A Project Report on

Detection of Pneumonia in Chest X-Ray using Transfer Learning Based Approach

Submitted by

Zuber Ansari (Roll no. 57)

Gaurav Bavdane (Roll no. 59)

Neha Bhujbal (Roll no. 61)

Nameera Shaikh (Roll no. 72)

in partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

in

Electronics and Telecommunication Engineering

Under the Guidance of

Ms. Pallavi Patil

St. Francis Institute of Technology, Mumbai

University of Mumbai

2020 - 2021

CERTIFICATE

This is to certify that Zuber Ansari, Gaurav Bavdane, Neha Bhujbal, Nameera Shaikh are the bonafide students of St. Francis Institute of Technology, Mumbai. They have successfully carried out the project titled "Detection of Pneumonia in Chest X-Ray using Transfer Learning Based Approach" in partial fulfilment of the requirement of B. E. Degree in Electronics and Telecommunication Engineering of Mumbai University during the academic year 2020-2021. The work has not been presented elsewhere for the award of any other degree or diploma prior to this.

(Guide Name)

(Dr. Gautam Shah)	**(Dr. Sincy George)**
EXTC HOD	**Principal**

Project Report Approval for B.E.

This project entitled ***'Detection of Pneumonia in Chest X-Ray using Transfer Learning Based Approach'*** by **Zuber Ansari, Gaurav Bavdane, Neha Bhujbal, Nameera Shaikh** is approved for the degree of Bachelor of Engineering in Electronics and Telecommunication from University of Mumbai.

Examiners

1. - - - - - - - - - - - - - - - - - -

2. - - - - - - - - - - - - - - - - - -

Date:

Place:

ACKNOWLEDGEMENT

We are thankful to a number of individuals who have contributed towards our final year project and without their help it would not have been possible. Firstly, we offer our sincere thanks to our project guide, Guide-Name for his constant and timely help and guidance throughout our preparation.

We are grateful to all project co-ordinators for their valuable inputs to our project. We are also grateful to the college authorities and the entire faculty for their support in providing us with the facilities required throughout this semester.

We are also highly grateful to Dr. Gautam A. Shah, Head of Department (EXTC), Principal, Dr. Sincy George, and Director Bro. Jose Thuruthiyil for providing the facilities, a conducive environment and encouragement.

Signatures of all the students in the group

(Zuber Ansari)

(Gaurav Bavdane)

(Neha Bhujbal)

(Nameera Shaikh)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signatures of all the students in the group

(Zuber Ansari)

(Gaurav Bavdane)

(Neha Bhujbal)

(Nameera Shaikh)

Abstract

Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans. According to World Health Organization(WHO), one in three deaths in India is caused due to pneumonia. Early diagnosis can provide a significant chance for correct treatment and survival. Deep Learning techniques during the last few decades had tremendous impact on various fields be it image recognition or speech recognition. It is also highly relevant for medical imaging. There is lack of data availability as some of the medical data is subjected to patient privacy issues while the outbreak of a new disease also arises the same issue. Current trend in deep learning technique involves training a model over large dataset and exposing it for testing but this convention might not hold against real world applications where uncertainty is common syndrome. Thereby the proposed method, use of transfer learning technique, serves a multi prong solution as it need not require data from the same feature space. Particularly training a large CNN architecture (ResNet50) over a large ImageNet Dataset then transferring the weights of initial layer and fine-tuning the last layers will result in a higher precision and recall value and faster performance in terms execution time as compared to existing methods. In this work, the collected dataset is passed through six different preprocessing steps before it is fed to the ResNet-50 module, in order to improve the validation and classification accuracy of the proposed model and achieve remarkable test accuracy. The same methodology will also hold good for any detection and localization of abnormality in Medical Images (eg. classification of Covid-19) with consistent performance which involves even multi-class classification problems.

Keywords: *Pneumonia, Chest X-ray, ResNet-50, Medical Image Processing, Deep Convolutional Neural Networks, Transfer Learning*

Contents

Acknowledgement **iii**

Declaration **iv**

Abstract **v**

List of Figures **viii**

List of Tables **ix**

List of Abbreviations **x**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Methodology 2

1.4 Organization of Project Report 3

2 Literature Review **4**

2.1 Pneumonia Detection Using CNN based Feature Extraction 4

2.2 Chest X-ray Disease Diagnosis with Deep Convolutional Neural Networks 6

2.3 Abnormality Detection and Localization in Chest X-Rays using DCNN 8

3 Software and Hardware Support **10**

3.1 Python 10

3.2 Pip Packages 10

3.3 Online Platform 11

3.4 Offline Platform 11

3.5 Hardware Requirement 11

4 Proposed Methodology 12

4.1 Data Collection 12

4.2 Data-Preprocessing Stage 13

4.3 Feature Learning Stage 14

4.3.1 ResNet50 14

4.3.2 Transfer Learning 19

4.3.3 ResNet50 + Transfer Learning 22

4.4 Classification Stage 23

5 Results and Discussion 25

5.1 Learning curves 25

5.2 Output of train loss vs validation loss 26

5.3 Model evaluation 28

5.4 Testing the model on real chest x-rays 31

5.5 Testing the model for multi-class classification 33

6 Conclusion and Future Work 35

6.1 Conclusion 35

6.2 Future Work 36

References 37

Appendix-I: Flow chart of the Project 39

List of Figures

2.1 Architecture of proposed model [1] 6

4.1 Proposed System Architecture 12

4.2 Single Residual Block [2] 15

4.3 ResNet-50 Architecture [3] 15

4.4 ResNet Layer [2] 17

4.5 Traditional ML approach vs Transfer Learning [4] 20

4.6 Transfer Learning using Feature Extraction [4] 21

4.7 Transfer Learning using Fine-Tuning [5] 22

5.1 Loss vs Learning rate 26

5.2 Diagnostic Plot showing output of proposed architecture 27

5.3 Diagnostic Plot monitoring validation set for different metrics w.r.t epochs 28

5.4 Confusion matrix for results of proposed model 30

5.5 Image of pneumonia infected patient 32

5.6 Predicted output for image in Figure 5.5 32

5.7 Confusion matrix for multi-class classification 33

6.1 Flow chart of the Project 40

List of Tables

5.1 Confusion Matrix 29
5.2 Evaluated metrics 31

List of Abbreviations

CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Networks
FC	Fully Connected
FN	False Negative
FP	False Positive
ReLU	Rectified Linear Activation function
ResNet	Residual Network
SVM	Support Vector Machine
TL	Transfer Learning
TN	True Negative
TP	True Positive

Chapter 1

Introduction

1.1 Motivation

The potential of AI through machine learning and big data is rewiring the way everyday life is approached. Man has been successful in leveraging its power in different healthcare departments like radiology, oncology, ophthalmology, etc. These innovations are catalyzing the existing systems and transforming them into smarter and efficient mechanisms to tackle the problem at hand.
Firstly, an increased focus on clinical diagnosis and disease identification can greatly help in automating the traditional healthcare services. NITI Aayog's national strategy for AI envisages "AI for all" for inclusive growth of the nation and identifies healthcare as one of the key areas of focus for social impact [6]. Most importantly, India has a thriving AI start-up ecosystem with cutting edge solutions being developed day by day. There is thus an increasing potential in the rise of an AI-centric economy in India.
Secondly, the convention and orthodoxy behind ML approach are that training data and testing data must be from the same feature space and have the same distribution. This assumption may not last while approaching unprecedented real-world applications where uncertainty is common syndrome. For example, during the latest Covid-19 situation there was uncertainty in the source of the virus, its symptoms, existence of its variants, etc. The situation was triggering and alarming which was evident by the declaration of lockdown throughout the subcontinent within a short notice of 4 hours. The effort is to explain that there

might not be enough data available in the same feature space to detect the difference between two or more classes.

Finally, for a classification task in one domain of interest while having sufficient training data only in another domain of interest, the latter data may be in a different feature space or follow a different data distribution altogether. In such a situation, it can be concluded that knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data labelling efforts, hectic task of rebuilding the model from scratch as well as save a lot of time invested in training.

1.2 Problem Statement

Detection of pneumonia with transfer learning using fine-tuning will serve a multi-prong solution as it does not require data from same feature space making it fit for real world applications.

1.3 Methodology

In this project, the functionality of a pre-trained CNN model namely, ResNet-50 architecture has been used followed by Transfer Learning for the classification of abnormal and normal chest X-Rays. Pneumonia can be misleading because of many other problems like congestive heart failure, lung scarring etc. which can mimic pneumonia. Even for a trained radiologist, the task at hand of examining the chest X-rays is challenging. Moreover, the availability of an expert radiologist in rural areas is scarce. Currently, deep learning-based methods cannot replace trained clinicians in medical diagnosis and they aim to supplement clinical decision making. This is because, despite several advantages of using X-ray imaging, in some cases it is not possible to identify the correct region of interest in the radiographic image for detecting the disease. Thus, there is a need to improve the diagnosis accuracy and its accessibility in rural areas as well.

The scope for AI in healthcare is huge, specifically in medical diagnosis. Making a prototype to solve medical problems is not a competition but an opportunity to join forces and create an impact on the world. With collaborative efforts it is possible to create a health technology ecosystem for matching demand, optimizing costs and demonstrating value output. It will greatly improve the way doctors diagnose and treat diseases.

The development of an algorithm for detecting thoracic diseases like Pneumonia would increase the accessibility of clinical settings in remote areas as well without any delay. The goal is to enhance the diagnostic accuracy of the automated system to reach the human level. The proposed model is capable of classifying whether a patient is infected with pneumonia or not with a high confidence level. With this project, the aim is to beat the state-of-the-art result, improve reliability on automated disease diagnosis systems and make disease classification more efficient and faster.

1.4 Organization of Project Report

This project report is organized as follows:

- Literature Survey: In the chapter 2 of this report, a survey of literature to study the work that has already been done in the field of Deep Neural Networks and its applications in disease classification is covered.
- Software and Hardware requirement: In the chapter 3 of this report, a brief introduction to the software support used is provided.
- Body of the Project: In the chapter 4 of this report, the design overview of the project is explained and covered.
- Results and Discussion: In the chapter 5 of this report, the results for Detection of Pneumonia in Chest X-Ray using Transfer Learning are shown and discussed.
- Conclusion and Future work: In the chapter 6 of this report, the concluding comments and scope for future work is discussed.

Chapter 2

Literature Review

In this chapter, different research papers [1], [7], [8] are discussed. Majorly, one paper has been summarized in brief. The papers referred are as follows:

1. Pneumonia Detection Using CNN based Feature Extraction
2. Chest X-ray Disease Diagnosis with Deep Convolutional Neural Networks.
3. Abnormality Detection and Localization in Chest X-Rays using DCNN

2.1 Pneumonia Detection Using CNN based Feature Extraction

- In this paper [1], the author aims to study different CNN models for feature extraction followed by different classifiers for the classification of abnormal and normal chest X-Rays. The paper is divided into three stages.

- Stage 1: Pre-processing stage: The original 3-channel images were resized from 1024 x 1024 into 224 x 224 pixels. This helped to reduce heavy computation and enabled faster processing.

- Stage 2: Feature Extraction Stage: This stage involves the extraction of the feature vector which acts as input for Classification (Stage 3). Feature learning was done using different CNN models and the statistical results obtained proposed DenseNet-169 as the optimal model.

- Why use DenseNet-169?

  Initially, Deep Convolutional Neural Networks (DCNNs) [9] were the most productive due to the peculiar types of Convolutional and Pooling layers. But as the network gets deeper, the input information or the gradient passing through the layers vanishes by the time the last layer is reached. This is known as the vanishing gradient problem [10]. DenseNets overcome this problem of gradient vanishing as they connect all the layers with equal feature sizes directly with each other. Thus, the chief motive of using Dense Nets is that deeper the network, more generic features can be obtained.

- Architecture of DenseNet-169:

  DenseNets have direct connection from any layer to the other layer, therefore improving the flow of gradient. This requires the concatenation of the feature-maps of the preceding layers which cannot be done unless all the feature-maps are of the same sizes. But as CNNs primarily intended towards the down sampling of size of feature-maps, the DenseNets architecture is divided into multiple densely connected dense blocks. As mentioned in Figure 2.1, there are 4 dense blocks which consist of a Transition layer in between each block. The Transition layer also facilitates in removing the redundant features.

- Extraction of Features: The output obtained from Stage II was flattened to obtain the feature vector of dimension 50176 x 1

- Stage 3: Classification Stage: Different classifiers such as Random Forest, K-nearest neighbors, Support Vector Machine etc. were used for the classification task among which SVM turned out to outperform on the basis of selection of kernel (gaussian radial basis function) and 350 combination of parameters (C, gamma) for maximum AUC score.

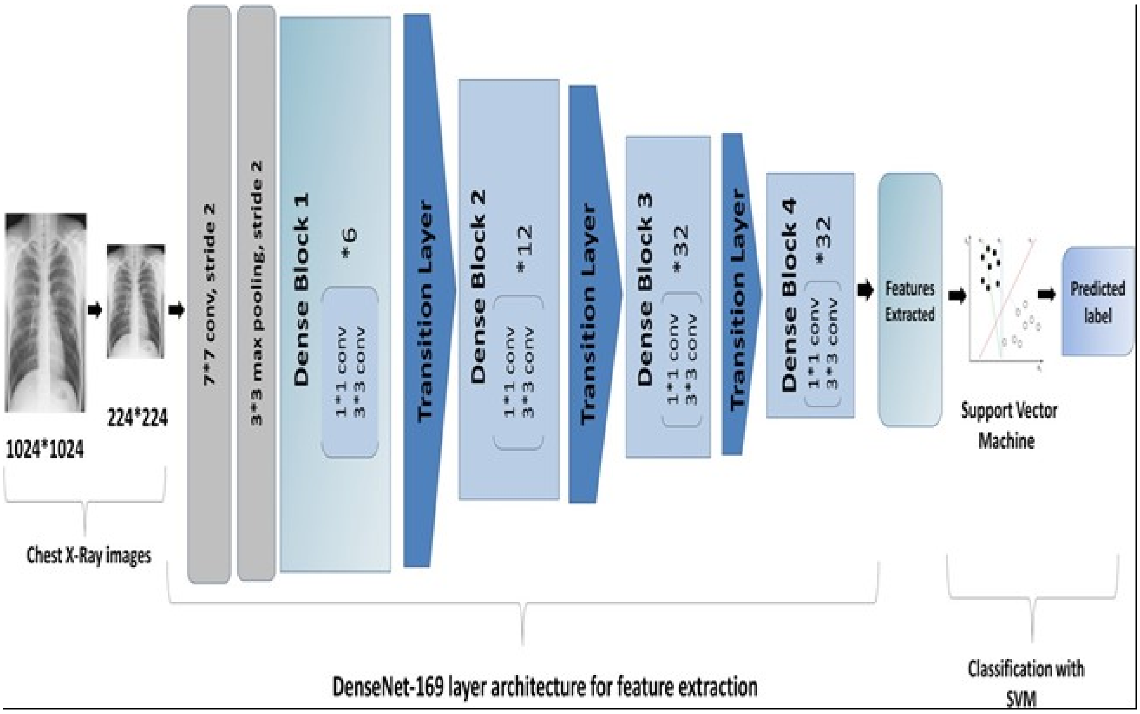

Figure 2.1: Architecture of proposed model [1]

- Model proposed by the author:
  DenseNet-169 + SVM Classifier
  (feature extractor) (Gaussian rbf kernel at C=3.5, gamma=1.9 e-05)
- AUC Score: 0.8002
- Knowledge about different variants of CNN models was acquired from this paper which helped to further extend the study on ResNet-50.

2.2 Chest X-ray Disease Diagnosis with Deep Convolutional Neural Networks

- In this paper [7], the authors proposed development of multi-label CNN to (1) detect and (2) localize the 14 thoracic pathologies
- They trained the whole network of different CNN architectures like DenseNet 121, 169 and 201, Resnet50 etc followed by hyper parameter changes and provided a comparative analysis.

- The results for ResNet and DenseNet architecture were better than the baseline results from previous research. Deeper networks provided better approximation of the relationship between chest X-rays and the disease labels. These architectures were compared based on AUC.

- What is AUC?
  AUC stands for Area Under the ROC Curve. It measures the two-dimensional area enclosed by the ROC curve. In order to study AUC, it is essential to understand ROC curve. The ROC curve or Receiver Operating Characteristic Curve is a graph which shows the performance of a classification model at all possible classification thresholds. The AUC provides an aggregate measure of performance across every possible threshold for classification.
  The values of AUC range from 0 to 1, with 0 indicating all the predictions are wrong and 1 indicating all the predictions are correct.

- In addition to leveraging CNN on X-ray images, they also implemented a simple Feed-Forward Neural Network (FNN) incorporating socio-demographic patient information like age, gender and diagnosis history, having one hidden layer and a sigmoid activation layer with an aim to improve patient-level prediction models.

- The significance of using sigmoid function was to produce independent probability estimates for each diagnosis label.

- The FNN standalone did not perform well with respect to AUC due to the relatively naive architecture.

- Furthermore, pathology localization and visualisation were performed with the help of heat maps generated using the weights obtained in the final layer of CNN. This greatly helped to visualize specific thoracic conditions and analyse areas within the image that led to disease diagnoses.

- Key takeaways from this paper are: patient information is not a reliable metric for classification of diseases and training deeper networks enables efficient feature learning.

2.3 Abnormality Detection and Localization in Chest X-Rays using DCNN

- In this reference paper [8], the author's have used the publicly available Indiana chest X-Ray dataset, JSRT dataset and Shenzhen Dataset and studied the performance of known deep convolutional network (DCN) architectures on different abnormalities.

- The author had reported DCN based classification and localization on the publicly available datasets for chest X-rays.

- In this paper, the quality of detection was evaluated in terms for four measures: Accuracy, Area under receiver operating characteristics (ROC) curve (AUC), Sensitivity and Specificity.

- Accuracy: The accuracy is the ratio of number of correctly classified samples to total samples.

- ROC curve: It is the graphical plot of true positive rate (TPR) vs false positive rate (FPR) of a binary classifier when classifier threshold is varied from 0 to 1.

- Sensitivity: Sensitivity also known as True positive rate (TPR) is the proportion of pathological samples that are correctly identified as pathological sample

- Specificity: Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This proportion could also be called a false positive rate.

- Sensitivity based localization provides correct localization for spatially spread out diseases.

- The paper states that, when the number of training examples is low, a consistent detection result can be achieved by doing multiple train-test with random data split and the average values are used as the accuracy measure.

- Shallow features or earlier layers consistently provide higher detection accuracy compared to deep features.
- Here, ensemble models to improve classification significantly compared to single model when only DCNN models are used, are also implemented. Combining DCNN models with rule based models degraded the accuracy.
- Ensemble learning combines the predictions from multiple neural network models to reduce the variance of predictions and reduce generalization error. Techniques for ensemble learning can be grouped by the element that is varied, such as training data, the model, and how predictions are combined.
- Whereas in the proposed methodology Transfer Learning has been employed. A pre-trained model has been used over a large Dataset (ImageNet) followed by finetuning the later layer according to defined task.

Chapter 3

Software and Hardware Support

3.1 Python

Python is a high level, general purpose, object oriented, interpreted programming language developed in the 1970's.Python's design philosophy emphasizes code readability with its notable use of significant whitespace.Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Advantages of using Python: Despite there being many programming languages to choose from, the proposed model has been developed using Python due to the following reasons:

- Ease of learning and simple syntax formatting
- Provides powerful and easy-to-use libraries for learning and evaluating deep learning modules.
- Open source
- Vast community support
- Open and free updates

3.2 Pip Packages

- PyTorch

- Torchvision
- fastai and fastbook
- Pandas, NumPy, sklearn, matplot
- PIL

3.3 Online Platform

- Kaggle (for Data Resource)
- Colab (for kernel)
- Drive (to store data)
- Paper Space.

3.4 Offline Platform

- Jupyter Notebook (if using prefer! git clone @ 1,2,3)

3.5 Hardware Requirement

- If not using 3.3 procedure, then System with:
  o Nvidia installed GPU (this is must because of cuda)
  o RAM higher than or equal to 8GB
  o Processor more than 4 core

Chapter 4

Proposed Methodology

In this chapter, the proposed system design is explained. It comprises four sections: Data Collection (4.1), Data Preprocessing Stage (4.2), Feature Learning Stage (4.3), Classification Stage (4.4). The complete model architecture is illustrated in Figure 4.1

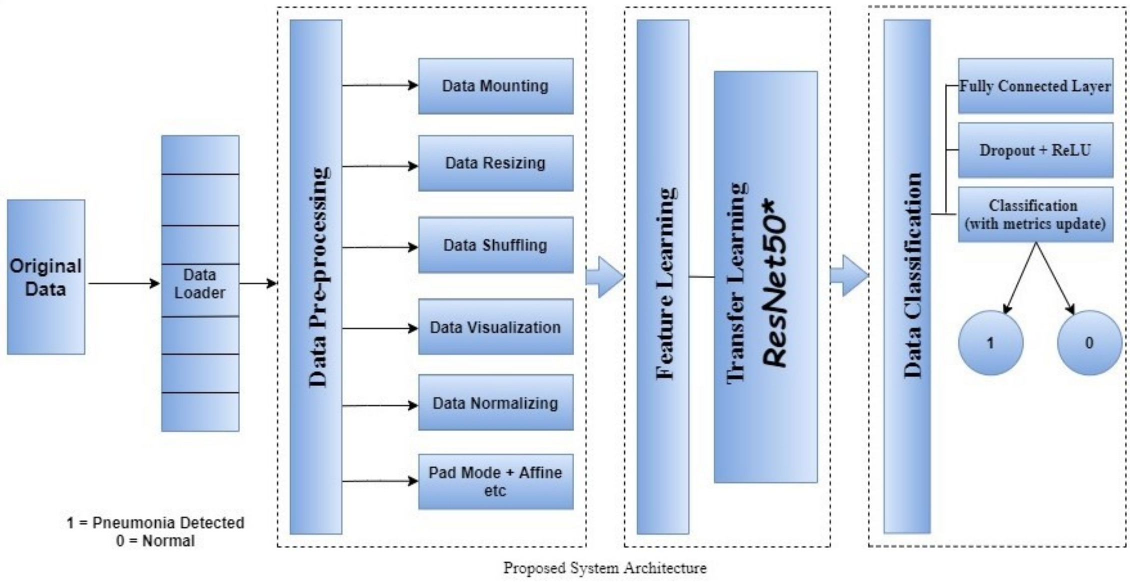

Figure 4.1: Proposed System Architecture

4.1 Data Collection

Data which has been used for training the model is Chest X-Ray Images (Pneumonia) available on Kaggle which is a subset of the Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification acquired from Mendeley Data in association with Elsevier [11]. Mendeley Data is a secure cloud-based repository where one can store data and ensure easy sharing and access. The dataset has 3 sections: training set, test set and validation set. Each section is divided into two classes that is Normal Chest X-Ray and Pneumonia detected Chest X-Ray. The dataset consists of 5k plus images.

4.2 Data-Preprocessing Stage

In the Data Preprocessing stage, following steps have been incorporated:

- **Data mounting**: It is achieved by making structured representations of data in files, subfiles, etc. This is made accessible for further use in the preprocessing stage.

- **Data Resizing**: Data resizing is used to bring all the different sized images to the same size by downscaling them to match the dimensions from the smallest image available.

- **Data Shuffling**: It is very important that the dataset is shuffled well to avoid any element of bias/patterns in the split datasets before training the ML model. This improves the model quality and predictive performance. Further the data is also split into train, validation and test set.

- **Data Visualization**: Data visualization is the representation of data or information in a graph, chart, or other visual format. It communicates relationships of the data with images. This is important because it allows trends and patterns to be more easily seen.

- **Data Normalization**: Normalization is a rescaling of the data from the original range so that all values are within the range of 0 and 1. Normalization requires that to accurately estimate the minimum and maximum observable values.

- **Padding**: Sometimes an image dataset is such that edges are important for analysis. So, while downscaling i.e., data resizing is performed edges get vanished, so to retain these edges even after data resizing, use data padding where zero padding is done.

- **Affine transformation**: An affine transformation is any transformation that preserves collinearity and ratios of distances. Examples of affine transformations include translation, scaling, homothety, similarity, reflection, rotation, shear mapping, and compositions of them in any combination and sequence.

4.3 Feature Learning Stage

In machine learning, the stage of feature learning allows a machine to learn the features and use them to perform a specific task such as classification or prediction. In this project, feature learning is accomplished by customizing a pre-trained CNN namely, Resnet-50.

4.3.1 ResNet50

ResNet, short for Residual Networks, is a classic neural network used as a backbone for many computer visions tasks [12]. This model was the winner of the ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed to train extremely deep neural networks with 150+ layers successfully [13] Prior to ResNet training very deep neural networks were difficult due to the problem of vanishing gradients discussed in Section 2.1. The next sub-subsection covers how ResNet solved this problem of vanishing gradients.

Skip Connection : The Strength of ResNet

ResNet first introduced the concept of skip connection which is shown in Figure 4.2.

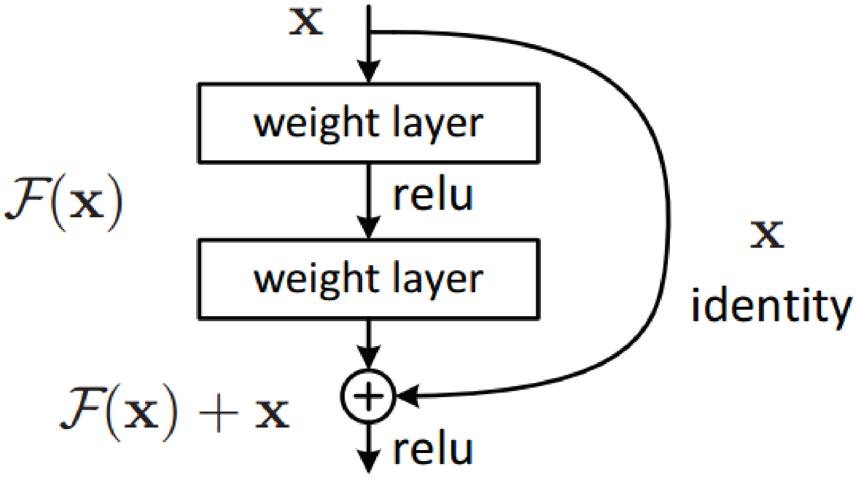

Figure 4.2: Single Residual Block [2]

There are two reasons why skip connections work:

1. They mitigate the problem of vanishing gradient by allowing this alternate shortcut path for the gradient to flow through.

2. They allow the model to learn an identity function which ensures that the higher layer will perform at least as good as the lower layer, and not worse.

ResNet-50 Model Architecture

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters. The architecture is illustrated in the figure below.

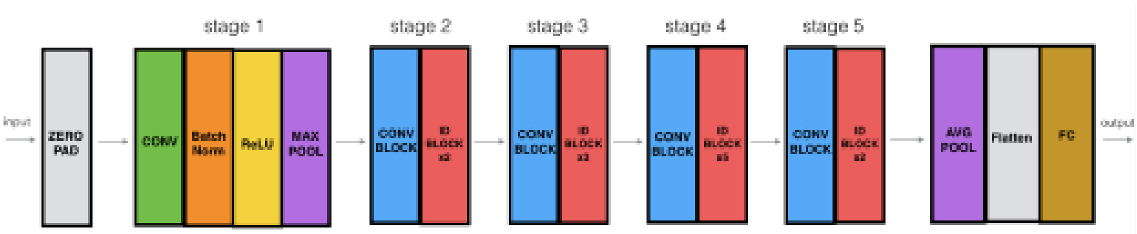

Figure 4.3: ResNet-50 Architecture [3]

The Resnet-50 architecture contains the following elements:

- A convolution with kernel size of 7 x 7 and has 64 different kernels all with a stride of size 2 giving **1 layer**.

- Next it contains max pooling layer with a stride of 2.
- In the next convolution operation, there is a 1 x 1, 64 kernels followed by a 3 x 3, 64 kernels and at last a 1 x 1, 256 kernels. These three layers are repeated 3 times so giving a total of **9 layers** in this step.
- Next there is a kernel of 1 x 1, 128 after that a kernel of 3 x 3, 128 and at last a kernel of 1 x 1, 512 this step is repeated 4 times thus rendering **12 layers** in this step.
- After that there is a kernel of 1 x 1, 256 and two more kernels with 3 x 3, 256 and 1 x 1, 1024 and this is repeated 6 times giving a total of **18 layers**.
- After that average pooling is applied, then a fully connected layer containing 1000 nodes and at the end a softmax function all of which contributes as 1 layer.
- And then again, a 1 x 1, 512 kernels with two more of 3 x 3, 512 and 1 x 1, 2048 and this is repeated 3 times giving a total of **9 layers**.
- After that average pooling is applied, then a fully connected layer containing 1000 nodes and at the end a softmax function all of which contributes as **1 layer**.
- Only the convolutional layers are counted and not the activation functions or the max/average pooling layers.Summing up all the layers, **1 + 9 + 12 + 18 + 9 + 1 = 50 layers** of Deep convolutional network is achieved.
- Each 2-layer block present in ResNet-34 is replaced with a 3-layer bottleneck block, resulting in a 50-layer ResNet (Figure 4.4). This model has 3.8 billion FLOPs (floating-point operations per second). It is a method of encoding real numbers within limits of finite precision available on computers. Using floating-point encoding, extremely long numbers can be handled relatively easily.
- A **Bottleneck Residual Block** is a variant of the residual block that utilizes 1 x 1 convolutions to create a bottleneck. Using this bottleneck reduces the number of parameters and matrix multiplications.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 4.4: ResNet Layer [2]

- The main idea is to make residual blocks as thin as possible to increase depth and have less parameters. These bottlenecks have been introduced as part of the ResNet architecture, and are used as part of deeper ResNet such as ReNet-50 and ResNet-101.

- The first step on the ResNet before entering the common layer behavior is a block called here Conv1 consisting of a **convolution + batch normalization + max pooling operation**.

- **Conv2D** is the most common type of 2D convolution layer and is usually abbreviated as conv2D. A filter or a kernel in a conv2D layer has a height and a width and is generally smaller than the input image and it is moved across the whole image.

- **Batch normalization** is a technique designed to automatically standardize the inputs to a layer in a deep learning neural network. Batch normalization is mostly used to accelerate the training of deep learning neural networks in Python with Keras.

- **Max pooling** also call as maximum pooling is a pooling operation that calculates the maximum value in each patch of each feature map. The results are then down sampled or pooled feature maps that highlight the most present feature in the patch.

- ResNet50 network starts with a Convolution layer with a kernel shape of 7 x 7 and 64 kernels (filters), with a stride of 2. It is followed by Batch Normalization and ReLU activation. Batch Normalization is used after every convolutional block and before the activation function.

- The **Rectified Linear Activation function or ReLU** for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance [14]. The popularity of ReLU is due to some limitations of sigmoid and tanh activation functions. The function is given by:

$$y = max(0, x) \tag{4.1}$$

- This is then followed by a Max Pooling layer of 3 x 3, and a stride of 2.

- It is then followed by subsequent ResNet (Residual / Bottleneck) blocks. There are 4 Resnet blocks, and the number of kernels and the kernel size in each block is shown in figure 4.4.

- From Figure 4.4 it can be seen that a kernel size of 7 is used, and a feature map size of 64. The output size of that operation will be a (112x112) volume.

- Since each convolution filter (of the 64) is providing one channel in the output volume, a (112x112x64) output volume is achieved which is free of the batch dimension to simplify the explanation.

- Similarly, the output shape after each layer can be computed until the global average pooling layer. Finally, for the FC layer the number of input nodes is 2048 and the output layer contains 1000 classes originally for the ImageNet challenge.

4.3.2 Transfer Learning

Transfer learning [15] is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.
It is a popular approach in deep learning [16] where pre-trained models are used as the starting point on Computer Vision and Natural Language Processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

Common Assumptions on Machine Learning and Deep Learning:

- If the train and test data are drawn from the same feature space and the same distribution, then already built model can be reused however when the distribution changes, the model needs to be built from scratch. This requires collection of new training data.
- It is expensive to recollect the needed training data and rebuild the models. To reduce the effort of recollecting data, knowledge transfer or transfer learning between task domains is used.

The objective of Transfer Learning is to take advantage of data from the first setting to extract information that may be useful when learning or even when directly making predictions in the second setting.
With reference to Figure 4.5, using transfer learning one needs to only retrain the classifier on the dataset as compared to the traditional machine learning approach which requires to train data from scratch.
If there is scarcity of data as in medical domain data is less due to some hospital policy of not revealing data. On such problems where enough data is not available, transfer learning can help to develop skillful models that simply could not be developed in the absence of transfer learning. The choice of source data or source model is an open problem and may require domain expertise and/or intuition developed via experience.

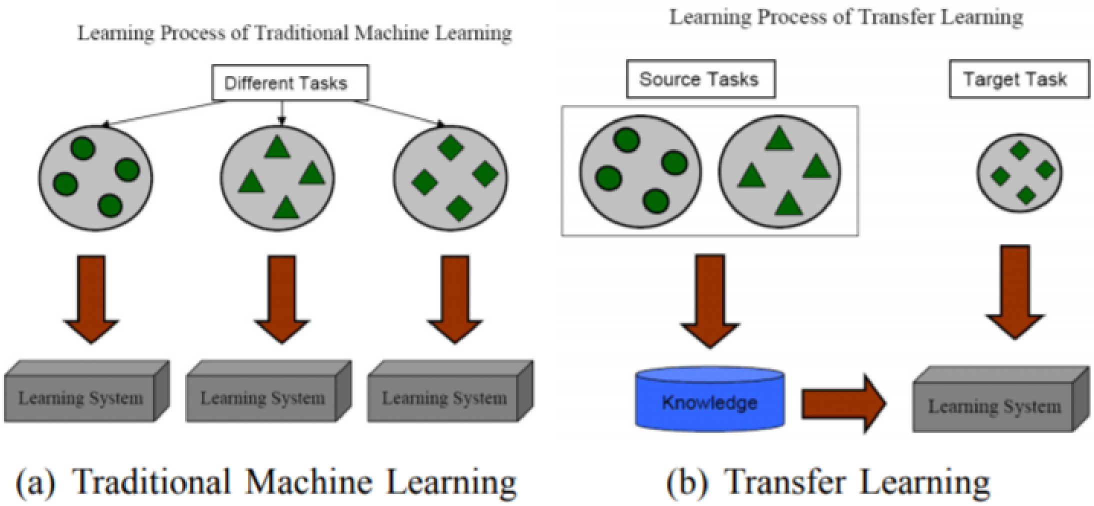

Figure 4.5: Traditional ML approach vs Transfer Learning [4]

Working of transfer Learning

- Neural networks usually try to detect edges in the earlier layers, shapes in the middle layer and some task-specific features in the later layers.
- In transfer learning, the early and middle layers are not used and only the higher layers are re-trained. It helps to leverage the labeled data of the task the model was initially trained on.

Approaches to Transfer Learning

1. **Using feature extraction**

- Implement TL, remove the last predicting layer of the pre-trained model and replace them with the predicting layers FC_T1 and FC_T2 shown in Figure 4.6.
- Weights of these pre-trained models are used as a feature extractor.
- Weights of the pre-trained model are frozen and are not updated during the training.

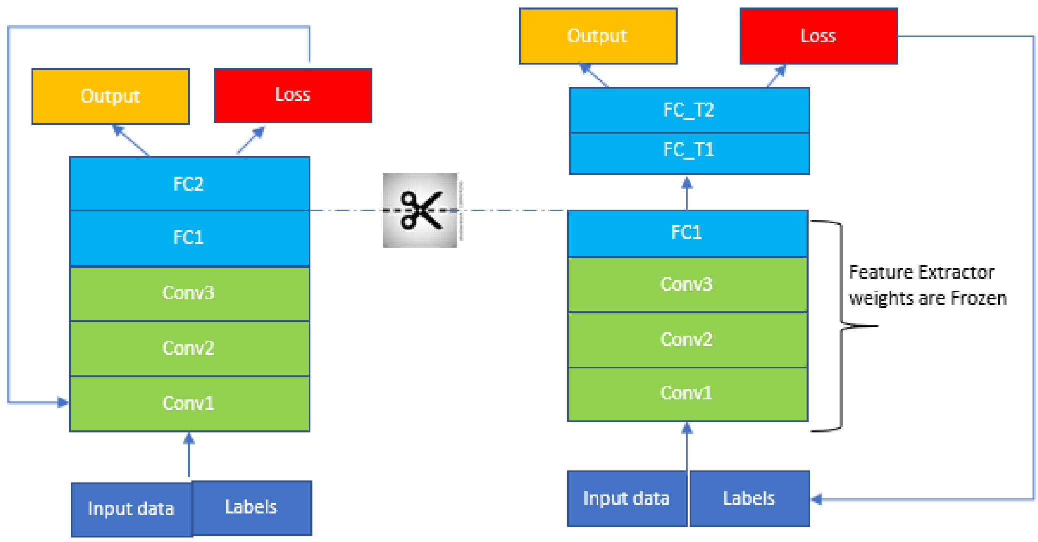

Figure 4.6: Transfer Learning using Feature Extraction [4]

2. **Fine Tuning the Network**

- Instead of random initialization of weights, the network is initialized with a pretrained network like the one trained on ImageNet dataset.
- Unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model. This allows to "fine-tune" the higher-order feature representations in the base model in order to make them more relevant for the specific task.
- This technique indeed increases the speed of learning. With fine-tuning, one is not limited to retraining only the classifier stage (i.e., the FC layers), but it also incorporates retraining of the feature extraction stage, i.e., the convolutional and pooling layers.
- Thus, in the proposed work, fine-tuning of the learning parameters of the pre-trained network has been implemented to accommodate new learning tasks.

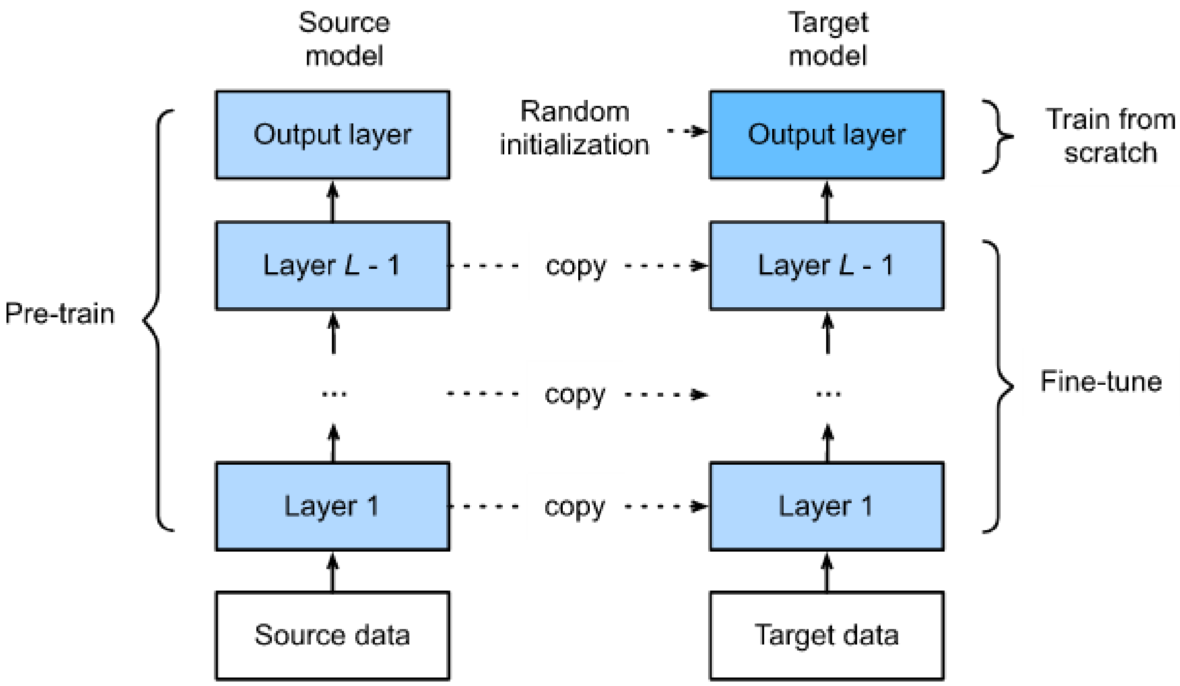

Figure 4.7: Transfer Learning using Fine-Tuning [5]

4.3.3 ResNet50 + Transfer Learning

In the proposed model, Transfer Learning has been employed to train the powerful ResNet-50 network followed by fine tuning of the network parameters and hyperparameters.

- This is achieved by loading the model of ResNet-50 with pre-trained weights (parameters) from the ImageNet Dataset.
- Here, the last fully connected layers are not loaded which acts as a classifier. This is done in order to add new fully connected layers on top of the ResNet50 model for task-specific classification.
- The initial lower layers of the network learn very generic features from the pre-trained model. To achieve this, the weights of the initial layers of the pre-trained model are freezed and not updated during the training.
- The higher layers are used for learning task-specific features. Higher layers of pre-trained models are trainable or fine-tuned. This improves performance with less training time.
- The output obtained from the pre-trained model i.e., convolution and pooling

layers are 3-D volumes. But, the FC layer in Classification Stage (4.4) expects a 1-D vector of numbers.

- Therefore, the output obtained from ResNet-50 is flattened to a vector of size 1 x 30720. Flattening simply refers to rearranging the 3-D volume of numbers into a 1-D vector. This acts as input for the Classification Stage (4.4).

Transfer Learning provides a much faster and easier learning/training as compared to training a network from scratch. In other words, transfer learning is an optimization technique as it acts as a shortcut to save training time along with giving better performance.

4.4 Classification Stage

- Data classification refers to the process of predicting and segregating a given dataset into different classes by assigning labels.

- According to the proposed architecture (Fig 4.1), the feature vector obtained from ResNet-50 is passed through a FC layer. The FC layer acts as a classifier for the extracted features and assigns a probability for input image classification. While CNN [17] detects different features in an image, the FC layers learn how to use these features in order to correctly classify images.

- The FC layers are configured with dropout regularization technique to prevent overfitting. During training time, at each iteration, a neuron is temporarily dropped or disabled with a probability p. These dropped out neurons are resampled with probability p at every training step. This means that a dropped-out neuron at one step can be active at the next one.

- After this, the neurons are activated with $ReLU$ function. The $ReLU$ function is *MAX(X,0)* that sets all negative values in the matrix X to zero while all other values are kept constant. The reason for using $ReLU$ over other activation functions like sigmoid is that it tends to converge much more quickly and reliably.

- Finally, to lay out the probabilities for the two classes (i.e., Normal, Pneumonia), Sigmoid Function is used and is defined as follows:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

where 'x' is the weighted sum of inputs and bias

Chapter 5

Results and Discussion

The designed pipeline which feeds in bunch of data, then those data are converted into data loader format which simply mean dividing data in batches which will run parallelly over CPU's n-core for increasing the computational speed, according to Figure 4.1, after Transfer learning, a vector is obtained which is then classified according to feature learning process. The model has been fine-tuned to find the optimal learning rate using learning curves.

5.1 Learning curves

A learning curve is a plot of model learning performance over experience or time as represented in Figure 5.1. Learning curves are a widely used diagnostic tool in machine learning for algorithms that learn from a training dataset incrementally [18]. The model can be evaluated on the training dataset and on a holdout validation dataset after each update during training and plots of the measured performance can created to show learning curves. The common practice is, opting out a sloppiest section of the graph and using values perpendicular to it over x-axis to be a good learning rate for model to start with.

From Figure 5.1 it can be observed that the suggested learning rate during training is 0.1. Fitting the model with learning rate = 0.1 the model is run for 20 epochs. This is because training the large network involves optimization of a large set of parameters which are highly interdependent. Thus, given the large number of parameters to be learnt the model will take a lot of training samples before the network settles

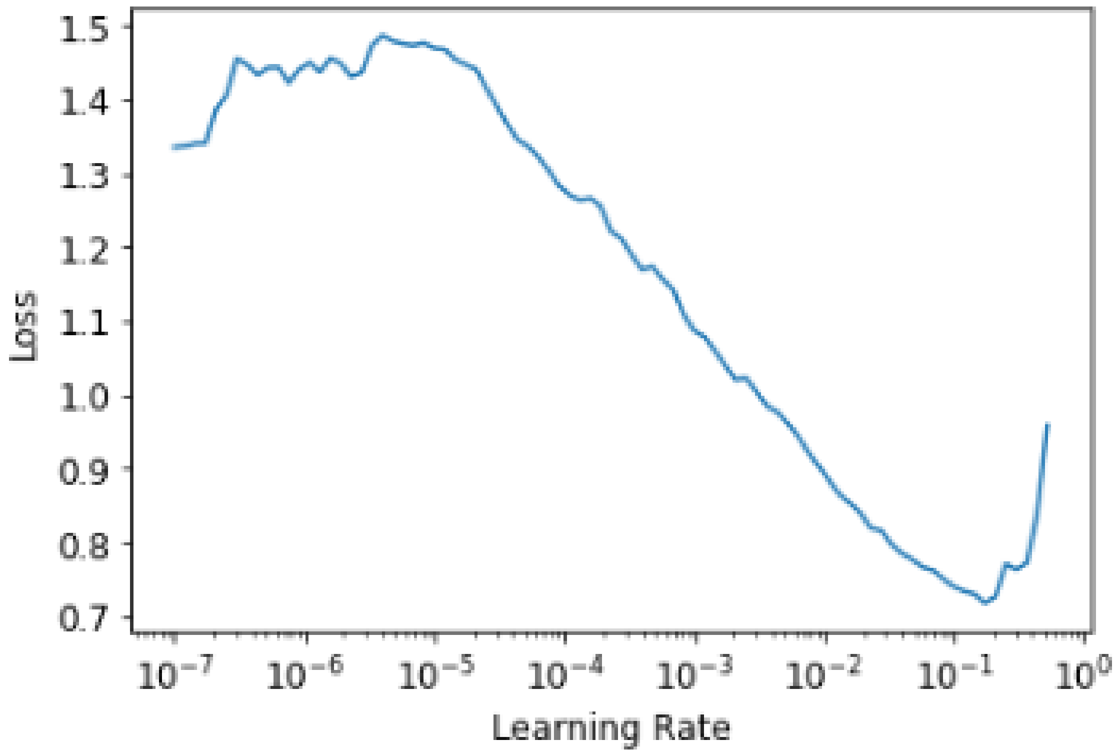

Figure 5.1: Loss vs Learning rate

anywhere close to the solution space of the optimal solution. There may be enough information present in the training samples but the optimization algorithm takes time to extract it. And as the size of the dataset used is small therefore multiple passes over the data need to be done in order for the algorithm to converge and provide best results.

5.2 Output of train loss vs validation loss

The plot of training loss vs validation loss helps to judge the goodness of fit of the trained model. This process of visual tracking of model performance helps to find whether the model is underfitting, overfitting or near about right. These inferences can be drawn with the following points:

- A model is said to be overfitting on the training data if the training loss is very high as compared to the validation loss. In other words, the model has learnt the feature representation of the train data well but shows poor generalization for any other data.

- A model is said to be underfitting on the training data if the training loss is lower than the validation loss. In this case the model shows poor performance on training data as well as poor generalization for other sets of data.

- Finally, an optimal model is one that is able to solve the above-mentioned drawbacks and provide results which can be reliable. In an ideal case, with increase in the number of epochs, the training loss should be almost equal to the validation loss.

Figure 5.2 represents the output of proposed methodology whereby initially training loss and validation loss showcase huge difference but as number epoch increases, the differences start to converge. Because data in training the model is getting increased apart from training over a number of epochs and fine tuning enclosed in transfer learning helps the model to understand and generalize over unseen data. The loss is calculated using the cross-entropy loss.

Cross-Entropy loss :Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.

$$loss = \sum p_i \times \log(q_i) \tag{5.1}$$

where p_i= *Original Probability,* q_i= *Estimated Probability*

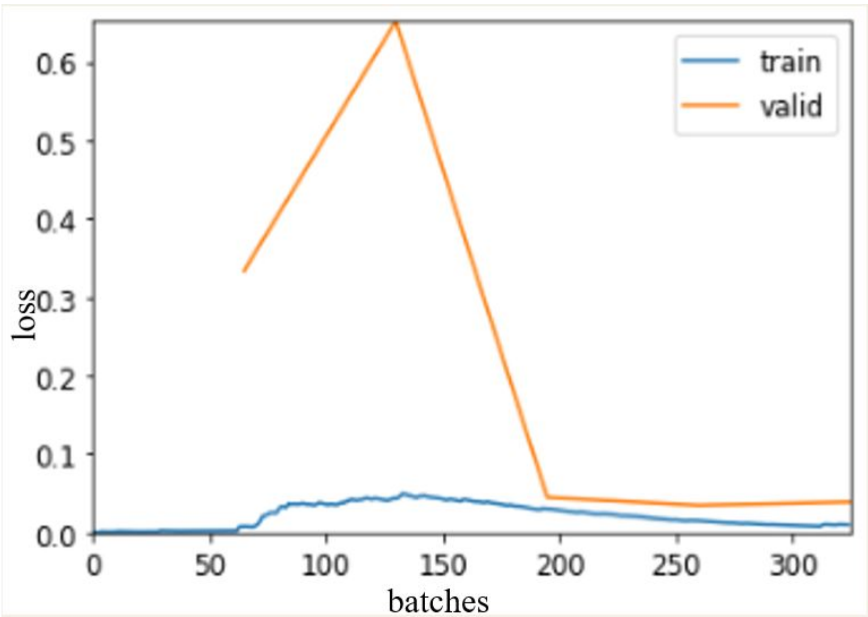

Figure 5.2: Diagnostic Plot showing output of proposed architecture

Furthermore, the diagnostic plots of validation loss for all the metrics namely, accuracy, error rate, precision, recall and f1 score are depicted in Figure 5.3.

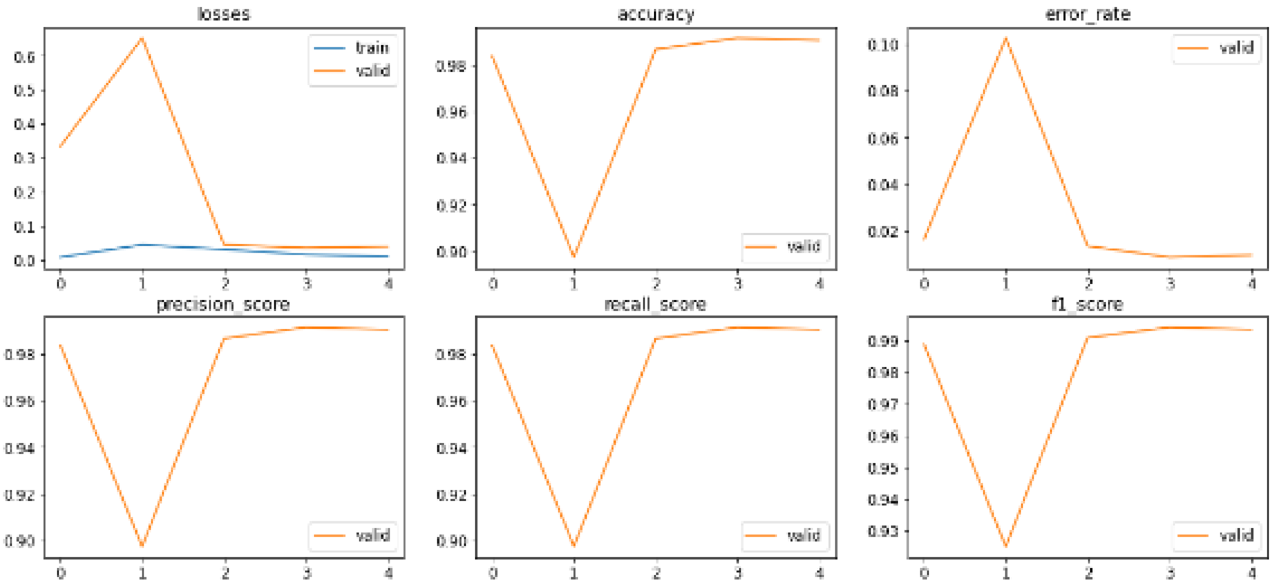

Figure 5.3: Diagnostic Plot monitoring validation set for different metrics w.r.t epochs

From the graphs shown in Figure 5.3 it can be inferred that the scores for each evaluated metric shows large deviations in the initial epochs and the scores start showing uniformity with increasing training epochs. The number of epochs can be set to any number but the training should be terminated when the validation error starts to increase or attains a point of saturation where it does not increase or decrease with further increase in training epochs. From Figure 5.3 it can be inferred that the model attains optimal performance after 5 epochs.

5.3 Model evaluation

The output obtained is then evaluated and the metrics are computed accordingly. In order to derive deeper insights from the classification results a cleaner and unambiguous way of presenting the results of a classifier is by using the confusion Matrix shown in Table 5.1. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 5.1: Confusion Matrix

With the help of the confusion matrix a number of metrics like Accuracy, error rate, precision, recall and f-1 score can be calculated. These metrics used to evaluate the model performance are defined as follows:

- **Error rate (ERR)** : Calculated as number of all incorrect predictions divided by total number of data samples. The best is 0.0 worst is 1.0

$$ERR = \frac{FP + FN}{P + N} \tag{5.2}$$

where P= All positive classes, N= All Negative classes

- **Accuracy** : It is simply defined as the fraction of correct predictions for the test data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.3}$$

- **Precision score (P)** : It helps to understand how precise/accurate the designed model is. It states that out of all the predicted positive values, how many are actually positive. It is a good measure to determine when the cost of False Positive alarms is high.

$$Precision = \frac{TP}{TP + FP} \tag{5.4}$$

- **Recall score (R)** : Also known as sensitivity, it tells that out of all the true positive values, how many are actually labelled correctly. This metric is used when there is a high cost associated with False Negatives.

$$Recall = \frac{TP}{TP + FN} \tag{5.5}$$

- **F-1 score** : It provides a single score that balances both the concerns of precision and recall in one number. It is calculated as the harmonic mean of these two metrics.

$$F-1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.6}$$

This metric takes values between 0 and 1 with 1 indicating perfect P and R. Precision, Recall and F-1 score facilitated in identifying any underlying errors. Therefore, the proposed model is identified to have consistent performance with good precision and recall or F1 score. The confusion matrix and a concise summary of the evaluated metrics is displayed in Fig 5.4 and Table 5.2 respectively.

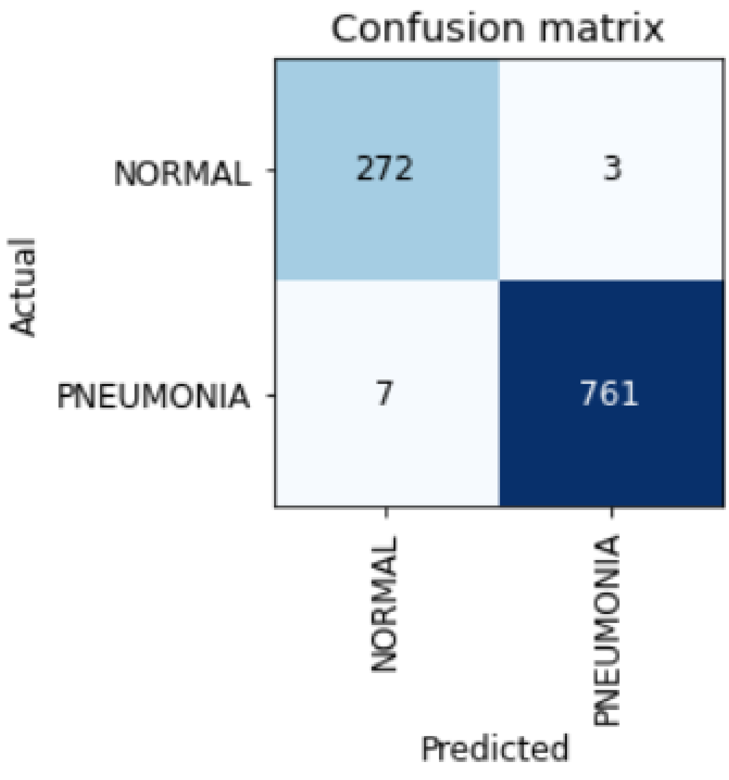

Figure 5.4: Confusion matrix for results of proposed model

Metric	**Score**
Error Rate (ER) = (FN + FP)/Total	**0.009**
Accuracy = 1 - ER	**0.991**
Precision (P) = TP/ (TP + FP)	**0.991**
Recall (R) = TP/ (TP + FN)	**0.991**
F1 Score = 2PR/ (P + R)	**0.994**

Table 5.2: Evaluated metrics

Accuracy is not the only metric, but merely one of the four useful metrics which comment a lot on the overall performance of the system. Specifically, when it comes down to AI systems for making patient diagnoses or reporting findings, it becomes crucial to ensure that these technologies are at par or at least as good as human interpretations. It is certainly not desired to see a day when AI starts to incorrectly tell patients whether they are pneumonia infected or not with an error rate that is higher than expert radiologists.

For the problem statement under consideration, it can be said that having a high recall is crucial as it is important to detect as many correctly labelled patient x-rays with pneumonia as possible. The recall score of 0.991 attained for the model is an appreciable score. At the same time, it is equally important to keep a tract of the patients incorrectly classified as having pneumonia as it can be indicative of some other ailment. In such cases the F1 score is used which strikes out a balance between precision and recall and obtained as the harmonic mean of the two. With the aim of a higher F1 score, the value attained for this metric is 0.994 where a score of 1.0 is indicative of best performance. Thus, on the basis of the attained metrics it can be commented that the proposed model has a good precision and recall as well.

5.4 Testing the model on real chest x-rays

When interacting with any deep learning problem it is crucial to separate the training and testing phase of the model to avoid overfitting and to ensure that the

model has generalized well on unseen data. For the classification problem at hand, the model has been tested on some real chest x-ray images which have been acquired from authentic sources and are no present in the training set.
One of the x-ray images of a patient infected with pneumonia along with the predictions made by the model is depicted in Fig 5.5 and Fig 5.6 respectively.

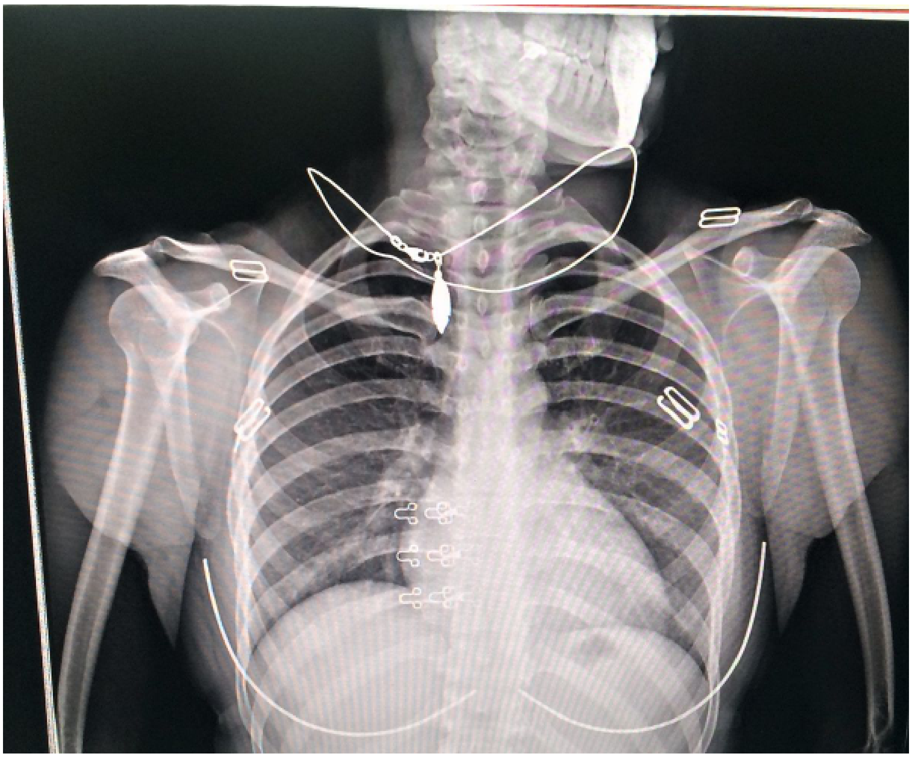

Figure 5.5: Image of pneumonia infected patient

```
img = PILImage.create(uploader.data[0])
is_p,_,probs = learn.predict(img)
print(f"Is Pneumonic?: {is_p}.")
print(f"Probability of Pneumonia: {probs[1].item():.6f}")
```

```
Is Pneumonic?: PNEUMONIA.
Probability of Pneumonia: 0.999915
```

Figure 5.6: Predicted output for image in Figure 5.5

The results show that the model is capable of correctly predicting the output on any unseen data as well. For the above-mentioned example the probability of correctly predicting the patient as pneumonic is 0.9999.

5.5 Testing the model for multi-class classification

With the discussion presented in the previous sections it can be firmly maintained that the proposed architecture is reliable. The same methodology has been tested for multi-class classification problem involving three classes, NORMAL, PNEUMONIA and COVID19. The confusion matrix obtained is displayed in Fig 5.7 below.

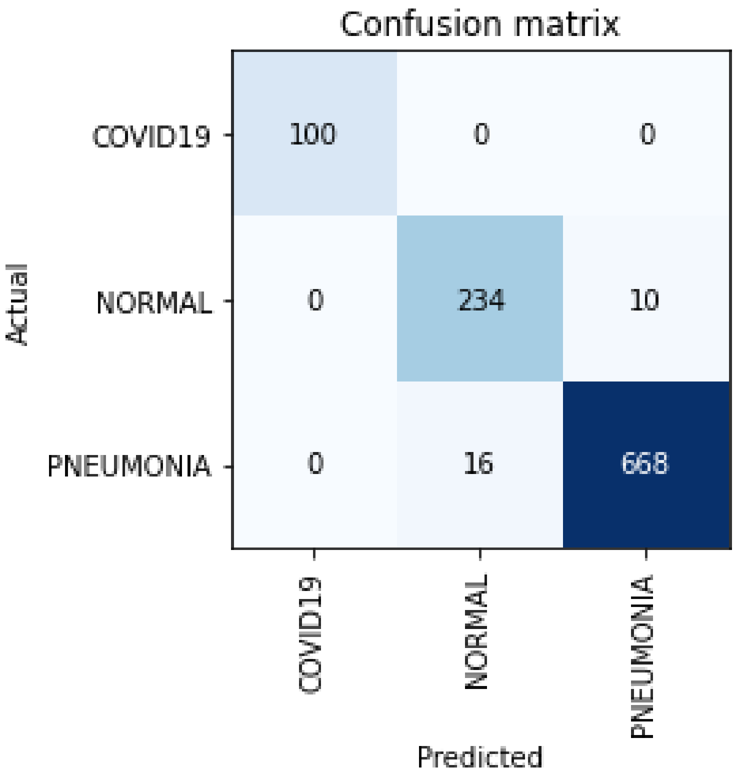

Figure 5.7: Confusion matrix for multi-class classification

The model is able to differentiate well between the three classes. In spite of the dataset being unbalanced it can be obtained that the model is successful in segregating the x-rays with a high level of accuracy of 97.47%. The f-1 score for COVID19 and PNEUMONIA class is 1.0 and 0.98 respectively. Thus, it can be concluded that the model holds good for detection and localization of abnormality in medical images apart from pneumonia with consistent performance for multi-class classification problems.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

- An efficient model for classifying the infected pneumonia chest x-ray images with high level of classification accuracy has been developed. To increase the robustness of the classifier, transfer learning of the powerful ResNet-50 CNN pretrained on ImageNet has been employed.

- When it comes to other pre-trained architectures, DenseNet is considered to be more efficient on some image classification benchmarks. On the other hand, ResNet did outperformed the DenseNet accuracy when accompanied with transfer learning, since transfer learning helped ResNet to avoid initial layer parameter calculation or fitting and to focus over last few layers.

- Therefore, initial layer parameters of ResNet-50 have been accompanied by ImageNet using transfer learning where the last few layers parameter have been finetuned over given data accordingly.

- Apart from this compared with ResNet, DenseNet uses a lot more memory, as the tensors from different layers are concatenated together.

- The development of algorithms in the healthcare domain can be highly effective for providing better services as well as help to prevent adverse consequences (even death) in extreme cases.

- Eventually, the reported results are superior to the automated analysis of pneumonia infected x-ray images reported in literature.

6.2 Future Work

Transfer learning is definitely going to be one of the key drivers for machine learning and deep learning success in mainstream adoption in the industry. Expecting to see more pre-trained models and innovative case studies which leverage this concept and methodology.

The proposed methodology can be further extended for research based on the following points:

- To detect the subclasses of pneumonia like bacterial, viral, etc.
- End-to-end deployment of the project in the form of webapp for easy accessibility and diagnosis. Disease detection can be obtained simply by uploading the x-ray image in the web application which will be evaluated by the model in back-end and classification results will be presented.

One can definitely expect some of the following in future course of time

- Transfer Learning for NLP
- Transfer Learning on Audio Data
- Transfer Learning for Generative Deep Learning
- More Complex Vision problems like Image Captioning

Looking forward to see more success stories around transfer learning and deep learning which enable to build more intelligent systems to make the world a better place to live in as well as drive one's own personal goals.

References

[1] Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan,Ankush Mittal, "Pneumonia Detection Using CNN based Feature Extraction", *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, DOI 10.1109/ICECCT.2019.8869364

[2] Muneeb ul Hassan, "ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks", NeuroHive, Jan 2019

[3] Priya Dwivedi, "Understanding and Coding a ResNet in Keras", Jan 2019

[4] Hafidz Zulkifli, "Understanding Learning Rates and How It Improves Performance in Deep Learning", Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Jan 2018

[5] Renu Khandelwal, "Deep Learning using Transfer Learning -PythonCode for ResNet50", Aug 2019

[6] Niti Aayog, National Strategy On Artificial Intelligence, `http://niti.gov.in/national-strategy-artificial-intelligence`, 2018

[7] Christine Herlihy, Charity Hilton, Kausar Mukadam, "Chest X-ray Disease Diagnosis with Deep Convolutional Neural Networks", 2019

[8] Mohammad Tariqul Islam, Md Abdul Aowal, Ahmed TahseenMinhaz, Khalid Ashraf "Abnormality Detection and Localizationin Chest X-Rays using Deep Convolutional Neural Networks", *arXiv:1705.09850v3*, Sept 2017

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks" Commun. ACM 60 (6), pp. 84â€“90, 2017

[10] S. Jia, "Vanishing Gradient vs Degradation", Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Sep 2018.

[11] Dataset URL: https://data.mendeley.com/datasets/rscbjbr9sj/2?__hstc=25856994.135bb75797d952995f31cdc14442f972.1620393636688.1620730571572.1622131474731.4&__hssc=25856994.1.1622131474731&__hsfp=2145598509, 2018

[12] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", Microsoft Research, 2015

[13] Sabyasachi Sahoo, "Residual blocks- Building blocks of ResNet", *towards data science*, Nov 18

[14] J. Brownlee, "A Gentle Introduction to the Rectified Linear Unit (ReLU)", Machine Learning Mastery Inc., Deep Learning Performance, Jan 2019.

[15] Sinno Jialin Pan and Qiang Yang Fellow, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering, Vol 10, Issue 22*, Oct 2010

[16] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, "Dive into Deep Learning", Nov 2020

[17] Fei-Fei Li, "CS231n: Convolutional Neural Networks for Visual Recognition", Computer Science, Stanford University, 2019

[18] Jason Brownlee, "How to use Learning Curves to Diagnose Machine Learning Model Performance", Machine Learning Mastery Inc., Deep Learning performance, Feb 2019

Appendix-I: Flow chart of the Project

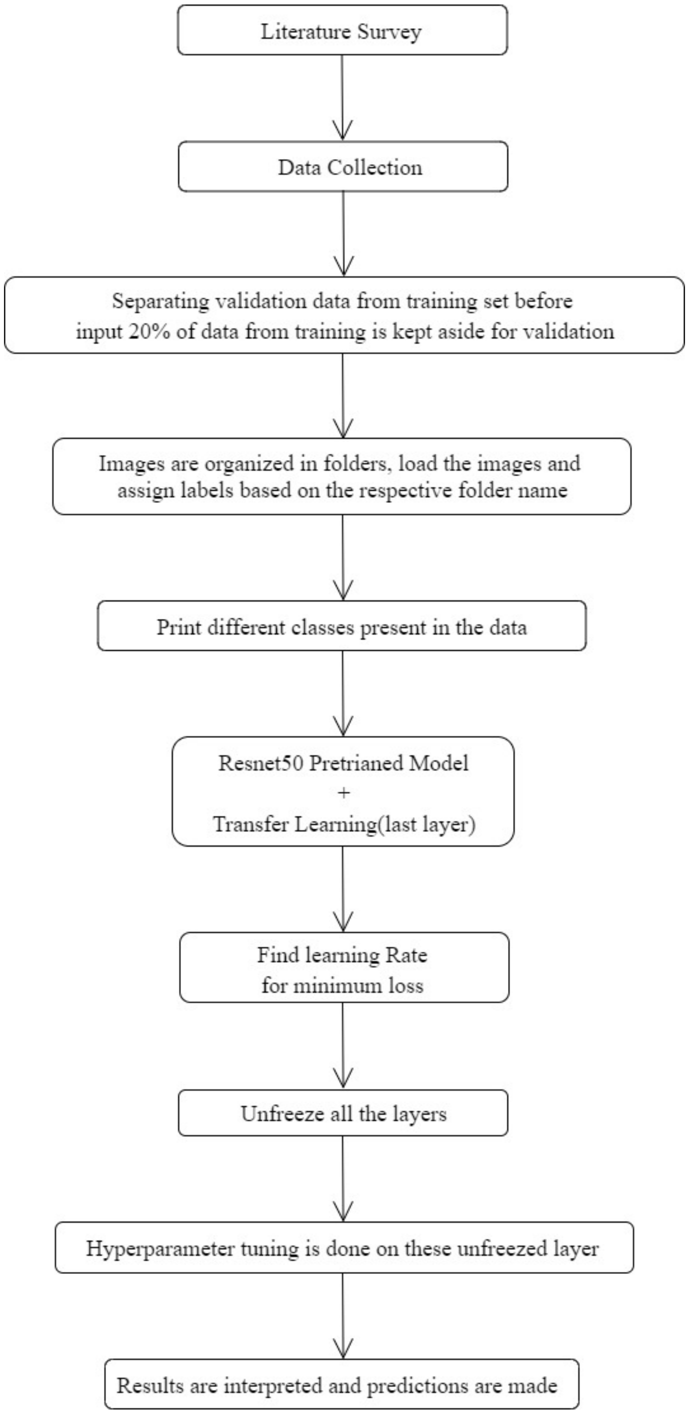

Figure 6.1: Flow chart of the Project