UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

Deep Learning Models for X-ray Image Classification

A Project Report Submitted by

Pentapati Avinash

Supervisor

Prof.Nicola Adami

in partial fulfillment of the requirements for the award of the degree of

Master's Degree in Communication Technologies and Multimedia

University of Brescia

Department of Information Engineering

Telecommunication section

September,2022

Declaration

I, **Pentapati Avinash**, hereby certify that I am the sole author of this thesis titled, **Deep Learning Models for X-ray Image Classification** and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas,techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s)in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Signature

Pentapati Avinash

31/08/2022

Acknowledgements

At the end of this degree, I would first like to mention professor **Nicola Adami** for accepting my dissertation proposal, giving me the opportunity to carry out this work under his guidance, and dedicating his time to me, availability, and essential support in the preparation of the thesis. Because of him, I had the option to do some significant experience in deep learning models. He was likewise truly adept at making sense of and explaining each of the ideas.

I thank my classmates and fellow students for the exchange of views, advice, and mutual help that allowed me to enrich my background knowledge and pass even the most difficult exams.

Sincere and diligent thanks go to my family, who were united during these years. They gave me patience, comfort, and advice. They accompanied me on this journey.

Finally, I'd like to express my heartfelt gratitude to my cousin **Satish Babu**, who supported and, more importantly, set me up. He knew how to pay attention to me and solace me.

Abstract

Chronic obstructive pulmonary disease (COPD) is one of the main reasons for non-communicable causes of death worldwide[16]. In India, COPD, pneumoconiosis (pneumonia), Tuberculosis (TB), and asthma are the most common respiratory diseases that require diagnosis[18]. Post 2019, coronavirus has become the major cause of many deaths throughout the world. While some diseases can be cured, others cannot be cured, but an early-stage diagnosis can help in giving proper medication and prolonging the lives of patients.

Since most of the patients go through a basic X–ray requirement, the objective of the thesis is to use the X-ray to analyse the condition of the patient and detect any kinds of anomalies that can lead to the above-mentioned diseases at an early stage. Since the X-ray is a large image and there is a high level of complexity involved in finding the diseases, deep learning models are required. Some models are available that take care of only 2 diseases. In this manner, there is a necessity to foster another model that can recognize every one of the diseases above mentioned.

The first step of the process is to gather the data from different sources. The data contains X-rays of a normal people and persons with different conditions as mentioned above. Pretrainded models have been fine tuned, enabling thir usage for the detection of multiple diseases. Also, a new in-house model has been be developed from scratch and its effectiveness has be evaluated. Finally, the in-house made model and the available models are compared against each other and the best one has been selected.

This model can be deployed in real time for the fast processing of patient data and give the results very quickly, decreasing the unnecessary load on the doctors. This thesis covers a lot of theoretical aspects of modelling, providing an understanding of deep learning convolution neural networks. It also shows recommendations for future work using segmentation. All in all, one gets to know the building blocks of CNN and apply them to solve real-world problems.

Contents

0.1 Introduction 2
0.1.1 Convolution Neural Network 3
0.1.2 Deep Learning and Neural Network 6
0.2 Related Works 10
0.2.1 Deep Convolutional Neural Networks (CNN's) based Transfer learning 11
0.2.2 Segmentation 12
0.2.3 Description of the Existing methods 13
0.3 Methodology 22
0.3.1 Datasets Description 24
0.3.2 Pre-processing and Augmentation 25
0.3.3 Experiment 26
0.3.4 Performance Matrix 26
0.4 Results & Discussion's 28
0.4.1 Results 28
0.4.2 Discussion 39
0.5 Conclusion and Future works 41
References 43

List of Figures

1.1.1 CNN architecture . . . 5
2.3.1 VGG16 Architecture . . . 16
2.3.2 ResNet50 architecture . . . 17
2.3.3 InceptionV3 architecture . . . 19
2.3.4 ResNet101 Architecture . . . 20
2.3.5 MobileNet Architecture . . . 21
3.0.1 Over View of the complete Experiment . . . 22
3.0.3 Pneumonia . . . 23
3.0.4 Tubercolosi . . . 23
3.0.5 Chest X-ray images . . . 23
4.1.1 CNN Architecture . . . 28
4.1.2 CNN Accuracy . . . 29
4.1.3 CNN loss . . . 29
4.1.4 InceptionV3 Accuracy . . . 30
4.1.5 InceptionV3 loss . . . 30
4.1.6 ResNet50 Accuracy . . . 31
4.1.7 ResNet50 Loss . . . 31
4.1.8 MobileNet Accuracy . . . 32
4.1.9 MobileNet Loss . . . 32
4.1.10 ResNet101 Accuracy . . . 33
4.1.11 ResNet101 Loss . . . 33
4.1.12 VGG16 Accuracy . . . 34
4.1.13 VGG16 Loss . . . 34
4.1.14 InceptionV3 Classification Report . . . 35
4.1.15 Score-CAM heat map on chest x-ray images . . . 36

List of Tables

0.3.1 Dataset Description 24
0.4.1 Performance Matrix 35

0.1 Introduction

Chest x-ray is the most important diagnostic technique for physicians. It helps diagnose and treat a wide range of diseases, such as:

- Pneumonia
- Lung Cancer
- Heart Failure and other Heart problems
- Emphysema
- Other medical conditions(TB, Covid19 etc.,)

If a patient experiences fever and shortness of breath with coughing, their doctor may diagnose pneumonia. If the lungs look inflamed on a chest X-ray, they could be infected with germs called bacteria or viruses. Doctors may perform a physical exam and use a chest X-ray, CT scan of the chest, chest ultrasound, or needle biopsy of the lungs to diagnose a patient's condition. A chest X-ray allows doctors to see the lungs, heart, and blood vessels to determine if the patient has pneumonia.

When the radiologist interprets the x-ray, they look for white spots in the lungs (infiltrates) that identify infection. Currently, Coronavirus is one of the most widely recognized infections that can cause sinus, nose and upper throat contamination. It is the biggest classification for an RNA infection. In most cases, it's hard to tell if the coronavirus or another cold virus is causing you a fever. It can cause pneumonia and other respiratory infections[21]. A healthcare provider collects a sample from the nose (nasal and throat swab), throat (throat swab), or saliva. Then for analysis, the samples are sent to a laboratory. A chest x-ray is a rapid test that may diagnose covid-19. Pooled results showed that chest x-ray accurately analysed Coronavirus in 80.6% of individuals who had Coronavirus[15]. Similar outcomes showed that lung ultrasounds accurately analysed Coronavirus in 86.4% of individuals with Coronavirus[19].

Tuberculosis (TB) is a critical contamination that particularly impacts the lungs. The microorganism that causes tuberculosis is unfold from character to character. Chest x-ray (CXR) is a fast-imaging device that allows for the clean detection of lung abnormalities.

So, it's an important approach for early detection of tuberculosis (TB), and consequently WHO's set out this as an essential approach to stop TB.

In this thesis we're seeking to diagnose illnesses like Pneumonia, TB (Tuberculosis) and Covid-19, with the aid of a data set of various patient's chest x-ray.

Disease detection is a crucial element in prevention. Deep learning is a way with a view to permitting us to teach artificial intelligence to expect outputs with a given information set. We can use each of the supervised and unsupervised learning's to teach Artificial intelligence.

Deep learning is utilized in classifying images (Google images), speech recognition (Siri, Alexa etc.,) Video recommendations (YouTube etc.,) disorder detection, defense, and protection areas. Deep learning represents synthetic neural networks which are stimulated via way of means of the human mind. Just like the human mind, it includes neurons, as the most effective distinction is the quantity and pace of learning. In other words, information units and processing strength are hard to teach in synthetic neural networks.

0.1.1 Convolution Neural Network

Convolutional neural networks (CNNs) use convolution, a mathematical operation where the signal is applied to the input rather than the input. Unlike traditional ANN's each neuron in a CNN learns how to perform a task by analysing examples and gaining experience through training and reinforcement learning.

The network will continue to express a single perceptual score function (the weight) of the input raw image vectors. The last layer of this neural network contains loss functions associated with the classes, and all the regular strategies that you've probably been using for traditional ANN's still apply.

CNN's are specially designed to be trained with images and are usually used in the field of pattern recognition. We can now train your convolutional neural networks with

image-specific features, making them better suited for image-focused tasks.

The traditional kinds of ANN's are limited in terms of the process complexity needed to compute image data. MNIST is a widely used dataset for learning classification algorithms and machine learning tasks, mainly due to its relatively small dimensionality (28×28). The MNIST dataset is a collection of handwritten digits. It is often used as a reference dataset for testing machine vision algorithms. 784 weights can be stored in a single neuron in the primary hidden layer (28*28*1 where 1is vacant keeping in mind that MNIST is normalized to only black and white values).

Considering a larger 64×64 colour image input, the number of weights in a single first-layer neuron increases significantly to $12,288$. To manage the input scale, the network must do this too. There are two problems with this. The first reason is that we do not have unlimited computing power and time to train these huge ANN's. We need a model that can be trained in small amounts of time with little memory requirements, allowing us to take advantage of large computing power.

The second reason is to prevent or reduce the effects of overfitting. It is a common problem in machine learning. This occurs because our networks may be optimized for a specific dataset and are unable to generalize to other types of data. To prevent this issue, we can reduce the amount of training data that our network needs. By doing so, we ensure that our models can generalize across a variety of datasets.

One of the main reasons for reducing the complexity of our ANN's is to ensure that the model has fewer parameters to be trained, thereby reducing overfitting. The better predictive performance is a result of a smaller set of parameters and less likelihood of overfitting.

CNN's architecture consists of three layers. They are convolutional layers, pooling layers, and fully connected layers. For MNIST classification of CNN architecture see in Figure 1.

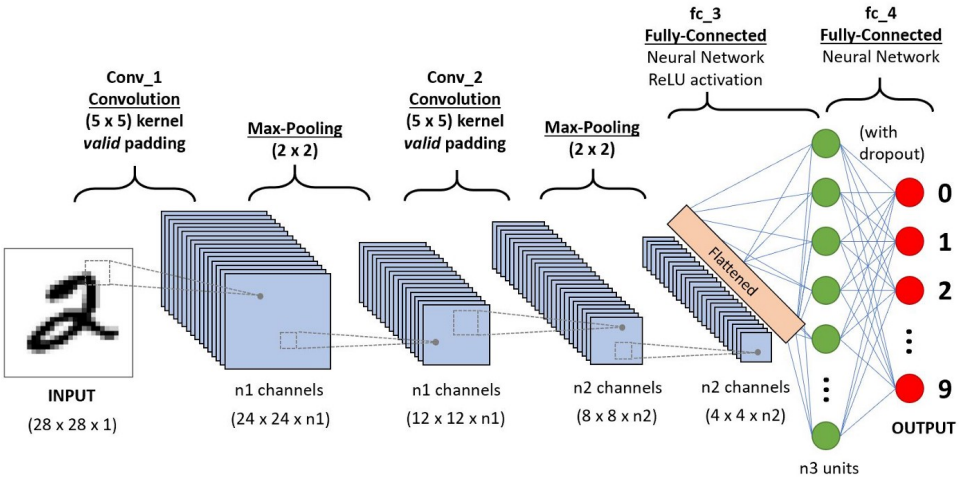


Figure 1.1.1: CNN architecture

The functionality of the CNN above can be divided into four key areas:

1. In different types of ANN, the input layer stores the pixel values of the image.

2. The convolutional layer of a deep neural network determines the output of neurons that are mapped to local regions of the input. The ReLu unit applies an 'elementwise' activation function such as sigmoid to the output of the activation generated by the previous layer.

3. A pooling layer is a type of convolutional layer which is used to reduce the noise and extract more relevant features from the input data by combining several of the same value. The pooling layer is one way to improve the performance at lower levels of abstraction by decreasing the dimensionality of the feature maps.

4. In a fully connected layer, you have multiple smaller than standard ANNs, but they're connected to each other. These layers then perform the same tasks as standard ANNs and attempt to generate class values from the activations, which are used for classification. To improve performance, ReLu can be used in between these layers.

0.1.2 Deep Learning and Neural Network

What is Neural Network?

Neural networks are the most popular machine learning algorithm today. Neural networks mimic the human brain, made up of neurons and synapses. For solving the problem every neuron or node is responsible. They pass what they know and have learned to other neurons in the network for nodes to connect to solve the problem and provide an output. Trial and error are a big part of neural networks and the key to learning nodes. Neural networks are a statistical process (inspired by the human brain) that learns from new information. There is an input of information that flows between interconnected neurons or nodes. These interconnected neurons or nodes within the network use algorithms to learn about them, then the solution is transferred between layers that do the prediction and provide the final determination.

The design of the cerebral cortex motivates brain organizations. At the principal level, the perceptron is the logical depiction of a characteristic neuron.

Neural Network Parts:

Elements that aid in the operation of a Neural Network

- Neurons: neurons that take the output from the layer ahead of it and put it through some function. The output is a number between 1 and 0, representing true or false.
- The input layer and input neurons.
- Hidden layers: It contains a lot of neurons, and a neural network can have more hidden layers inside.
- Output layer: this is where the outcome comes after the data has been fragmented through every one of the hidden layers.
- Synapse: The association among neurons and layers inside a brain organization.

A single neuron, which represents one logical maximum between a piece of in-put and output, can be thought of as a logistic regression model. The neurons take the input feature as input and then combine it with the activation function to produce an output number that it sends to the next neuron.

The network of hidden layers, through which the basic information passes, consists of one or more layers of neurons. In the case of classification, the Probability of a certain event occurrence, or certain values can be predicted by the output layer.

Hidden layers are added to our neural network until the error is negligible. A typical loss measuring standard deviation is used and optimizers are used to ad-just the weights. These can be either gradient descent or Adam, RMSprop etc. In neural networks, we typically use the backpropagation algorithm, which adjusts the weights using an optimizer and a loss function. Backpropagation is an algorithm which adjusts the weights of a network until the error in prediction be-comes negligible. Here, the weighting of the hidden layers adjusts how much each point contributes to the overall accuracy of our model.

How Neural Network Learn

Neural networks must be "learned" so that they begin to function and learn independently. Then they can learn from the results and information they get, but they must start somewhere. There are a few processes used to start training a neural network.

Training: A trained neural network is initially given random numbers or weights. There are two types of training's supervised or unsupervised. Supervised training includes mechanisms to provide grades or corrections to the network. In unsupervised training, the network learns autonomously through its own training strategies. Neural networks can be trained with supervised methods to allow the system to learn more efficiently.

Transfer learning: Transfer learning is a powerful technique in artificial neural networks that makes use of the same solution to learn better insights. Gaining knowledge from previous solutions could be beneficial to use the same predictive methods or methods that have been exhausted during training with new data.

Feature extraction: It takes all the data in the input, removes redundant data, and groups it into more manageable segments. This reduces the memory and processing power required to run a problem through a neural network by feeding the network only the information it needs.

What can a Neural Network do?

So, as soon as we understand what neural networks are, we want to understand how they work. Neural networks have three major applications and understanding what they are will help you get a feel for how neural networks and deep learning are impacting the world of technology.

Classification: Classification refers to the process of assigning a category to a set of data. Classification is a useful feature in neural networks because it separates the data into different classes according to your specifications. Patterns can be found using neural networks within the data. When we use a neural network, we are essentially creating a system that automatically classifies and separates your data into different categories. This can be done in a variety of ways and gives you many more options for classification than if you were to do it by hand.

Clustering: Clustering, like classification, is used to separate similar items. The process breaks down items into smaller groups, but in unsupervised training and neural networks, the groups aren't labelled or separated by your requirements. When researchers need to find the differences between sets of data, clustering can be used.

Predictive Analysis: To predict the future, we utilize predictive analysis in neural networking. Neural networks predict future based on the data they receive. Amazon is a good example of predictive analysis.

Visualization of Neural Network

Visualization of networks is a way to understand how a network decision-making process works. It creates an image that represents the output from a particular layer in the long-term memory of a neural network. There are many types of visualization methods used in neural networks. A good example is GANs, or Generative Adversarial Networks, which are based on adversarial learning techniques.

The key to increase confidence in neural network models is to visualize the logic behind the inference. Visualization techniques increase model transparency by visualizing the logic behind the inference, which can be clarified in a way that is easily understood by humans, thereby increasing confidence in the output of neural networks. The starting point for this strategy is the gradient of the class evaluation function with respect to the input image. This gradient can be elaborated as a sensitivity map, and there are multiple techniques created on this basic idea. They include Smooth Grad, Grad-CAM, Grad-CAM ++, and Score-CAM.

0.2 Related Works

The first CAD system to detect infested lung cells was launched in the late 1980s, but these efforts were not sufficient. Resources weren't available to implement advanced image processing techniques at that time.

The application of image processing techniques to detect diseases such as pulmonary disease is time consuming and tedious. Deep learning models are being used for lung disease detection, which will improve diagnosis, prognosis, and treatment. As artificial intelligence has emerged as a revolutionary and multifaceted approach to solving complex problems is presented in[11] this presents state of the art techniques in the classification and analysis of chest radiographs. The referenced work describes this topic alongside the organization of a novel database of 108,948 radiographs called ChestX-ray8, where 32,717 X-rays are from individual patients. The authors of[11] conducted deep CNN to validate their results on lung imaging data, with promising results.

A database of chest radiographs[7] has become additionally tailored to be used with more than one classification of lung disease. A deep learning framework for lung cancer and pneumonia prediction using two deep learning methods has been proposed in[3] first, a modified AlexNet was used for chest X-ray diagnosis. Additionally, an SVM for classification[3] is implemented in a modified AlexNet. The authors used LIDC-IDRI and the Chest X-ray dataset[4]. These datasets are also utilized in[[14]-[20]]

Extensive work on integrated detection with DenseNet121 and VGG16 is presented in[1] the system is based on computer-aided diagnosis based on deep learning

Detection of pulmonary masses/nodules on chest X-ray[12] images clinically done by CAD systems based on deep learning. Along with that, a deep learning model is suggested in ref[5] for the diagnosis of pneumonia, where several types of transfer learning methods like DenseNet121, AlexNet, Inception V3, etc., are used. It is very difficult to do parameter tuning for the implemented methods.

Performance points for classification task and prediction to a large label dataset called as CheXpert which contains 223,316 chest X-rays from 65,240 patients described in[7], authors assign labels to this dataset based on the views dis-played by the CNN models. This can be done by observing the output generated by lateral and frontal X-rays. Also, [7] uses a standard data set. Moreover, with the highly anticipated availability of large datasets, images of all objects should be easily detected and segmented. So, we need different methods like FCN and F-RCNN[**?**] which can perform both tasks. Mask R-CNN is an extended F-RCNN network which is better in terms of accuracy and efficiency. For object detection and segmentation authors in[17] used the mask R-CNN method and compared their algorithm with others and provided the best algorithm from COCO 2016[6] for lung nodule detection in[13]they used MixNet (fusion of two or more networks).

Based on the above studies, further studies on lung disease detection and classification using large and new datasets are needed.

0.2.1 Deep Convolutional Neural Networks (CNN's) based Transfer learning

As mentioned earlier, deep convolutional neural networks are often used for image classification due to their efficiency. Image features are extracted using convolutional layers and filters in the network.

When the dataset is small, transfer learning can be useful in applications of CNN. Transfer learning is one of the most widely used machine learning techniques, with applications in many fields, including medicine, finance, and marketing. It reduces the time and cost of training a model to perform a new task by using the weights of a previously trained model to improve performance on that task. It can be done in 2 methods: **Feature Extraction:** In this, we use a pre-trained model on a dataset such as ImageNet and train the classification part of the model. Next, we remove the classification part of a network and then let any other algorithm run on the feature extractor.

Fine Tuning: Fine tuning means that you are not only changing the network archi-

tecture and its parameters by training a new model but also modifying the weights of the previous model. In other words, you change both the architecture and weights of a previously trained model. The idea is that doing so should make the system better at recognizing by letting the pre-trained model weights adjust as new data is added.

When data is scarce and there is not enough to draw all the models, transfer learning can help. Transfer learning helps avoid over fitting by randomly sampling the training data and then using those samples as a preview for other data sets.

0.2.2 Segmentation

Segmentation is the process of dividing an image into objects. It involves extracting parts of an image that represent the same object or different object categories. Segmentation is important because of its ability to highlight different features of an image, making it quick and easy for a machine to identify features in an image and develop meaningful uses for them.

A segmentation algorithm processes an image based on some criteria. The result of the processing is a pre-defined binary mask that defines the location and boundaries from which attributes of the data are extracted. This article focuses on several different types of segmentation techniques used in medical imaging. The techniques described here will allow you to classify your images based on specific anatomical features such as brain structure, hemodynamic response, projection structures, or even anatomical features such as bones, muscles, and tissue strains.

Different kinds of Image segmentation:

1. Approach-based classification
    - Region-based approach (detecting similarity).
    - A boundary-based approach (detecting discontinuity).

2. Technique-based classification

- Structural techniques
- Stochastic techniques
- Hybrid techniques

Image Segmentation Techniques:

- Clustering-based segmentation
- Thresholding segmentation
- Region-based segmentation
- Histogram-based segmentation
- Edge-based segmentation

There are several methods of segmentation models, including U-Net, which is widely used in semantic image segmentation and provides better accuracy in medical imaging. The U-net is a fully convolutional neural network with a U-line structure consisting of a contracting path and an expanding path. In this project, we are not working on segmentation.

0.2.3 Description of the Existing methods

This section discusses the advantages and disadvantages of convolutional neural networks, transfer learning, and pre-trained convolutional neural networks.

In this work, we review the top 5 state-of-the-art (SOTA) and widely used pre-trained models for image classification in the industry. Individual models can be discussed in more detail, but this article has been limited to providing an over-view of their architecture.

In image classification, there are several popular datasets used for research. The standouts are:

- Image Net
- CIFAR
- MNIST

Finest pre-trained models for Image Classification:

- VGG-16
- ResNet50
- Inception V3
- MobileNet
- ResNet101

0.2.3.1 Deep learning model: VGG-16

VGG-16 is the best deep learning model for image classification. Developed in the Visual Graphics Group at Oxford University, it was quickly adopted by researchers and industry for image classification tasks. Launched at his famous ILSVRC conference in 2014, this model's popularity is still unmatched by others, and it even continues to be updated with new releases.

The VGG-16 is the most successful and influential deep learning model in human recognition. It is a convolutional network for computer vision and a standard in this area. Its features include fast, accurate, and consistent classification and positioning. It has set a new standard in this field.

VGGNet-16 is also known as VGG-18. It uses a dense fully connected layer followed by 4x4 max pooling for feature extraction. The network design is very similar to the pre-trained VGGNet weights, which allows us to use the exact same weights with AlexNet networks. It has 16 convolutional layers and is imposing with its architecture. Like AlexNet, it has only 3x3 convolutions but many filters.

VGGNet consists of 138 million parameters with a probability distribution, which can be somewhat challenging to train. In order to achieve VGGNet via transfer learning, we use optimized parameters and our model algorithm to train the model on a dataset and then update the parameters with our training data.

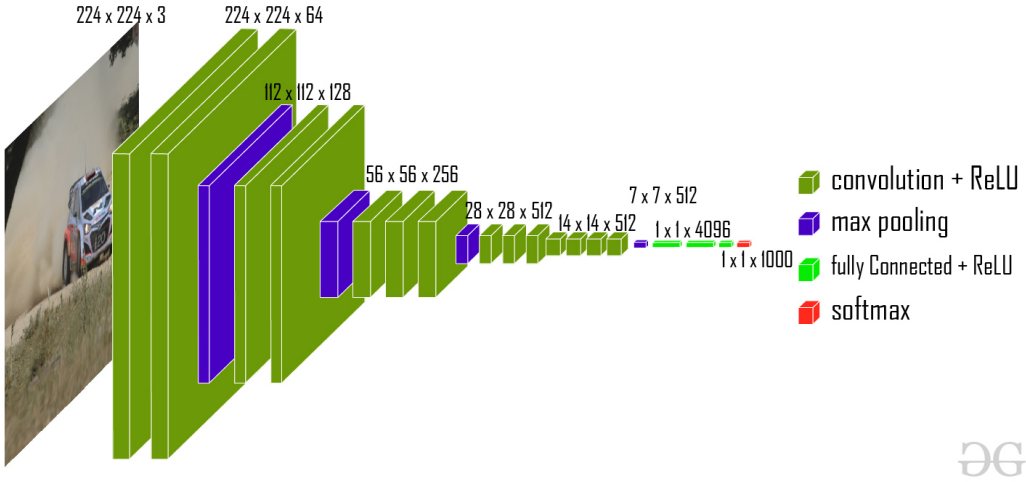


Figure 2.3.1: VGG16 Architecture

The model layers are:

- Convolution layers =13
- Pooling layers = 5
- Dense layers = 3

This model can classify 1000 images into 1000 different categories with 92.7% accuracy, and easy to use with transfer learning. The drawback of this model is that it is slow to train and not many pre-trained networks are available.

0.2.3.2 Deep learning model: ResNet50

In 2015, a new model that can solve the vanishing gradient problem in G.D. was introduced. Its main goal is to avoid bad accuracy when the model goes deeper while improving generalization performance.

ResNet50 is the first model that comes from ResNet's family. It solves the vanishing gradient problem in G.D. by using a new structure called the residual block, which is a type of residual block called the residual net with extra layers added between two blocks. This achieved deeper accuracy without sacrificing regularization strength.

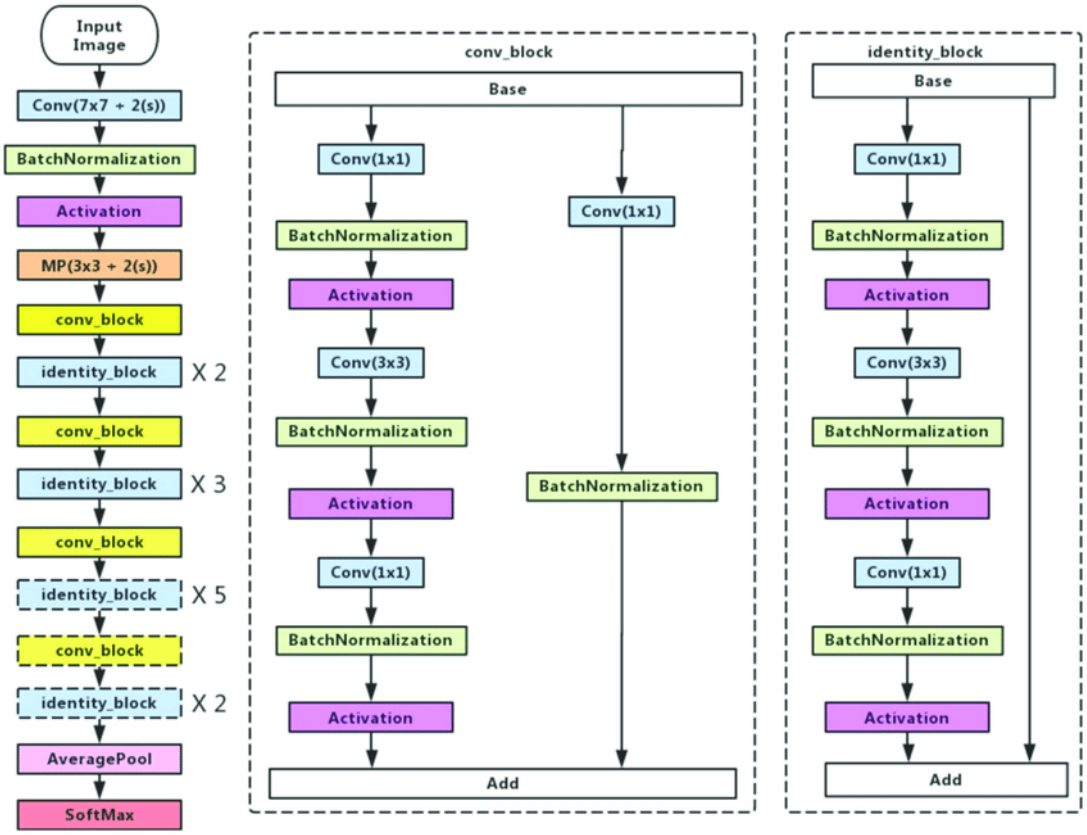


Figure 2.3.2: ResNet50 architecture

Networks with many layers can be easily trained without increasing the percentage of training errors, and they help to cope with the vanishing gradient problem through identity mapping. A residual network is a type of neural network that uses multiple layers to learn the structure of the data. It is useful for learning the correlation and structure of

complex datasets, especially when deep networks are not used. When training a network, it relies heavily on the number of layers and neurons. The more layers and nodes, the better the network generalizes. However, the deeper networks challenged existing hardware because there were too many parameters to be learned. It takes more time because there are a lot of convolutional layers in these architectures.

0.2.3.3 Deep learning model: Inception V3

Inception v3 features a new architecture that is designed to achieve faster training times and better results on standard computer hardware. These improvements are achieved by making use of batch normalization at the layers in the side head, as well as an inception module which acts as a middleware between the input data and the final model. This makes it possible for the system to use weakly-supervised learning methods (which are less accurate in the most challenging cases) with strong supervision for highly accurate predictions.

Inception module, 4-layer Auto encoders. The idea is to use 1*1 convolution with different filter sizes at the input, perform max-pooling, and concatenate for the next Inception module. The introduction of a 1*1 convolution reduces the parameters. The final convolutional layer outputs a single feature map of the input image with ReLu activation. The 1*1 convolution reduces the parameters by half and makes the training hyper parameters more flexible. The inception module takes an input image and applies different filters to it, then performs the same process with a higher resolution. The output of this module is then combined with an additional convolution layer and max pooling layers. This leads to a higher resolution version of the original image.

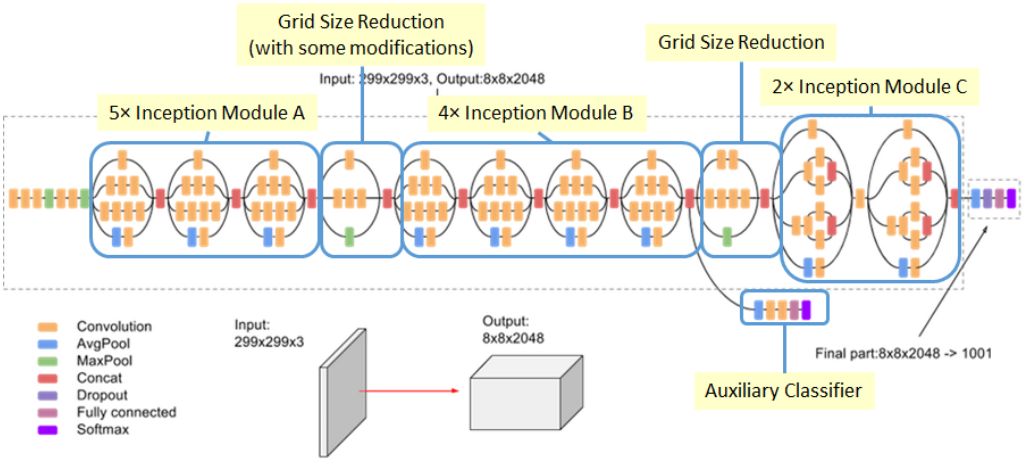


Figure 2.3.3: InceptionV3 architecture

Inception v3 is a fast and efficient deep neural network model that is built for image classification, image recognition, and deep learning algorithms. It has been tested on various AI tasks such as Image Captioning, Text Classification, Object Detection and more.

0.2.3.4 Deep learning model: ResNet101

ResNet-101 is a convolutional neural network with 101 layers. Each layer of ResNet-101 has three or four convolutional filters which collect information from the previous layer, building up a representation of the input image here. You can use a pre-trained model of the network, trained with more than a million images from the ImageNet database.

ResNet-101 is the best open-source deep learning library for computer vision and deep learning. It can classify images into 1000 object categories from images from the ImageNet database.

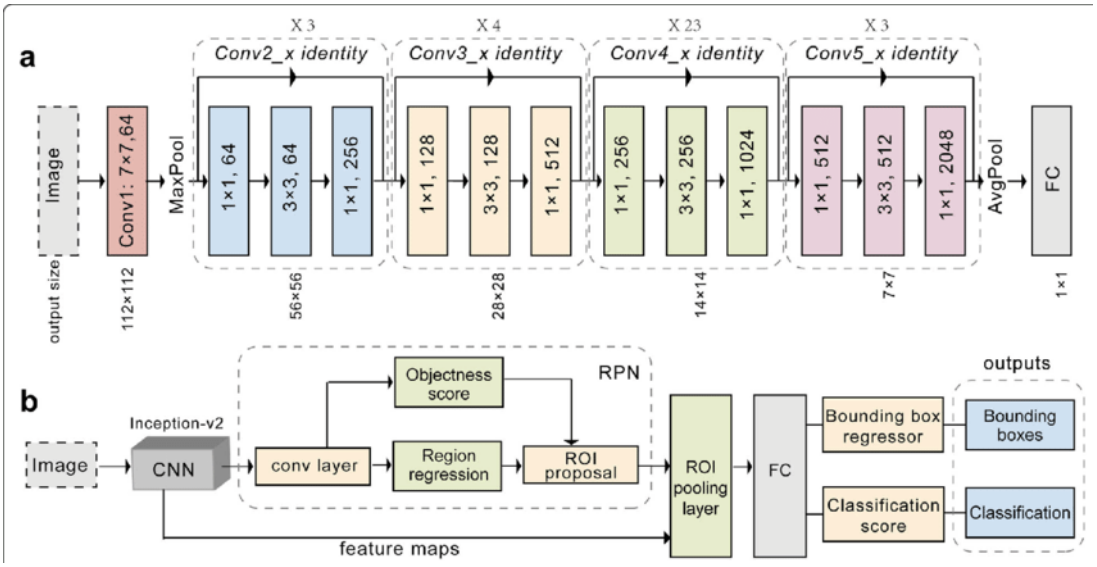


Figure 2.3.4: ResNet101 Architecture

ResNet101 is the backbone of many computers vision tasks, including handwritten digit recognition, classification of over 1000 object categories, and object detection in images.

0.2.3.5 Deep learning model: MobileNet

A neural network which is designed for mobile and embedded imaging applications is MobileNet. It is based on a lean architecture that uses depth-wise separable convolution to build lightweight deep neural networks that can have low latency for mobile and embedded devices, and it is the first mobile computer vision model of Tensor Flow.

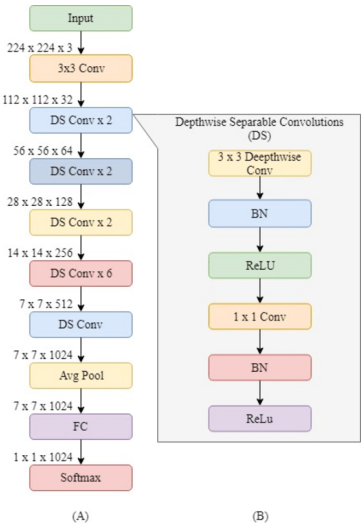


Figure 2.3.5: MobileNet Architecture

MobileNet is an end-to-end model for mobile classification. It combines several different architectures to learn the discriminative connections from visual content to fine-grained semantic categories such as action, appearance, and emotions.

The convolutional layers include:

- 13 3x3 Depth wise Convolution
- 1 3x3 convolutional layer
- 13 1x1 convolutional layers.

This model was developed by Andrew G. Howard and other researchers at Google. MobileNetV2 was developed at Google and trained on the ImageNet dataset of 1.4 million images and 1,000 classes of web images. We use this as a base model to train on our dataset and classify the images of cats and dogs.

0.3 Methodology

As you can see in Figure3.1, there are two different methods (segmentation and without segmentation) where we can classify the images.

In this project, we are working with chest X-ray images without segmentation. In the next, we use the chest X-ray images, after pre-processing and augmentation to train/fine tune the models that will be later used for the classification of images.

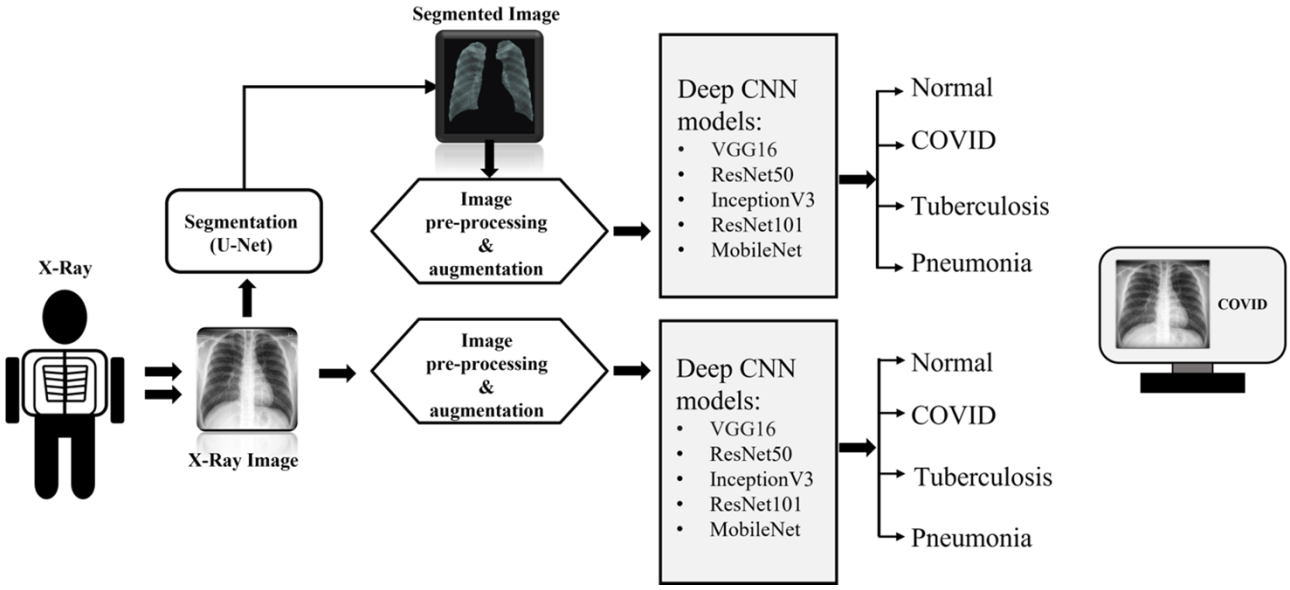
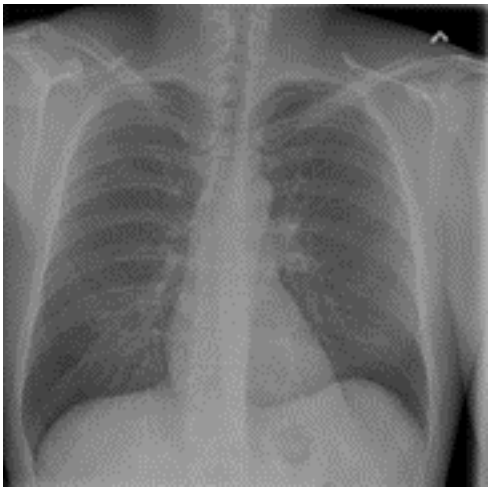


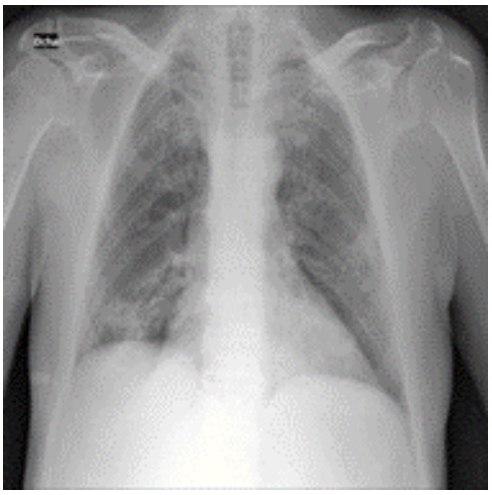
Figure 3.0.1: Over View of the complete Experiment

Analysis of chest X-ray images

A chest X-ray test can be used to diagnose or rule out conditions such as Pneumonia, Pleurisy, and —Tuberculosis. However, chest x-rays are also commonly used to evaluate heart disease, lung cancer, and lymphomas. In this work, Kaggle Chest x-ray images were used for training the models. There are 4 different classes (Normal, Pneumonia, Covid19, and Tuberculous) in the complete dataset. All X-ray images are annotated by expert radiologists. Sample X-ray images are shown below.



(a) Normal



(b) A Covid19

Figure 3.0.3: Pneumonia

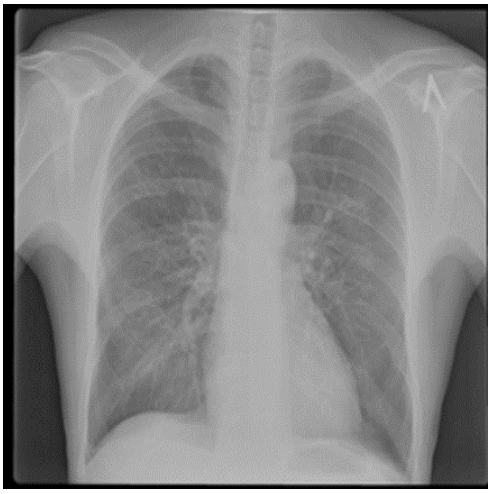


Figure 3.0.4: Tubercolosi

Figure 3.0.5: Chest X-ray images

0.3.1 Datasets Description

This section provides a description of the datasets. The dataset used in this work is composed by images belonging to different datasets as reported in Table1.

In this work, we used a dataset for Pneumonia[9],Dataset for Tubercolosis[8],Dataset for Covid19[10] which is a combination of three different datasets collected from Kaggle. We used 2835 total images in this project for all four classes, which are divided into 2 folders (Test and Train).

Training data consists of 2010 images and Test data consists of 825 images for all the classes.

With the help of ImageDataGenerator, the training data is further divided into train (1608 images) and validation (402 images) folders. which is showcased in the below table.

Original dataset	Class type	Total no.of X-ray images	Train	validation	Test
Mendeley Data	Pneumonia	700	400	100	200
NLM,Belarus and NIAID	Tuberculosis	700	400	100	200
Covid19 Radiography Database	Covid-19	700	400	100	200
All	Normal	735	408	102	225

Table 0.3.1: Dataset Description

Pneumonia Data from Mendeley Data (University of California, San Diego) The 700 pneumonia images in the dataset contain all images of chest x-ray and its posterior from Mendeley data, which contains 5863 images divided into two categories (Pneumonia and Normal). Chest X-ray images (anterior and posterior) were selected from retrospective cohorts of paediatric patients aged one to five years old from the Guangzhou Women's and Children's Medical Centre, Guangzhou. All chest X-ray imaging was performed as part of patient's routine clinical care.

Tuberculosis For Tuberculosis, 700 images were collected from the Kaggle dataset. It consists of 700 Tuberculosis images and 2500 normal images, which is the combination of the below datasets.

1. NLM datasets: The US National Library of Medicine (NLM) publishes two lung X-ray datasets (The Montgomery dataset and the Shenzhen dataset).

2. Belarusian dataset: The Belarusian dataset was collected for drug resistance studies initiated by the National Institute of Allergy and Infectious Diseases, Ministry of Health of the Republic of Belarus.

3. NIAID TB Dataset: The NIAID TB Portal Program dataset contains approximately 3000 TB of positive CXR images from approximately 3087 cases.

Covid Data from COVID-19 Radiography Database
For Covid. I took 700 images from the COVID-19 X-ray database, which consists of 3616 COVID images. A database of chest X-rays for COVID-19, created by a team of researchers from Qatar University in Doha, Qatar and the University of Dhaka in Bangladesh, and collaborators in Pakistan and Malaysia in collaboration with doctors, was positive.

Normal
For the Normal class, 728 normal images were collected from all the above datasets.

0.3.2 Pre-processing and Augmentation

In this work, we used ImageDataGenerator, which will augment our images in realtime while our model is still training. The size of the input image is different for different CNN'S; therefore, the dataset is pre-processed[2] to resize. All images are normalized.

The image augmenting technique is important to train properly a model. When we don't have an adequate number of data to train our model, we can use this method to transform images or photos in a way that an objects can be "seen" by the network in different ways and/or under different illumination conditions. Image Augmentation works by using many random transformations like rotation,flipping, and shifting of the original image, which results in multiple transformed copies of the original image.

ImageDataGenerator[22] syntax:

"tf.keras.preprocessing.image.ImageDataGenerator ()".

0.3.3 Experiment

The experiment has been performed in two approaches. The first approach, concerns the training of a neural network from scratch.

1. Preparing Data.
2. Designing neural networks.
3. Compiling neural networks.
4. Training the Neural network.
5. Saving the model.
6. Testing the model with test images-Classification.
7. Taking performance matrix of the model.

The second approach, takes advantage of pretrained models that are fine tuned to specialize the model on the task of interest. In this case, transfer learning, the part of the pretrained network performing feature extraction are kept while only the last stages, operating the classification are actually trained.

1. Setting up the system
2. Using the same dataset used in first method
3. Loading the existing model (VGG16, ResNet50, etc. ;)
4. Performing test and train a pre-trained model using transfer learning(Image Classification).
5. Taking the performance matrix of the pretrained models.

0.3.4 Performance Matrix

The performance of different networks on the test dataset was evaluated after completing the training and validation phases and compared using performance metrics (loss, accuracy, F1 score, accuracy, specificity, recall). These are important factors in deciding whether to use that particular model for a particular task.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

$$Accuracy = \frac{(TP + TN)}{(TP + FN)(FP + TN)} \tag{1}$$

Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all the examples that truly belong to the class.

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

Precision is defined as the fraction of relevant examples (true positives) among all the examples that were predicted to belong in a certain class.

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

$$Specificity = \frac{TN}{(FP + TN)} \tag{4}$$

$$F1Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{5}$$

Here, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are used.

0.4 Results & Discussion's

In this section, we evaluated and compared the performance of six CNN model configurations (Basic CNN, VGG16, ResNet50, ResNet101, InceptionV3, and MobileNet) on chest X-ray images with multiple classes (Tuberculosis, COVID,Pneumonia, and Normal).

0.4.1 Results

0.4.1.1 Basic CNN structure and results

In our basic CNN model, we took the sequential model. The sequential model is one of the simplest to understand and implement. Its main idea is simply to arrange the keras layers in a sequential order, kind of like stacking blocks. There are very good chances that this model will be more effective and robust than other options because it performs well on continuous data, which makes it ideal for building state-of-the-art machine models. A sequential model basically means putting a layer on top of the previous one. It is important to use sequential models because the output can be predicted based on the input features and layers of the previous ones. A sequential model is an approach that uses a fixed order of operations to process a sequence or collection of data or other inputs. The use of this model can be helpful in many tasks like filtering, grouping, indexing, and searching. The structure of our model can be seen in the below figure.

```
cnn.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 220, 220, 32)      2432

max_pooling2d (MaxPooling2D  (None, 110, 110, 32)     0
)

conv2d_1 (Conv2D)            (None, 108, 108, 64)      18496

max_pooling2d_1 (MaxPooling  (None, 54, 54, 64)       0
2D)

conv2d_2 (Conv2D)            (None, 53, 53, 128)       32896

max_pooling2d_2 (MaxPooling  (None, 26, 26, 128)      0
2D)

flatten (Flatten)            (None, 86528)             0

dense (Dense)                (None, 64)                5537856

dropout (Dropout)            (None, 64)                0

dense_1 (Dense)              (None, 32)                2080

dense_2 (Dense)              (None, 4)                 132

=================================================================
Total params: 5,593,892
Trainable params: 5,593,892
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.1.1: CNN Architecture

CNN

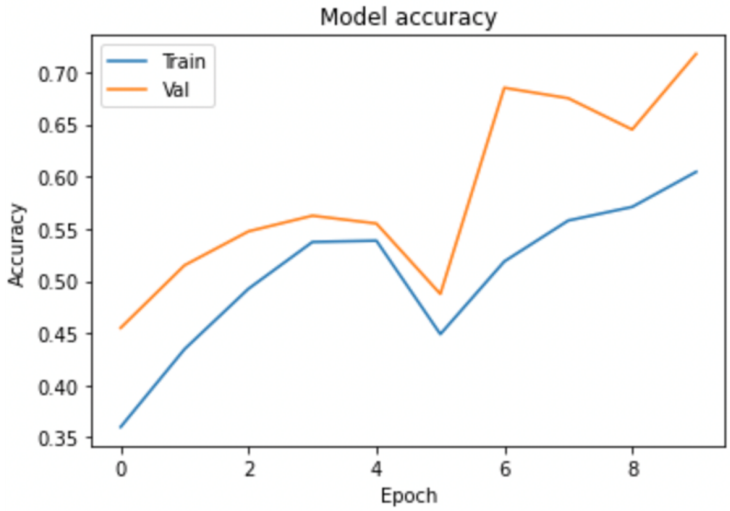


Figure 4.1.2: CNN Accuracy

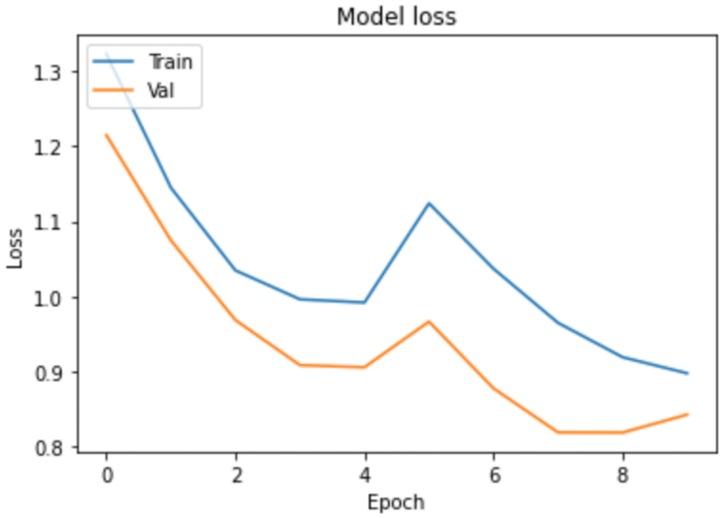


Figure 4.1.3: CNN loss

0.4.1.2 Pre-trained models and their results

All the models are imported from tensorflow.keras.application. After loading the model, we are adding our own layers to the pre-trained models, and we do compile and fit with our data to see the results. **InceptionV3**

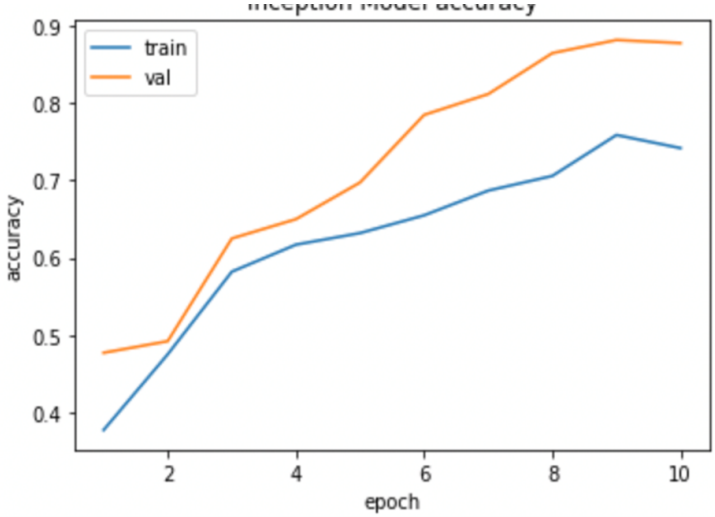


Figure 4.1.4: InceptionV3 Accuracy

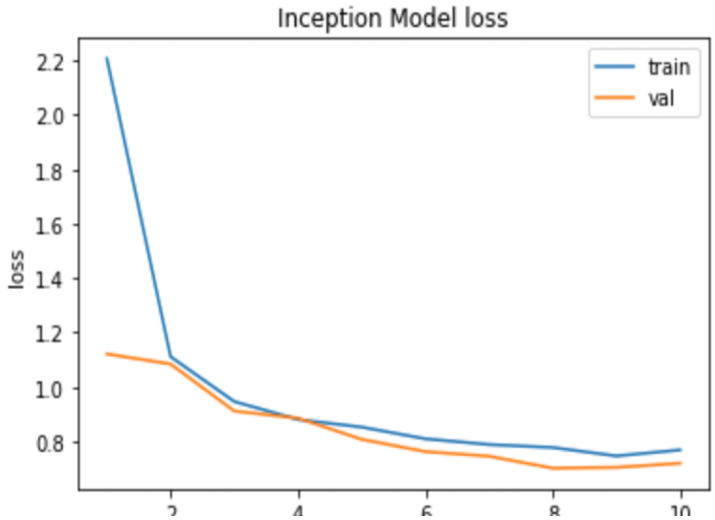


Figure 4.1.5: InceptionV3 loss

ResNet50

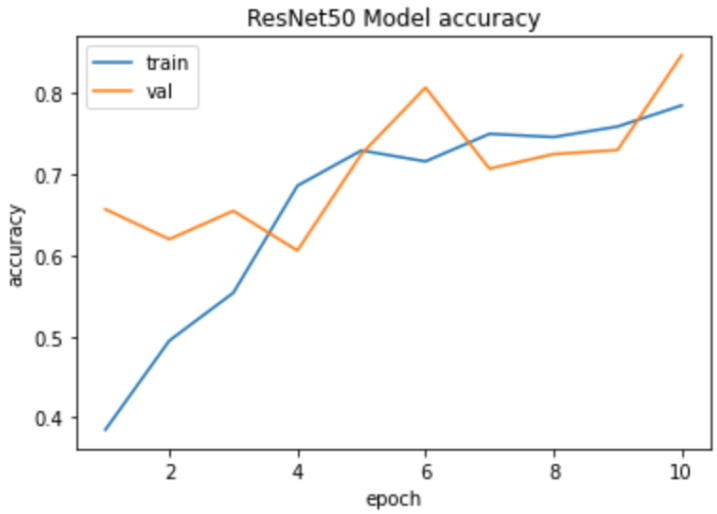


Figure 4.1.6: ResNet50 Accuracy

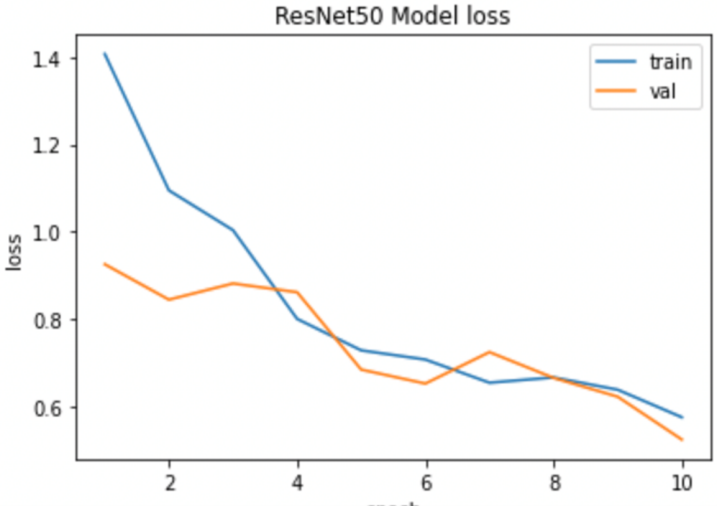


Figure 4.1.7: ResNet50 Loss

MobileNet

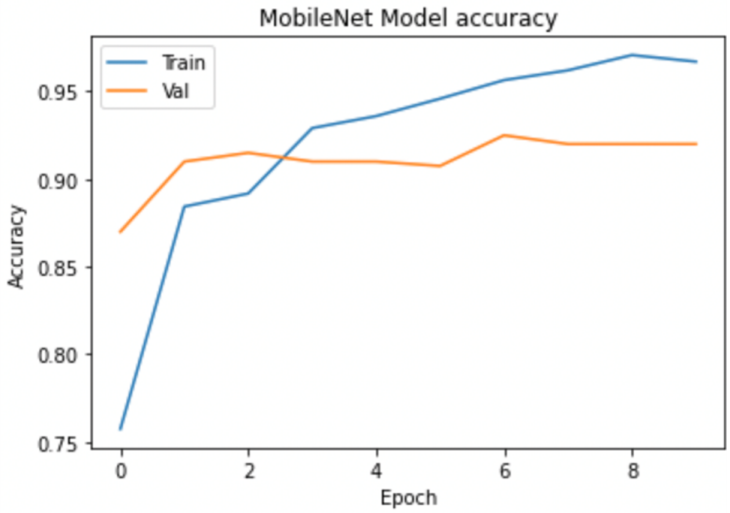


Figure 4.1.8: MobileNet Accuracy

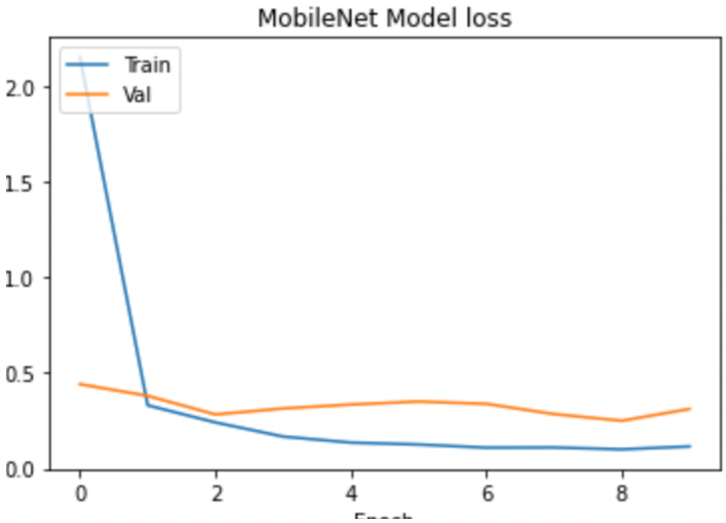


Figure 4.1.9: MobileNet Loss

ResNet101

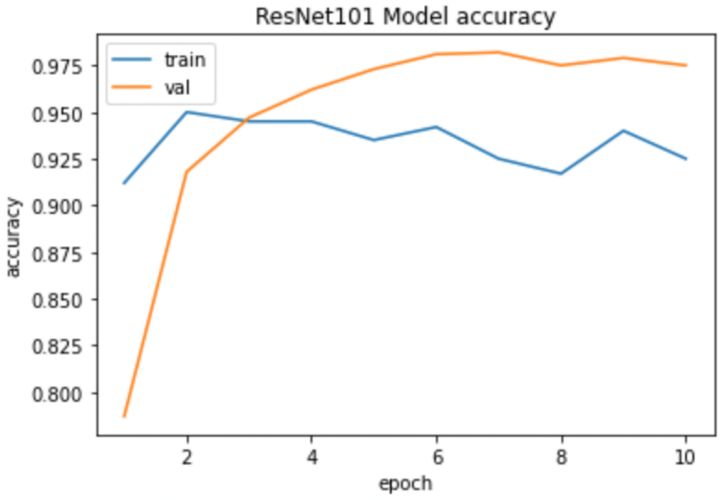


Figure 4.1.10: ResNet101 Accuracy

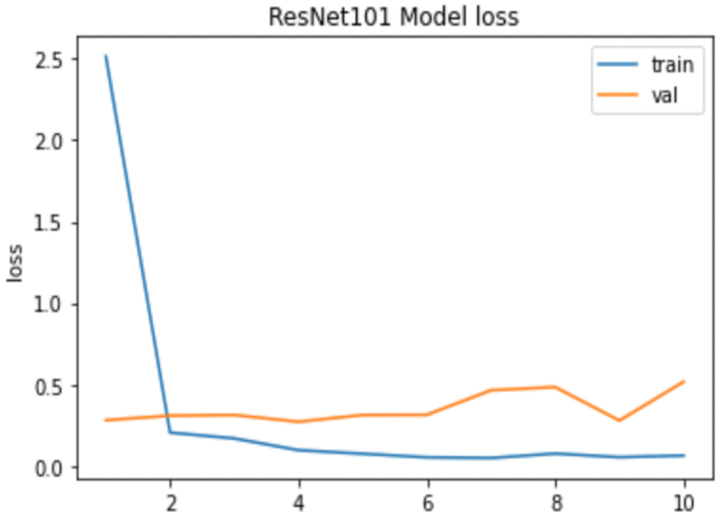


Figure 4.1.11: ResNet101 Loss

VGG16

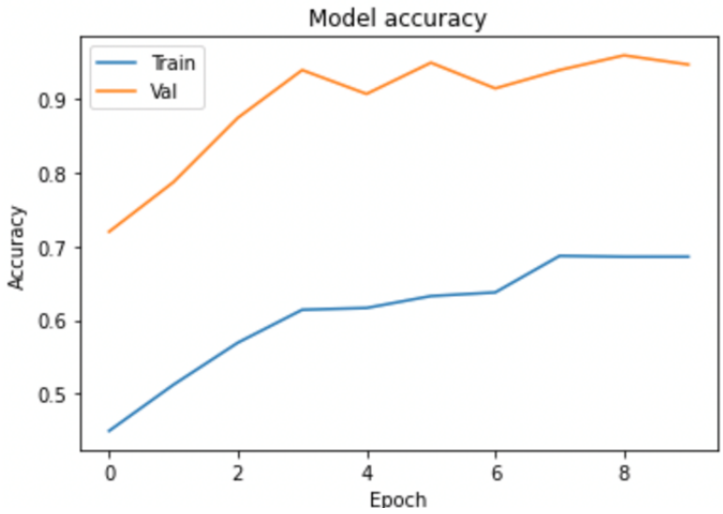


Figure 4.1.12: VGG16 Accuracy

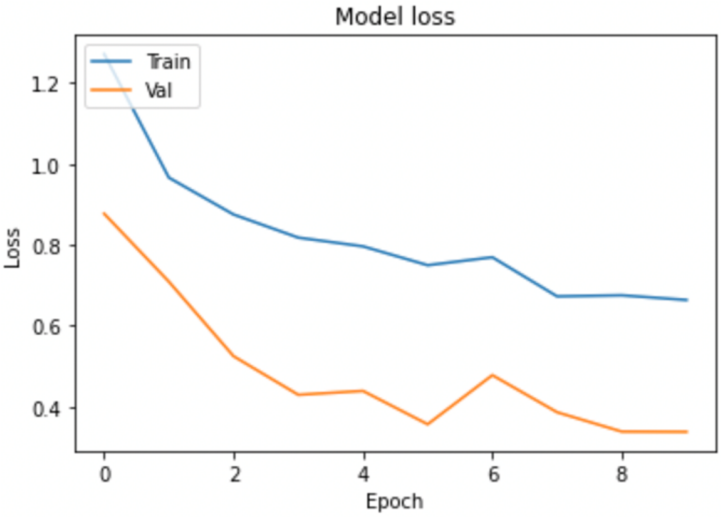


Figure 4.1.13: VGG16 Loss

Performance Matrix for all the models

Models	Accuracy	Precision	Recall	F1 Score
CNN	71.74	72.60	71.86	72.23
VGG16	94.75	89.50	89.00	89.25
ResNet50	84.75	84,75	84,72	84,73
InceptionV3	87.99	88,50	88,00	88,00
ResNet101	93.75	94,25	94,25	94,25
MobileNet	92.50	92,25	92,50	92,40

Table 0.4.1: Performance Matrix

From the above results, we can say transfer learning models give better results than the basic CNN model. For the dataset we used, VGG16 gave best result among all the above models.

As a example i am showing one model Classification report for inceptionv3 shown below it will help us to evaluate the performance of the model.

```
print(classification_report(ytest_n1,y_pred2))
```

	precision	recall	f1-score	support
Covid	0.94	0.83	0.88	99
Normal	0.86	0.92	0.89	100
Tuberculosis	0.80	0.87	0.83	100
pneumonia	0.94	0.90	0.92	101

Figure 4.1.14: InceptionV3 Classification Report

Visualization of Network

We worked with a visual explanation method called Score-CAM due to its better performance. It is based on an activation map that uses a linear combination of weights and activation maps to transfer the result to the target class. The activation map identifies the regions of the lung that contribute most to the decision. These regions are often associated with decision-making functions, like breathing.

An example, visualization using Score-CAM is shown in below Figure. Here, the lung area dominates convolution neural network decision-making. This helps in understanding how the network make decisions and give end-users assurance that the network is always making accurate decisions based on the relevant part of chest x-ray, the lungs.

Figure 4.1.15: Score-CAM heat map on chest x-ray images

Hyperparameters

These are independent of the dataset and external to the model, they are not a part of the trained model. As a result, the values are not saved and are not part of the model. The different types of hyperparameters we used in this project are:

1. Epoch.
2. Activation Function.
3. Batch Size.
4. Optimizer.
5. Loss Function.
6. Learning rate.

Epoch

For a neural network, an epoch refers to the process of completing one cycle of training using all the training data. Depending on the dataset we use, it is not a set value. Typically, we start with a large epoch value and then adjust it based on the error rate. In our modelling we used 10 epochs, because that gave sufficiently accurate results and prevents overfitting.

Activation Function

Without an activation function, a neural network is nothing more than a linear regression model. Therefore, we introduce an activation function that non-linearly transforms the neuron's input.

Various activation functions include Binary set Function, Linear Function, Sigmoid, Tanh, ReLu, exponential Linear Unit, SoftMax etc., ReLu and SoftMax were utilized in this project.

ReLu: -

Rectified Linear Unit, or ReLu, is a function that I used in our study since it has advantages like not activating all the neurons at once and being computationally more effective than other functions because just a minimal number of neurons are active.

SoftMax: -

The multi-sigmoidal combination known as SoftMax. It can be used for multiclass classification problems. The probability that a data point belongs to each class is returned.

SoftMax was used as the activation function for the last layers of the model because categorical CrossEntropy is what I am using as a loss Function.

Batch Size

It simply refers to the number of samples used in one iteration, and it has an impact on both the overall training time and the training time per epoch. Given that the dataset is small in this work, I set the batch size to 32.

Optimizer

The optimizer will try to reduce the loss function by modifying the model's parameters in response to the output. several optimizers, including Gradient Descent, Stochastic Gradient Descent, RMSprop, Adam, and Aagrad etc., I used Adam in our work because it is the combination of both Gradient Descent and RMSprop. It requires less memory.

Loss Function

There are different types of loss functions that anticipate the neural network's error. Those are Mean square error, categorical and binary crossentropies, etc. In our work, we used categorical Crossentropy because it is the optimum loss function when using multi-class classification.

Learning rate

This is a key parameter that describes how quickly the model can be problem-adapted. I used the default learning rate of 0.001 in our work.

0.4.2 Discussion

The CNN is an in-house prepared architecture with only 3 convolution layers and 3 pooling layers. Thus, this acts as a base model for the transfer learning techniques. The F1 score is roughly 0.72 (72%), giving a clear picture that there are many improvements to be made.

VGG-16 is a standard model that is still used and has a good reputation for its reproducibility and architecture. It has achieved an accuracy of 94% and the F1 score is 90%, giving pretty good results. The VGG-16 has shown higher validation accuracy than the training data from the start, and the trend continued till the last epoch.

ResNet50 is a much more complex model than VGG and takes a higher running time. It has improved the accuracy of training rapidly, but the results after 10 epochs of ResNet50 vs VGG-16 showed that a relatively simple model has performed well.

InceptionV3 has an efficient utilization of resources with little increase in computation load. The model showed a consistent increase in accuracy with epochs and gave a decent accuracy and F1 score of 88%.

ResNet101 is a huge model with a depth of 101 layers, and hence it is expected to perform better. The results certainly showed the same, giving 92% accuracy and a 94% F1 score. Thus, this can be relied upon in practical applications. The performance has increased sharply with just 2 epochs', showing the complexity that can be handled by the model.

MobileNet is very similar to VGG-16, but 30 times smaller and 10 times faster.From the results, we can observe that it has performed better than VGG-16.

As we used different models, each having different architecture, it would be insightful to observe the amount of training needed to get the best results and see which architecture gave best results with less effort.

Starting with the in house model of CNN that required roughly 5.6 million weights to be

trained but gave just average level of accuracy, precision and recall. VGG-16 and InceptionV3 require just about 1.6 million weights to be trained, thus decreasing the calculation power to one fourth of what is used by the conventional CNN but giving good results. Models such as ResNet50 and ResNet101 are computationally expensive, requiring about 6 million weights to be trained. Mobile Net has a medium level of weights to be trained going up to 3.2 million.Overall the VGG-16 is highly efficient computationally and is next to ResNet101 in terms of F1 score.

All these different models are compared in terms of accuracy, F1 score etc., but in order to see which models are close to the solution can be found by observing the accuracy vs epochs graphs. Most of the models have shown continuous increase of accuracy with only exceptions being ResNet101 and MobileNet.

MobileNet showed sharp increase of accuracy for the first 3 epochs and the improvement in model is minimal after the 3rd epoch showing that the most of the model fitting happened in the first 3 epochs. In contrast to the above the ResNet101 has shown consistent behaviour from the start and the accuracy was ranged. There is no significant improvement in the model accuracy. This shows that the ResNet101 is a reasonably good model for applications of similar kind.

0.5 Conclusion and Future works

0.5.0.1 Conclusion

The aim of the thesis is to create a good CNN (Convolution Neural Network) which can be trained and tested to identify which x-ray image belongs to which type of disease. We can find the accuracy of CNN. After that, we will use the transfer learning method on the same data to see which model gives the better performance.

To this end, we leveraged and comprehensively evaluated three important unexplored factors related to Deep Convolutional Neural Networks (CNN) architecture, dataset properties, and transfer learning.

Evaluate CNN performance using three different computer-aided diagnostic applications: tuberculosis, Covid, and lung disease classification. The empirical evaluation, CNN performance analysis, and final findings can be generalized to the design of high-performance CAD systems for other medical imaging tasks.

In continuation of the work already done, we strongly suggest that a further study can be performed on segmentation. These studies could not be performed due to the limitation of hardware availability, resources, and time. One can expect to see much better results using segmentation techniques.

0.5.0.2 Future works

The project had its own constraints even if it produced satisfactory results.Among these restrictions, the main one is the need for appropriate masked datasets for segmentation in all four classes, and the other is data scarcity.
I advise anyone interested in this project to expand it and include the following benefits:

1. Get the best and most comprehensive dataset for all four classes.
2. Obtain the masked data for each of the four classes in order to do segmentation to improve the classification results.

Appendix

Image Analysis

Image analysis is a branch of computer science in which computers process human-created visual media to extract data. Image processing is one aspect of the field and involves analysing changes in an image over time or across views, such as determining if a change has occurred or whether coloration has moved from one part of the image to another. Image analysis may also include tasks such as finding shapes, detecting edges, and removing noise. Other tasks include counting objects or calculating statistics for texture analysis or image quality. Image analysis is a task where computers break up an image, like a painting, and analyse each part. By studying images with higher resolution, you can learn how objects are made, what types of shapes they are, how many there are, and how much light is needed for them to be seen. The most common types of image analysis are filtering and classification. Filtering is used for adjusting the amount of light present in an image, while classification involves classifying images based on their content.

Image analysis procedure

Description

Describe what you see in as much details as possible. List the information about the images.

Identification

Record basic information about the data.

Interpretation

Analyse the images information, based on what you know about this image.

Evaluation

Is this image effective? Does it successfully communicate its intended information?

Digital Image Analysis

A method in which an image or other type of data is changed into a series of dots or numbers so that it can be viewed and studied on a computer.

Bibliography

[1] Hamed Behzadi-Khormouji, Habib Rostami, and Sana Salehi. Pmid: Deep learning, reusable and problem-based architectures for detection of consolidation on chest x-ray images. In *Comput Methods Programs Biomed*, 2019.

[2] Aniruddha Bhandari. Analytic:image augmentation on the fly using keras image-datagenerator. In *Analytic vidhya*, 2020.

[3] Abhir Bhandary and Gnanth Prabhu. volume: Deep-learning framework to detect lung abnormality – a study with chest x-ray and lung ct scan images. In *Pattern Recognition letters*, 2020.

[4] Subrato Bharati and Prajoy Podder. june: Lung cancer recognition and prediction according to random forest ensemble and rusboost algorithm using lidc data. In *Pattern Recognition letters*, 2019.

[5] Vikash Chouhan CID and Sanjay. Appl:a novel transfer learning based approach for pneumonia detection in chest x-ray images. In *Signal Processing and Machine Learning for Biomedical Data*, 2019.

[6] Kaiming He, Georgia Gkioxari, and Piotr Dollár. Arixv: Mask r-cnn. In *Computer Vision and Pattern Recognition*, 2017.

[7] Jeremy Irvin, Pranav Rajpurkar, and Michael Ko. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *AAAI Press*, 2019.

[8] kaggle. Dataset. https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset, 2018.

[9] kaggle. Dataset. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia, 2019.

[10] kaggle. Dataset. https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database, 2021.

[11] K Kallianos and J Mongan. Pmid: How far have we come? artificial intelligence for chest radiograph interpretation. In *National Library of medicine*, 2019.

[12] C-H Liang and Y-C Liu. Pmid:identifying pulmonary nodules or masses on chest radiography using deep learning: external validation and strategies to improve clinical practice. In *Clin Radiol*, 2019.

[13] Nasrullah, Jun Sang, and Mohammad S. Alam. :automated detection and classification for early stage lung cancer on ct images using deep learning. In *Pattern Recognition and Tracking XXX*, 2019.

[14] S. Rajaraman and S. K. Antani. doi:modality-specific deep learning model ensembles toward improving tb detection in chest radiographs. In *IEEE*, 2020.

[15] Ebrahimzadeh S and Dawit H Islam N. cochrance: How accurate is chest imaging for diagnosing covid-19? In *cochrance library*, 2022.

[16] Sandeep Salvi and G Anil Kumarand. Lancet: The burden of chronic respiratory diseases and their heterogeneity across the states of india. In *The Global Burden of Disease Study 1990–2016*, 2018.

[17] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Pmid: Convolutional networks for semantic segmentation. In *IEEE Trans Pattern Anal Mach Intell. 2017*, 2017.

[18] Madhuragauri Shevade and Komalkirti Apte. congress: What are the most common respiratory diseases encountered in clinical practice? In *European Respiratory Journal 2015*, 2015.

[19] Ali Taghizadieh and Alireza Ala. Pmc: Diagnostic accuracy of chest x-ray and ultrasonography in detection of community acquired pneumonia; a brief report. In *National Library of medicine*, 2015.

[20] Takeshi takaki and seiichi murakami. doi:calculating the target exposure index using a deep convolutional neural network and a rule base. In *Physica Medica*, 2020.

[21] Singh V and Sharma BB. Pmcid: Respiratory disease burden in india: Indian chest society sword survey. In *National Library of medicine*, 2018.

[22] wiki. Imagedatagenerator. https://en.wikipedia.org/wiki/Data_pre-processing, 2015.