): 1-6.

Sze-To, A. and Tizhoosh, H. 2020. Searching for Pneumothorax in Half a Million Chest X-Ray Images *Sci. Rep.*, 11(1): 1-13.

Sze-To, A., Riasatian, A. and Tizhoosh, H.R. 2020. Searching for pneumothorax in x-ray images using autoencoded deep features. *Artif. Intell. Med.*, 12299: 453-462.

Tang, Y. X., Tang, Y. B., Han, M., Xiao, J., and Summers, R. M. 2019. Abnormal chest x-ray identification with generative adversarial one-class classifier. *IEEE Trans. Cogn Int J Biomed Imaging.,* 18(3): 1358-1361.

Taylor, A,G., Mielke,C. and Mongan, T. 2018. Automated detection of moderate and large pneumothorax on Frontal chest X-rays using deep convolutional neural network: A retrospective study. *PloSMed.*, 15(11): e1002697

Tolkachev, A., Sirazitdinov, I., Kholiavchenko, M., Mustafaev, T. and Ibragimov, B. 2021. ChestX-ray8: "Deep Learning for Diagnosis and Segmentation of Pneumothorax: The Results on The Kaggle Competitionand Validation Against Radiologists. *In IEEE J. Biomed. Health Inform.,* 25(5):1660-1672.

Wang, Q., Liu, Q., Liu, G., Liu, Z., Huang, J., Zhou,Y., Xu, W. and Cheng, J.Z. 2020. Automated segmentation and diagnosis of pneumothorax on chest X-rays with fully convolutional multi-scaleScSE-DenseNet: a retrospective study. *BMC Med. Inform. Decis. Mak.,* 20(14): 1-12.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M. and Summers, R.M. 2017. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Dieases. *In Prec. IEEE Comp. Soc. Conf. C.,*: 2097-2106.

Wang, X., Yang, S., Lan, J., Fang, Y., He, J., Wang, M., Zhang, J. and Han, X. 2020. Automatic Segmentation of Pneumothorax in Chest Radiographs Based on Two-Stage Deep Learning Method. *IEEE Trans. Cogn Dov. Syst.,* 14(1):205-218.

Wang, Y., Sun, L. and Jin, Q. 2021. Enhanced Diagnosis of Pneumothorax with an Improved Real-Time Augmentation for Imbalanced Chest X-rays Data Based on DCNN. *IEEE/ACM Trans. Comput. Biol. Bioinform.,* 18(3): 951-962

Yao, L., Prosky, J., Poblenz, E., Covington, B. and Lyman, K. 2018. Weakly supervised medical diagnosis and localization from multiple resolutions. *arXiv.,*: 1803.07703.

Yu, J. T., Lin, Y. R., Lai, C. C., Chang, Y. K., Huang, T. K., Lee, C. C. and Weng, S. J., 2021. Use Neural Networks to Detect Pneumothorax on X-ray Images. *JAMA Intern. Med.* 15(S): 39-46.

CURRICULUM VITAE

Name	: Manjeet Kaur	**Phone Number**	: 7088363949
Mailing Address	: Village- Bahadurpur, P.O.- Jaspur, District- Udham Singh Nagar, State- Uttrakhand, 244712	**Permanent Address**	: Village- Bahadurpur, P.O.- Jaspur, District- Udham Singh Nagar, State- Uttrakhand, 244712
E-mail	:mk4319174@gmail.com		

Career Objective : **To work in a progressive organization where I can improve my skills and talents and to utilize all my potential to its fullest for the progress of the organization**

Educational Qualification

S. No.	Examination Passed	Institution	Year	Percentage/ CGPA
1.	M.Sc.	G.B.P.U.A.&T, Pantnagar	2022	7.06
2.	B.Sc	Indira Priyadashani Govt. Girls College Of Commerce, Haldwani Kumaun University, (Nainital)	2020	6.44
3.	Intermediate	Maria School, Jaspur	2017	7.85
4.	High school	Maria School, Jaspur	2015	8.45

- **Specialization:** Major: Mathematics
- **Thesis Title:** Binary Classification of Pneumothorax in Chest X-Ray Images Using Deep Neural Network
- **Software Skills :** MS Office, Java, Python programming
- **Professional Skills:**

Place: Uttrakhand

(Manjeet Kaur)

Date:

Name	:	Manjeet Kaur	**Id. No.**	:	56669
Semester and year of admission	:	1st Semester, 2020-21	**Degree**	:	M.Sc.
			Department	:	Mathematics, Statistics and Computer Science
Major	:	Mathematics	**Minor**		Computer Science
Thesis Title	:	**"Binary Classification of Pneumothorax In Chest X-Ray Images Using Deep Neural Network"**			
Page No.	:	67	**Advisor**		Dr. Arun Kumar

ABSTRACT

Currently, lung diseases are extremely common throughout the globe and few of which include chronic obstructive pulmonary disease, pneumonia, pneumothorax, tuberculosis etc. Air in the pleural cavity is thought to be as "pneumothorax". This is a state when the lung surface or chest wall is breached, allowing air to enter the pleural space and causing the lung to collapse, it is a serious situation and might be life-threatening. Chest radiographs are the foremost significant and extensively used diagnostic techniques to detect thoracic or lung diseases among various medical imaging technologies. Pneumothorax can be diagnosed by chest X-rays, CT scans or ultrasound techniques but chest X-ray is the most often used accessible radiological method for screening and diagnosing thoracic diseases. The diagnosis from the chest X-Ray can be of great efficiency and time saving if done through automated systems instead of manual image reading. Tremendous research works are being conducted to develop reliable automatic diagnostic systems for detecting diseases within the chest radiographs. Artificial Intelligence (AI) tools have proven to be effective in optimizing the medical industry. Several framework-supported computing techniques have been proposed for the automated identification of pneumothorax from chest radiographs. Numerous models for pneumothorax automated diagnosis are certainly available. In this study, we have taken chest X-ray images and then used the Deep Transfer Learning technique along with Machine Learning to detect the presence or absence of pneumothorax in chest X-ray images. We have aggregated the unique and computational attributes of Deep Learning and Machine Learning. We have used the Deep Transfer Learning technique which is a pre-trained model that is a Residual Network for image feature extraction and the Support Vector Machine (SVM) algorithm which creates the best decision boundary known as "Hyperplane" to classify data points, is a Machine Learning model used for pneumothorax binary classification. To achieve effective outcomes, a balancing data technique with augmentation is used to create a balance between the training and validation dataset, as well as an automatic adjusting learning rate technique called "ReduceLROnPlateau" to monitor validation loss and obtain optimal learning rates. As an implementation tool, "Google Colab" is used. In our proposed method model, it outperforms in terms of metrics that include accuracy, recall (Sensitivity), f-1 score, precision, loss, and AUC(Area Under The Curve). The performance of the framework is evaluated on a dataset that is available on "Kaggle" which is chest X-ray images. This research work has set a new record with a good performance by achieving state-of-the-art results as 0.8831 in terms of AUC and 0.4375 in terms of loss. The Precision, Recall, and f-1 score is obtained as 0.78, 0.81, and 0.794 respectively.

(Arun Kumar)
Advisor

(Manjeet Kaur)
Author

नाम	:	मंजीत कौर	**आईडी नंबर**	:	56669
सेमेस्टर और प्रवेश का वर्ष	:	प्रथम, 2020-21	**उपाधि**	:	स्नातकोत्तर
			विभाग	:	गणित, सांख्यिकी और कंप्यूटर विज्ञान
प्रमुख विषय	:	गणित	**गौण विषय**		कंप्यूटर विज्ञान
शोध शीर्षक	:	**"गहरी तंत्रिका नेटवर्क का उपयोग करके छाती एक्स-रे छवियों में न्यूमोथोरैक्स का द्विआधारी वर्गीकरण"**			
पृष्ठ सं०	:	1-67	**सलाहकार**		डा० **अरुण कुमार**

सारांश

वर्तमान में, फेफड़ों की बीमारियां दुनिया भर में बेहद आम हैं और जिनमें से कुछ में क्रोनिक ऑब्सट्रक्टिव पल्मोनरी डिजीज, निमोनिया, न्यूमोथोरैक्स, तपेदिक आदि शामिल हैं। फुफ्फुस गुहा में हवा को "न्यूमोथोरैक्स" के रूप में माना जाता है। यह एक ऐसी स्थिति है जब फेफड़ों की सतह या छाती की दीवार टूट जाती है, जिससे हवा फुफ्फुस स्थान में प्रवेश करती है और फेफड़े को ढहने का कारण बनती है, यह एक गंभीर स्थिति है और जीवन के लिए खतरा हो सकता है। छाती रेडियोग्राफ विभिन्न चिकित्सा इमेजिंग प्रौद्योगिकियों के बीच वक्ष या फेफड़ों के रोगों का पता लगाने के लिए सबसे महत्वपूर्ण और बड़े पैमाने पर उपयोग की जाने वाली नैदानिक तकनीकें हैं। न्यूमोथोरैक्स का निदान छाती के एक्स-रे, सीटी स्कैन या अल्ट्रासाउंड तकनीकों द्वारा किया जा सकता है, लेकिन छाती का एक्स-रे वक्ष रोगों की स्क्रीनिंग और निदान के लिए सबसे अधिक बार उपयोग की जाने वाली सुलभ रेडियोलॉजिकल विधि है। छाती एक्स-रे से निदान मैनुअल छवि पढ़ने के बजाय स्वचालित प्रणालियों के माध्यम से किए जाने पर बड़ी दक्षता और समय की बचत का हो सकता है। है। छाती एक्स-रे से निदान मैनुअल छवि पढ़ने के बजाय स्वचालित प्रणालियों के माध्यम से किए जाने पर बड़ी दक्षता और समय की बचत का हो सकता है। है। छाती एक्स-रे से निदान मैनुअल छवि पढ़ने के बजाय स्वचालित प्रणालियों के माध्यम से किए जाने पर बड़ी दक्षता और समय की बचत का हो सकता है। न्यूमोथोरैक्स स्वचालित निदान के लिए कई मॉडल निश्चित रूप से उपलब्ध हैं। इस अध्ययन में, हमने छाती एक्स-रे छवियों को लिया है और फिर छाती एक्स-रे छवियों में न्यूमोथोरैक्स की उपस्थिति या अनुपस्थिति का पता लगाने के लिए मशीन लर्निंग के साथ डीप ट्रांसफर लर्निंग तकनीक का उपयोग किया है। हमने डीप लर्निंग और मशीन लर्निंग की अनूठी और कम्प्यूटेशनल विशेषताओं को एकत्रित किया है। हमने डीप ट्रांसफर लर्निंग तकनीक का उपयोग किया है जो एक पूर्व-प्रशिक्षित मॉडल है जो छवि सुविधा निष्कर्षण के लिए एक अवशिष्ट नेटवर्क है और समर्थन वेक्टर मशीन (एसवीएम) एल्गोरिथम जो डेटा बिंदुओं को वर्गीकृत करने के लिए "हाइपरप्लेन" के रूप में जाना जाने वाला सबसे अच्छा निर्णय सीमा बनाता है, न्यूमोथोरैक्स बाइनरी वर्गीकरण के लिए उपयोग किया जाने वाला एक मशीन लर्निंग मॉडल है। प्रभावी परिणाम प्राप्त करने के लिए, वृद्धि के साथ एक संतुलन डेटा तकनीक का उपयोग प्रशिक्षण और सत्यापन डेटासेट के बीच संतुलन बनाने के लिए किया जाता है, साथ ही सत्यापन हानि की निगरानी करने और इष्टतम सीखने की दर प्राप्त करने के लिए "रिड्यूसएलआरओएनप्लेटौ" नामक एक स्वचालित समायोजन सीखने की दर तकनीक का उपयोग किया जाता है। एक कार्यान्वयन उपकरण के रूप में, "गूगल कोलाब" का उपयोग किया जाता है। हमारे प्रस्तावित विधि मॉडल में, यह मैट्रिक्स के मामले में बेहतर प्रदर्शन करता है जिसमें सटीकता, रिकॉल (संवेदनशीलता), एफ -1 स्कोर, परिशुद्धता, हानि और एयूसी (वक्र के तहत क्षेत्र) शामिल हैं। ढांचे के प्रदर्शन का मूल्यांकन एक डेटासेट पर किया जाता है जो "कागल" पर उपलब्ध है जो छाती एक्स-रे छवियां हैं। इस शोध कार्य ने एयूसी के मामले में 0.8831 और नुकसान के मामले में 0.4375 के रूप में अत्याधुनिक परिणाम प्राप्त करके अच्छे प्रदर्शन के साथ एक नया रिकॉर्ड स्थापित किया है। प्रेसिजन, रिकॉल और एफ -1 स्कोर क्रमशः 0.78, 0.81 और 0.794 के रूप में प्राप्त किया जाता है।

(अरुण कुमार)
सलाहकार

(मंजीत कौर)
लेखिका