DETECTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP CONVOLUTION NEURAL NETWORKS

A Project report submitted in partial fulfillment of the requirements forthe award of the degree of

BACHELOR OF TECHNOLOGY IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

M.Sai Pavithra(318126512155) D.Vijay Kumar(318126512135)

P.Vinay Mouli(318126512168) K.Tanoj Naidu(318126512148)

Under the guidance of
Mr.V.Vijay Kumar Raju
Assistant professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'B+' Grade*)
Sangivalasa, bheemilimandal,
visakhapatnam dist.(A.P)

2021-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)
(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'B+' Grade)
Sangivalasa, Bheemilimandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

***This is to certify that the project report entitled* "DETECTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP CONVOLUTION NEURAL NETWORKS" submitted by M.Sai Pavithra(318126512155), K.Tanoj Naidu(318126512148), D.Vijay Kumar (318126512135), P.Vinay Mouli(318126512168) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.**

Project Guide	**Head of the Department**
Mr. V.VijayKumarRaju	**Dr. V.Rajyalakshmi**
Assistant Professor	Professor & HOD
Department of E.C.E	Department Of E.C.E
ANITS	ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Scien
Sangivalasa

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr. V.Vijay Kumar Raju** Designation, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank all **non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

M. Sai Pavithra (318126512155),

P. Vinay Mouli (318126512168),

D. Vijay Kumar (318126512135),

K. Tanoj Naidu (318126512148).

ABSTRACT

COVID-19, regarded as the deadliest virus of the 21st century, has claimed the lives of millions of people around the globe in less than two years. Since the virus initially affects the lungs of patients, X-ray imaging of the chest is helpful for effective diagnosis. Any method for automatic, reliable, and accurate screening of COVID-19 infection would be beneficial for rapid detection and reducing medical or healthcare professional exposure to the virus. In the past, Convolutional Neural Networks (CNNs) proved to be quite successful in the classification of medical images. In this study, an automatic deep learning classification method for detecting COVID-19 from chest X-ray images is suggested using a CNN. A dataset consisting of 3616 COVID-19 chest X-ray images and 10,192 healthy chest X-ray images was used. The original data were then augmented to increase the data sample to 26,000 COVID-19 and 26,000 healthy X-ray images. The dataset was enhanced using histogram equalization, spectrum, grays, cyan and normalized with NCLAHE before being applied to CNN models. Initially using the dataset, the symptoms of COVID-19 were detected by employing eleven existing CNN models; VGG16, VGG19, MobileNetV2, InceptionV3, NFNet, ResNet50, ResNet101, DenseNet, EfficientNetB7, AlexNet, and GoogLeNet. From the models, MobileNetV2 was selected for further modification to obtain a higher accuracy of COVID-19 detection. Performance evaluation of the models was demonstrated using a confusion matrix. It was observed that the modified MobileNetV2 model proposed in the study gave the highest accuracy of 98% in classifying COVID-19 and healthy chest X-rays among all the implemented CNN models. The second-best performance was achieved from the pre-trained MobileNetV2 with an accuracy of 97%, followed by VGG19 and ResNet101 with 95% accuracy for both the models. The study compares the compilation time of the models. The proposed model required the least compilation time with 2 h, 50 min and 21 s. Finally, the Wilcoxon signed-rank test was performed to test the statistical significance. The results suggest that the proposed method can efficiently identify the symptoms of infection from chest X-ray images better than existing methods.

Keywords: COVID-19, chest X-ray image, CNN, Mobilenetv2, modified MobileNetV2, performance

CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE NO.

CHAPTER 1 INTRODUCTION

1.INTRODUCTION

The COVID-19 outbreak has risen to the status of one of the most severe public health issues of the last several years. The virus spreads rapidly: the reproduction number of COVID-19 varied from 2.24 to 3.58 during the initial months of the pandemic, indicating that each infected individual on average transmitted the disease to two or more others. Consequently, the number of COVID-19 infections grew up from a few hundred cases (most of them in China) in January 2020 to more than 43 million cases in November 2020 disseminated across the world .

It's believed that the coronavirus that causes COVID-19 is the same one that causes SARS-COV2 and MERS (MERS). COVID-19 has a wide range of symptoms that appear after an average incubation period of 5.2 days. Fever, dry cough, and tiredness are common symptoms, while others include headache, hemoptysis, diarrhea, dyspnea, and lymphopenia. In December 2019, the first human was infected with coronavirus (SARS-COV-2), and it is mostly transmitted by droplets produced when infected people talk, cough, or sneeze. Because the droplets are too heavy to travel far, they can only be propagated via direct contact . Recent research estimates that the COVID-19 can survive up to 3 h in the air, 4 h on copper, and 72 h on plastic and stainless steel, but the precise durations are yet unknown. However, the general health research community is yet to agree on answers to these issues, which are still under study. An infection with COVID-19 affects the tissues of the lungs. Some affected people may not notice any symptoms in the early stages, while fever and cough were the most common core symptoms for the majority of patients. Other side effects include muscle pains, a sore throat, and a headache. Cough medicine, pain killers, fever reducers, and antibiotics are provided to patients based on their symptoms, not the disease organism. The patient must be hospitalized and treated in an Intensive Care Unit (ICU), which may include the use of a ventilator to help the patient breathe . As a result of its severity and ease of transmission, COVID-19 has spread rapidly across the world. The greater effect on health care departments is primarily due to the number of individuals impacted day by day, as they need to give mechanical ventilators to critically ill patients admitted to ICU. As a result, the number of ICU beds must be significantly expanded. In the aforementioned scenario, early diagnosis is critical in ensuring patients receive appropriate treatment, while reducing the load on the healthcare system. COVID-19 is still a deadly disease due to the absence of early diagnostic techniques around the world, as well as having medical preconditions such as cancer, chronic liver, lung, and kidney diseases, and diabetes. Though RT-PCR diagnosis techniques are available in most parts of the world, under-developed countries still cannot afford to test all their people promptly . In 2020–2021, this disease claimed the lives of millions of people across the earth. Vaccines for COVID-19 are now being developed in a number of countries. Vaccines produced by Pfizer, AstraZeneca, Moderna, Serum Institute of India Pvt. Ltd., Janssen, Sinopharm, Sinovac are among the vaccines that have been approved by WHO for administration . Such approved vaccines have substantially reduced the deadliness of the disease.

Artificial intelligence has shown its efficiency and excellent performance in automated image categorization issues via various machine learning methods and is currently being used to automate the diagnosis of various diseases. Furthermore, machine learning refers to models that can learn and make decisions based on vast quantities of data samples. Artificial intelligence accomplishes activities that need human intellect, such as voice recognition, translation, visual perception, and more, by performing calculations and predictions depending on the incoming data . Deep learning is a collection of machine learning techniques that primarily concentrate on the automated extraction and categorization of image features, and has shown tremendous promise in a variety of applications, particularly in health care . Scientists from around the world are trying to develop technologies that can assist radiologists/doctors in their diagnoses. To identify the optimal network for the area of radiology and medical image processing, a variety of AI methods have been used so far . Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are two prominent deep-learning-based networks that have been extensively utilized in medical research fields like speech recognition, computer vision, and Natural Language Processing (NLP), and have often produced commendable results. The classification, localization, and segmentation of images using CNN have shown impressive results in medical image processing .

This is a significant step forward in the identification of COVID-19 and other forms of lung inflammation due to artificial intelligence (AI). The WHO recommends using RT-PCR as the main diagnostic method for COVID-19 detection. Chest X-rays or chest CT scans are also widely used, but should only be preferred when the RT-PCR test is not available in a timely manner . In most cases, it is seen that the nasal swap test results vary if done at different times of the day. For instance, the possibility of getting a positive result is higher if the test is done in the morning compared to a test done in the evening . Furthermore, the detection of COVID-19 positive patients was significantly delayed due to a high number of false-negative results . The RT-PCR has a low success rate of 70% and a sensitivity of 60–70% . Moreover, the false-negative cases of RT-PCR show positive results on chest X-ray imaging. Due to its widespread availability, X-ray imaging has played a significant role in many medical and epidemiological situations . Because of its operating speed, low cost, and ease of use for radiologists, chest X-ray seems to be promising for emergency situations and therapy. A prior study, however, found significant discrepancies in chest X-ray images obtained from patients who had COVID-19 .

The study proposed an intelligent deep learning architecture such as a modified MobileNetV2 with RMSprop optimizer to detect COVID-19 disease. For the study, 13,808 X-ray image data were collected from the dataset and augmented to a larger dataset of 52,000 chest X-ray images. Image processing methods such as enhancement, normalization, and data augmentation were used to help the proposed model not only to avoid overfitting but also to demonstrate the highest accuracy. The performance accuracy and compilation time of the newly proposed methods are compared to those of eleven existing CNN models. Finally, to access the statistical significance, a Wilcoxon signed-rank test was performed.

1.1 Project Overview

The main objective of this paper is to classify COVID-19-infected patients from chest CT images. A novel deep learning model is designed by using multi-objective differential evolution (MODE) and convolutional neural networks (CNN) for classification of human beings based upon whether they are affected from COVID-19 or not. A multi-objective fitness function is designed to classify COVID-19-infected patients by considering sensitivity and specificity. The hyperparameters of CNN are optimized by using the MODE algorithm. The proposed model is trained by considering the chest CT images of COVID-19 patients. The comparisons between the proposed MODE-based CNN with the competitive models such as convolutional neural networks (CNN), adaptive neuro-fuzzy inference systems (ANFIS), and artificial neural networks (ANN) are also drawn by considering the well-known classification metrics.

1.2 Project Outline

A quick and accurate diagnosis is essential during a pandemic, such as COVID-19. It leads to better outcomes for patients, and can relieve pressure on health care systems struggling to deal with an increasing rate of infection.The current preferred method for diagnosis of COVID-19 is polymerase chain reaction (PCR). However, some of the hardest hit areas are unable to source enough kits to meet demand and many countries are unable to process tests due to inadequate lab facilities.Deep learning models, a form of artificial intelligence (AI), are being widely researched and adopted for detection and diagnosis across a variety of diseases.In this instance, deep learning techniques could be used to identify infected patients using chest X-ray images – which are widely available worldwide. This method could be used in areas where the PCR diagnostic method is not currently feasible.The use of deep learning to analyse the X-rays could greatly reduce the length of time taken to diagnose patients – with an AI model processing up to 200 images in the average time taken for a radiologist to analyse one.

CHAPTER 2 IMAGE PROCESSING

2.IMAGE PROCESSING

2.1 Introduction

Before we jump into image processing, we need to first understand what exactly constitutes an image. An image is represented by its dimensions (height and width) based on the number of pixels. For example, if the dimensions of an image are 500 x 400 (width x height), the total number of pixels in the image is 200000.

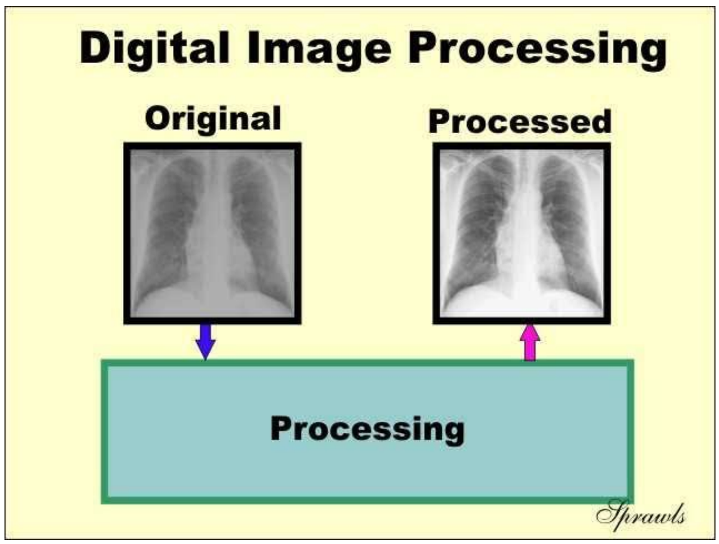


Fig 2.1 Digital image processing

This pixel is a point on the image that takes on a specific shade, opacity or color. It is usually represented in one of the following:

- Grayscale - A pixel is an integer with a value between 0 to 255 (0 is completely black and 255 is completely white).
- RGB - A pixel is made up of 3 integers between 0 to 255 (the integers represent the intensity of red, green, and blue).
- RGBA - It is an extension of RGB with an added alpha field, which represents the opacity of the image.

Image processing requires fixed sequences of operations that are performed at each pixel of an image. The image processor performs the first sequence of operations on the image, pixel by pixel. Once this is fully done, it will begin to perform the second operation, and so on. The output value of these operations can be computed at any pixel of the image.

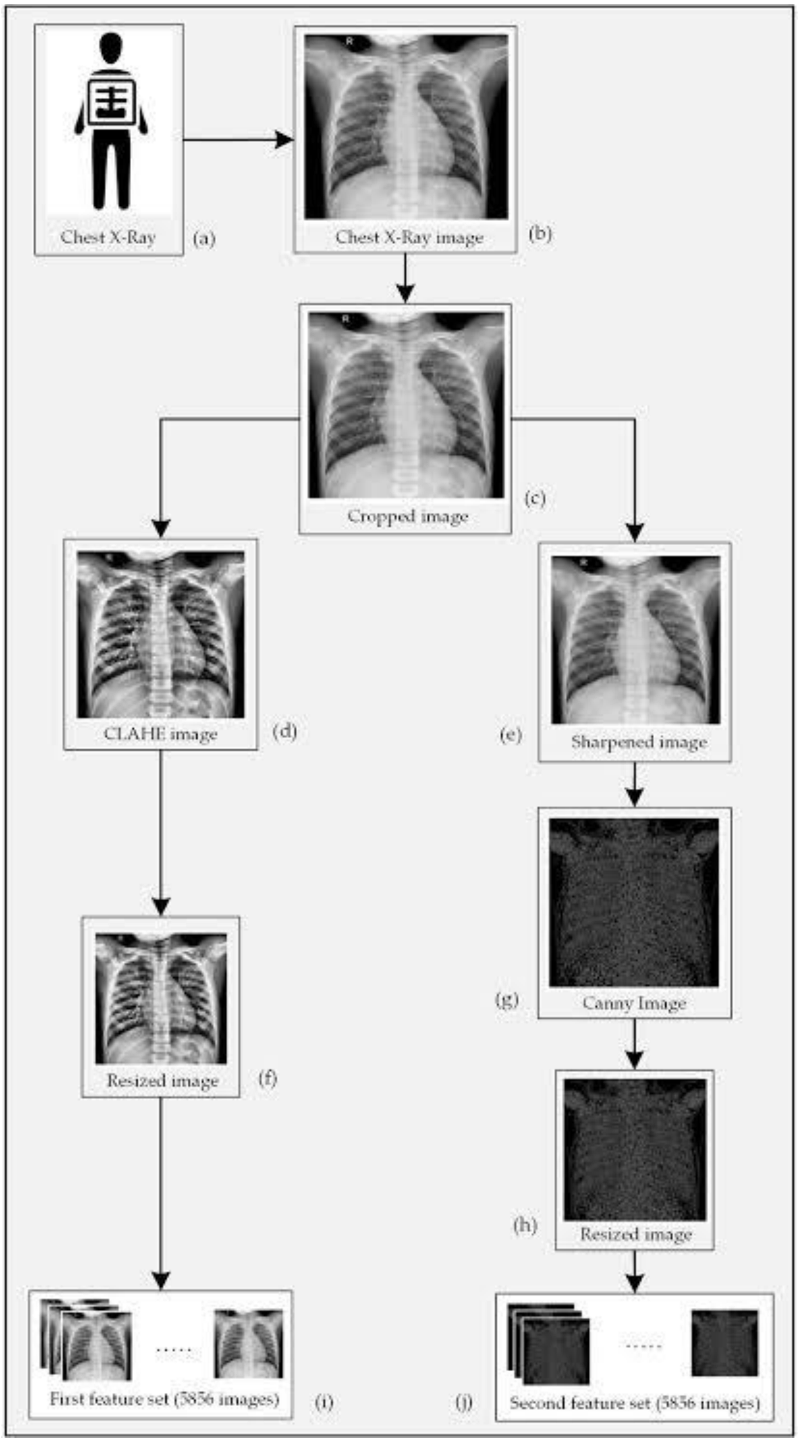


Fig 2.2 classification of image

2.2 Types of image processing

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.

There are five main types of image processing:

Visualization - Find objects that are not visible in the image

Recognition - Distinguish or detect objects in the image

Sharpening and restoration - Create an enhanced image from the original image

Pattern recognition - Measure the various patterns around the objects in the image

Retrieval - Browse and search images from a large database of digital images that are similar to the original image.

2.3 Fundamental image processing

Image Acquisition: Image acquisition is the first step in image processing. This step is also known as preprocessing in image processing. It involves retrieving the image from a source, usually a hardware-based source.

Image Enhancement: Image enhancement is the process of bringing out and highlighting certain features of interest in an image that has been obscured. This can involve changing the brightness, contrast, etc.

Image Restoration: Image restoration is the process of improving the appearance of an image. However, unlike image enhancement, image restoration is done using certain mathematical or probabilistic models.

Color Image Processing: Color image processing includes a number of color modeling techniques in a digital domain. This step has gained prominence due to the significant use of digital images over the internet.

Wavelets and Multiresolution Processing: Wavelets are used to represent images in various degrees of resolution. The images are subdivided into wavelets or smaller regions for data compression and for pyramidal representation.

Compression: Compression is a process used to reduce the storage required to save an image or the bandwidth required to transmit it. This is done particularly when the image is for use on the Internet.

Morphological Processing: Morphological processing is a set of processing operations for morphing images based on their shapes.

Segmentation: Segmentation is one of the most difficult steps of image processing. It involves partitioning an image into its constituent parts or objects.

Representation and Description: After an image is segmented into regions in the segmentation process, each region is represented and described in a form suitable for further computer processing. Representation deals with the image’s characteristics and regional properties. Description deals with extracting quantitative information that helps differentiate one class of objects from the other.

Recognition: Recognition assigns a label to an object based on its description.

2.4 Application of image processing

Medical Image Retrieval

Image processing has been extensively used in medical research and has enabled more efficient and accurate treatment plans. For example, it can be used for the early detection of breast cancer using a sophisticated nodule detection algorithm in breast scans. Since medical usage calls for highly trained image processors, these applications require significant implementation and evaluation before they can be accepted for use.

Traffic Sensing Technologies

In the case of traffic sensors, we use a video image processing system or VIPS. This consists of a) an image capturing system b) a telecommunication system and c) an image processing system. When capturing video, a VIPS has several detection zones which output an "on" signal whenever a vehicle enters the zone, and then output an "off" signal whenever the vehicle exits the detection zone. These detection zones can be set up for multiple lanes and can be used to sense the traffic in a particular station.Besides this, it can auto record the license plate of the vehicle, distinguish the type of vehicle, monitor the speed of the driver on the highway and lots more.

Image Reconstruction

Image processing can be used to recover and fill in the missing or corrupt parts of an image. This involves using image processing systems that have been trained extensively with existing photo datasets to create newer versions of old and damaged photos.

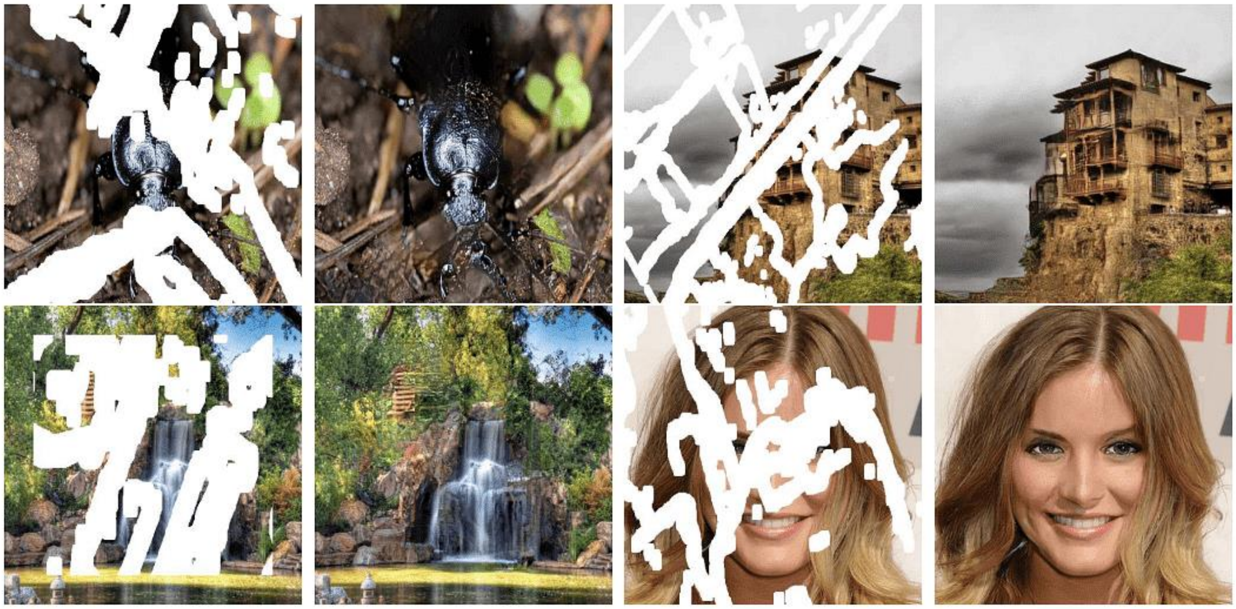


Fig 2.3 Reconstructing damaged images using image processing

Face Detection

One of the most common applications of image processing that we use today is face detection. It follows deep learning algorithms where the machine is first trained with the specific features of human faces, such as the shape of the face, the distance between the eyes, etc. After teaching the machine these human face features, it will start to accept all objects in an image that resemble a human face. Face detection is a vital tool used in security, biometrics and even filters available on most social media apps these days.

CHAPTER 3 COMPUTER VISION

3.COMPUTER VISION

3.1 Introduction

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive – and the market is continuing to grow. It is expected to reach USD 48.6 billion by 2022.

3.2 Working of computer vision

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.

Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN).

Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will "look" at the data and teach itself to tell one image from another. Algorithms enable the machine to learn by itself, rather than someone programming it to recognize an image.

A CNN helps a machine learning or deep learning model "look" by breaking images down into pixels that are given tags or labels. It uses the labels to perform convolutions (a mathematical operation on two functions to produce a third function) and makes predictions about what it is "seeing." The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions start to come true. It is then recognizing or seeing images in a way similar to humans.

Much like a human making out an image at a distance, a CNN first discerns hard edges and simple shapes, then fills in information as it runs iterations of its predictions. A CNN is used to understand single images. A recurrent neural network (RNN) is used in a similar way for video applications to help computers understand how pictures in a series of frames are related to one another.

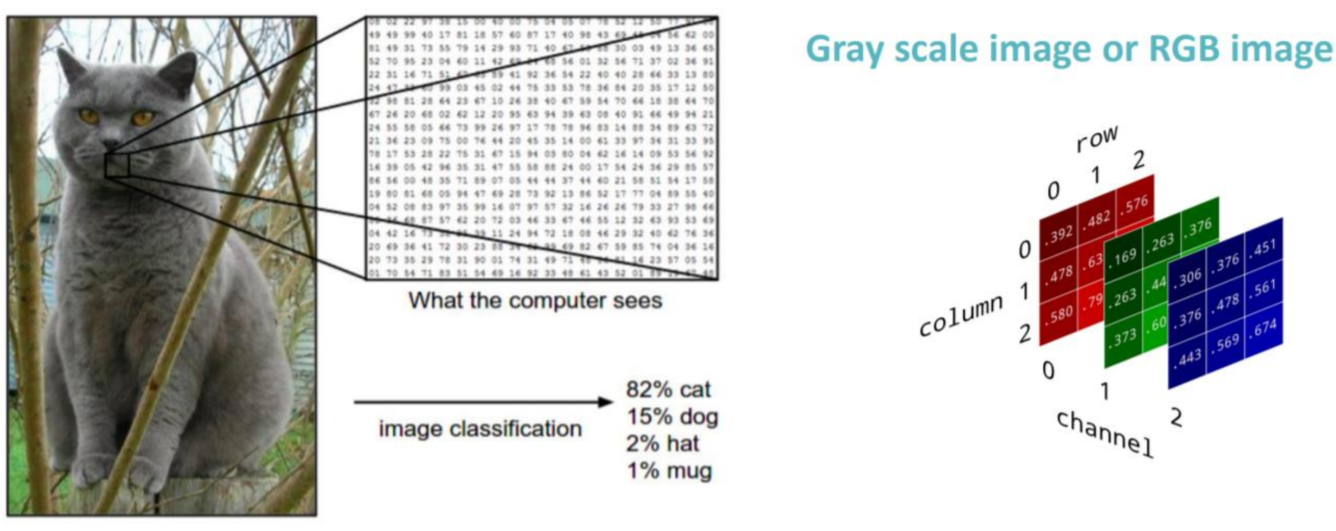


Fig 3.1 Computer vision

3.3 Application of computer vision

There is a lot of research being done in the computer vision field, but it's not just research. Real-world applications demonstrate how important computer vision is to endeavors in business, entertainment, transportation, healthcare and everyday life. A key driver for the growth of these applications is the flood of visual information flowing from smartphones, security systems, traffic cameras and other visually instrumented devices. This data could play a major role in operations across industries, but today goes unused. The information creates a test bed to train computer vision applications and a launchpad for them to become part of a range of human activities:

- IBM used computer vision to create My Moments for the 2018 Masters golf tournament. IBM Watson watched hundreds of hours of Masters footage and could identify the sights (and sounds) of significant shots. It curated these key moments and delivered them to fans as personalized highlight reels.
- Google Translate lets users point a smartphone camera at a sign in another language and almost immediately obtain a translation of the sign in their preferred language.(6)
- The development of self-driving vehicles relies on computer vision to make sense of the visual input from a car's cameras and other sensors. It's essential to identify other cars, traffic signs, lane markers, pedestrians, bicycles and all of the other visual information encountered on the road.
- IBM is applying computer vision technology with partners like Verizon to bring intelligent AI to the edge, and to help automotive manufacturers identify quality defects before a vehicle leaves the factory.

CHAPTER 4 CONVOLUTION NEURAL NETWORK

4. CONVOLUTION NEURAL NETWORK

4.1 Introduction

Convolutional Neural Networks are a special type of feed-forward artificial neural network in which the connectivity pattern between its neuron is inspired by the visual cortex.

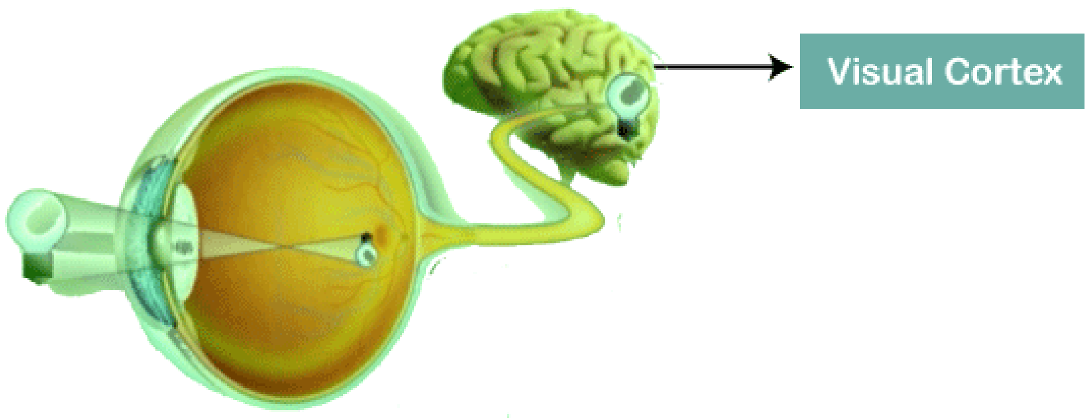


Fig 4.1 Visual Cortex

The visual cortex encompasses a small region of cells that are region sensitive to visual fields. In case some certain orientation edges are present then only some individual neuronal cells get fired inside the brain such as some neurons responds as and when they get exposed to the vertical edges, however some responds when they are shown to horizontal or diagonal edges, which is nothing but the motivation behind Convolutional Neural Networks.

The Convolutional Neural Networks, which are also called as covnets, are nothing but neural networks, sharing their parameters. Suppose that there is an image, which is embodied as a cuboid, such that it encompasses length, width, and height. Here the dimensions of the image are represented by the Red, Green, and Blue channels, as shown in the image given below.

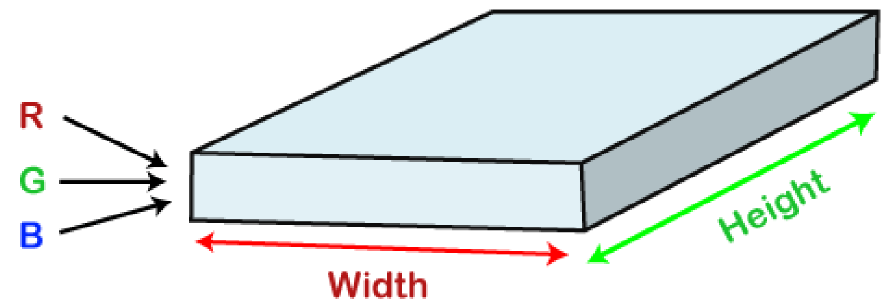


Fig 4.2 Cuboid

Now assume that we have taken a small patch of the same image, followed by running a small neural network on it, having k number of outputs, which is represented in a vertical manner. Now when we slide our small neural network all over the image, it will result in another image constituting different width, height as well as depth. We will notice that rather than having R, G, B channels, we have come across some more channels that, too, with less width and height, which is actually the concept of Convolution. In case, if we accomplished in having similar patch size as that of the image, then it would have been a regular neural network. We have some wights due to this small patch.

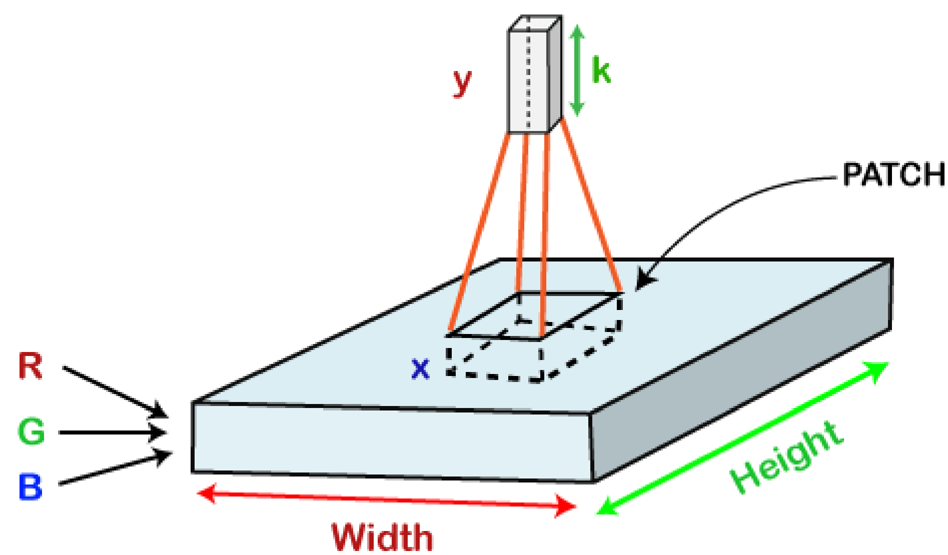


Fig 4.3 Cuboid

Mathematically it could be understood as follows;

- The Convolutional layers encompass a set of learnable filters, such that each filter embraces small width, height as well as depth as that of the provided input volume (if the image is the input layer then probably it would be 3).
- Suppose that we want to run the convolution over the image that comprises of 34x34x3 dimension, such that the size of a filter can be axax3. Here a can be any of the above 3, 5, 7, etc. It must be small in comparison to the dimension of the image.
- Each filter gets slide all over the input volume during the forward pass. It slides step by step, calling each individual step as a stride that encompasses a value of 2 or 3 or 4 for higher-dimensional images, followed by calculating a dot product in between filter's weights and patch from input volume.
- It will result in 2-Dimensional output for each filter as and when we slide our filters followed by stacking them together so as to achieve an output volume to have a similar depth value as that of the number of filters. And then, the network will learn all the filters.

4.2 Working of CNN

Generally, a Convolutional Neural Network has three layers, which are as follows;

- **Input:** If the image consists of 32 widths, 32 height encompassing three R, G, B channels, then it will hold the raw pixel([32x32x3]) values of an image.
- **Convolution:** It computes the output of those neurons, which are associated with input's local regions, such that each neuron will calculate a dot product in between weights and a small region to which they are actually linked to in the input volume. For example, if we choose to incorporate 12 filters, then it will result in a volume of [32x32x12].
- **ReLU Layer:** It is specially used to apply an activation function elementwise, like as max (0, x) thresholding at zero. It results in ([32x32x12]), which relates to an unchanged size of the volume.
- **Pooling:** This layer is used to perform a downsampling operation along the spatial dimensions (width, height) that results in [16x16x12] volume.

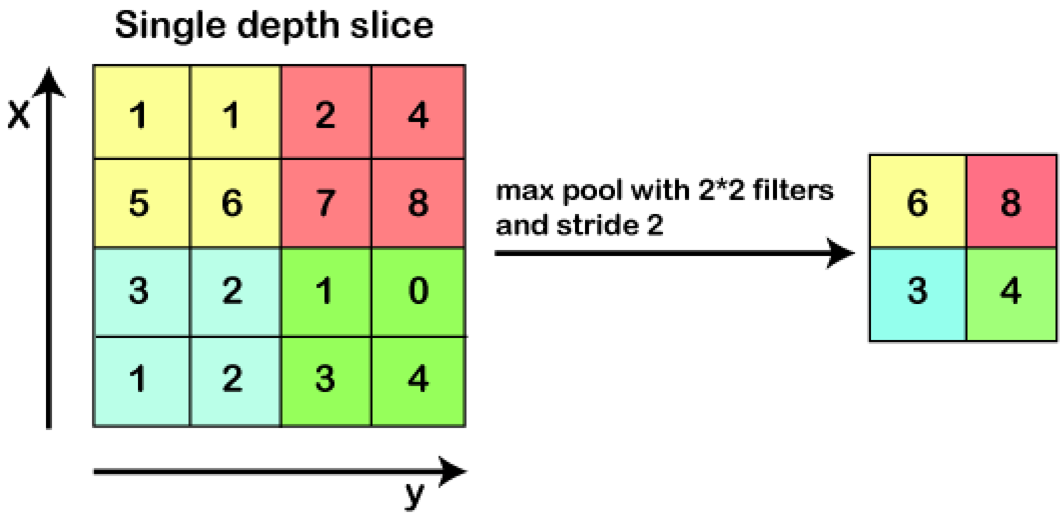


Fig 4.4 Polling Layer

- **Locally Connected:** It can be defined as a regular neural network layer that receives an input from the preceding layer followed by computing the class scores and results in a 1-Dimensional array that has the equal size to that of the number of classes.

We will start with an input image to which we will be applying multiple feature detectors, which are also called as filters to create the feature maps that comprises of a Convolution layer. Then on the top of that layer, we will be applying the ReLU or Rectified Linear Unit to remove any linearity or increase non-linearity in our images.

Next, we will apply a Pooling layer to our Convolutional layer, so that from every feature map we create a Pooled feature map as the main purpose of the pooling layer is to make sure that we have spatial invariance in our images. It also helps to reduce the size of our images as well as avoid any kind of overfitting of our data. After that, we will flatten all of our pooled images into one long vector or column of all of these values, followed by inputting these values into our artificial neural network. Lastly, we will feed it into the locally connected layer to achieve the final output.

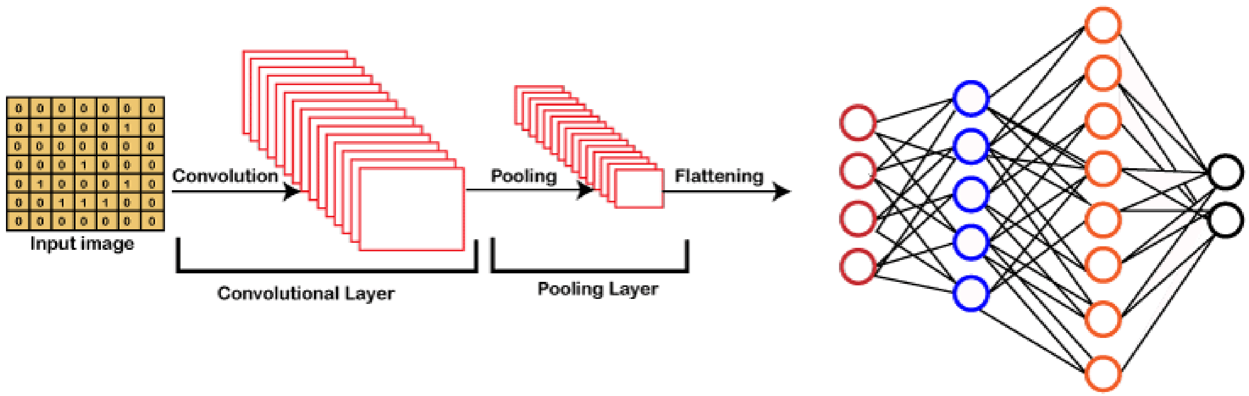


Fig 4.5 Complete CNN in one view

4.3 Building a CNN

we are going to build together the convolutional neural network and, more specifically, the whole architecture of the artificial neural network. So, it is actually going to start the same as with our artificial neural network because the convolutional neural network is still a sequence of layers.

Therefore, we are going to initialize our CNN with the same class, which is the sequential class.

Initializing the CNN

So, this is the first step where we are not only going to call the sequential class but will actually create the cnn variable, which will represent this convolutional neural network. And this **cnn** variable will be created once again as an instance of that sequential class allows us to create an artificial neural network as a sequence of layers.

First, we will need to call the TensorFlow that has a shortcut **tf** from which we are going to call Keras library from where we are going to get access to the model's module, or we can say from where we are going to call that sequential class.

After this, we will step by step use the add method to add different layers, whether they are convolutional layers or fully connected layers, and in the end, the output layer. So, we are now going to successfully use the add method starting with the step1: convolution.

Step1: Convolution

We will first take the **cnn** object or the convolutional neural network from which we will call the add method to add our very first convolutional layer, which will further be an object of a certain class, i.e., **Conv2D** class. And this class, just like the dense class that allows us to build a fully connected layer, belongs to the same module, which is the layer module from the Keras library, but this time it is the TensorFlow.

Inside the class, we are going to pass three important parameters, which are as follows:

- The first parameter is the **filters**, which is the number of feature detectors that we want to apply to images for feature detection.
- The **kernel_size** is exactly the size of the feature detector, i.e., the number of rows, which is also the number of columns.
- The third one is the **activation** but here we are not going to keep the default value for the activation parameter corresponding to the activation function, because indeed as long as we don't reach the output layer, we rather want to get a rectifier activation function. That is why we will choose the **ReLU** parameter name once again as it corresponds to the rectifier activation function.
- Lastly, the **input_shape** parameter because it is necessary to specify the input shape of inputs. Since we are working with the colored images, so the input_shape will be [64, 64, 3].

Step2: Pooling

Next, we will move on to applying pooling, and more specifically, if we talk about, we are going to apply the max pooling, and for that, we will again take cnn object from which we are going to call our new method. Since we are adding the pooling layer to our convolutional layer, so we will again call the add method, and inside it, we will create an object of a max-pooling layer or an instance of a certain class, which is called **MaxPool2D** class. Inside the class, we will **pass pool_size** and **strides** parameters.

Adding a second layer

Now we will add our second layer, for which again we have to undergo applying convolutional as well as pooling layer just like we did in the previous step, but here will need to change the **input_shape** parameter because it is entered only when we add our very first layer to automatically connect that first layer to the input layer, which automatically adds the input layer.

Since we are already here adding the second convolution layer, so we can simply remove that parameter. So, we are all set to move on to step3.

Step3: Flattening

In the third step, we will undergo flattening the result of these convolutions and pooling into a one-dimensional vector, which will become the input of a fully connected layer neural network in a similar way as we did in the previous section. We will start with again taking our **cnn** object from which we will call the **add** method because the way we are going to create that flattening layer is once again by creating an instance of the **Flatten** class, such that Keras will automatically understand that this is the result of all these convolutions and pooling, which will be flattened into the one-dimensional vector.

So, we just need to specify that we want to apply flattening and to do this we will have to call once again the layers module by the Keras library by TensorFlow from which we are actually going to call the flatten class, and we don't need to pass any kind of parameter inside it.

Step4: Full Conversion

In step 4, we are exactly in the same situation as before building a fully connected neural network. So, we will be adding a new fully-connected layer to that flatten layer, which is nothing but a one-dimensional vector that will become the input of a fully connected neural network. And for this, we will again start by taking a **cnn** neural network from which we are going to call the **add** method because now we are about to add a new layer, which is a fully connected layer that belongs to **tf.keras.layers**. But this time, we will take **a Dense** class followed by passing **units**, which is the number of hidden neurons we want to have into this fully connected layer and **activation function** parameter.

Step5: Output Layer

Here we need to add the final output layer, which will be fully connected to the previous hidden layer. Therefore, we will call the Dense class once again in the same way as we did in the previous step but will change the value of the input parameters because the numbers of units in the output layer are definitely not 128. Since we are doing binary classification, it will actually be one neuron to encode that binary class into a 'cat' or 'dog'. And for the activation layer, it is recommended to have a sigmoid activation function. Otherwise, if we were doing multiclass classification, we would have used the SoftMax activation function.

CHAPTER 5 METHODOLOGY

5.METHODOLOGY

5.1 Dataset Preparation

A chest X-ray database was used to experiment with this study. This database is currently one of the popular public X-ray databases, containing 3616 COVID-19 cases along with 10,192 healthy, 6012 lung opacity and 1345 viral pneumonia images. However, only COVID-19 (3616) and healthy (10,192) X-ray images were extracted for this study. As a result, the dataset includes studies of COVID-19 and healthy individuals with a matrix resolution of 299 × 299 (two X-ray examples are shown in Figure 5.1). EnsNet , a system for scene-text removal, was used to remove annotations from certain images. EnsNet is capable of automatically removing all of the text or annotation from an image without any prior knowledge. Data augmentation and image enhancement techniques are performed to enhance the quantity and variety of images given to the classifier for classification. Image augmentations used include horizontal flip, rotation, width shift and height shift on all the extracted data from the original dataset. As chest X-ray images are not vertically balanced, vertical flip was not applied. All augmentation parameters are shown in Table 1. After image augmentation, the dataset was increased to a larger dataset consisting of 26,000 COVID-19 and 26,000 healthy chest X-ray images. Besides, image enhancement applied Histogram equalization, Spectrum, Grays and Cyan. The N-CLAHE algorithm was then used to normalize pictures and highlight smaller features for machine learning classifiers to notice. Thereafter, the images were scaled down to the classifier's standard resolution (for instance AlexNet was 256 × 256 pixels, whereas GoogLeNet was 224 × 224 pixels). After resizing the picture, the machine learning classifier used the enhanced (52,000) images in a ratio of 80% data for training, whereas 20% was used for testing. Table 2 shows the details of the dataset.

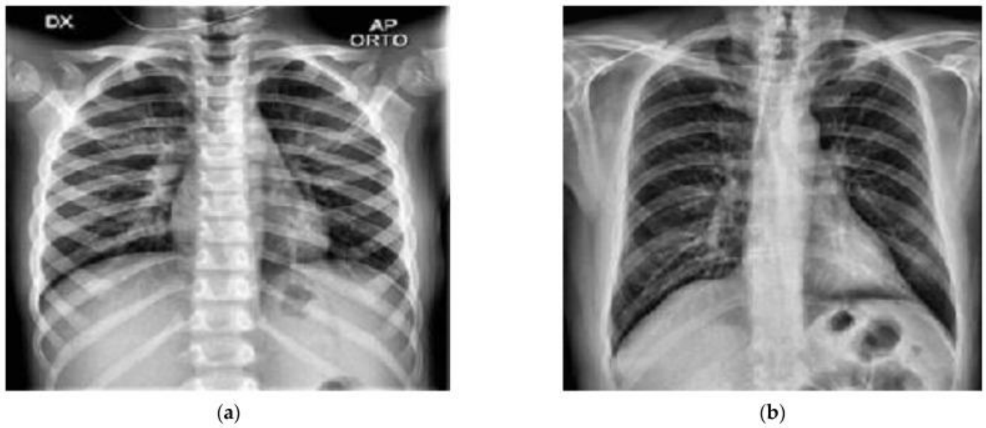


Fig 5.1 Chest X-ray image data samples(a)Healthy;(b)COVID-19

Table 1

Parameters of data augmentation.

Augmentation Technique	**Range**
Horizontal flip	True
Rotation range	10
Width shift range	0.1
Height shift range	0.1
Vertical flip	False

Table 2

Dataset Description.

Features	**Values**
Total Number of Images	52,000
Disease Types	2
Dimension (Size in Pixel)	Classifier’s Resolution (i.e., AlexNet is 256 × 256 pixels)
Color Grading	Grays, Cyan, Spectrum
COVID-19 Images	26,000 (After Augmentation)
Healthy Images	26,000 (After Augmentation)
Training Images	41,600
Testing Images	10,400

5.2 Model Selection

One of the main goals of this research is to obtain appropriate classification results utilizing freely available data (increased to high volume data by using enhancement techniques) with the combined transfer learning models. This research was undertaken to choose a CNN-based deep learning model that is appropriate for COVID-19 image classification investigation. The primary aim is to propose a modified novel deep-learning-based CNN model to gain the highest accuracy on a large volume of chest X-ray data with minimal compilation time and compare the modified novel approach (accuracy, efficiency, compilation time) with existing deep learning models on the same dataset. Figure 5.2 shows the system diagram of the experiment.

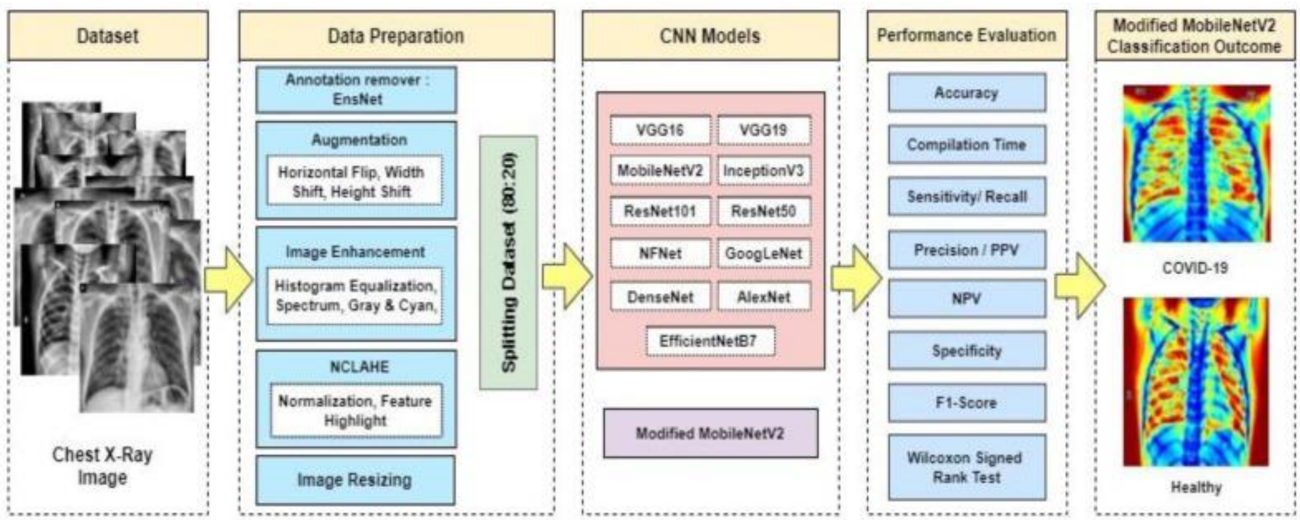


Fig 5.2 Proposed system diagram

As a result, the focus was on models that are widely used, appropriate for transfer learning, and easily available in packaged forms via trustworthy public libraries such as Keras to identify the best suitable model for this study. Due to this, some basic models are compared with the proposed novel model. These are all available as Keras API models [60], and all of these enable transfer learning by pre-applying the ImageNet weights to the model.

1.VGG19 and VGG16: The Visual Geometry Group is abbreviated as VGG. VGG16 is built using multiple 33 kernel-sized filters sequentially (11 and 5 in the first and second convolutional layers, respectively). VGG's input is set to a 224 × 244 RGB picture. The VGG-19 convolutional neural network was trained using over a million pictures from the ImageNet database. The network has a depth of 19 layers and is capable of classifying images of multiple classes. The VGG architectures' primary concept is to keep the convolution size modest and constant while designing an extremely deep network.

2.InceptionV3

InceptionV3 makes use of label smoothing, factorized 7 × 7 convolutions, and an auxiliary classifier to transmit label information down the network, as well as batch normalization for sidehead layers. It features smaller convolutions for quicker training and lower grid size to overcome computational cost constraints. Numerous optimization methods have been proposed for an InceptionV3 model in order to relax the restrictions and facilitate model adaptability. Factorized convolutions, regularization, dimension reduction, and parallelized calculations are all included in the methods.

3.ResNet50 and 101

ResNet50's architecture is divided into 4 stages. The network may accept an input image with a height, width of multiples of 32, and channel width. The network may accept an input image with a height, width of multiples of 32, and channel width Each ResNet architecture conducts initial convolution and max-pooling with a kernel size of 7 × 7 and 3 × 3, respectively. Each 2-layer block is replaced with this 3-layer bottleneck block in the 34-layer net, resulting in a 50-layer ResNet. A 101-layer ResNet is created by adding additional 3-layer blocks.

4.GoogLeNet

GoogLeNet is a deep convolutional neural network with 22 layers and almost 12× fewer parameters compared to Inception architecture. However, by adding more layers, the number of parameters grows, and the network may overfit. The pre-trained network accepts images with a resolution of 224 × 224. In GoogLeNet, global average pooling was utilized instead of a fully linked layer. The architecture makes use of the Activation, AveragePooling2D, and Dense layers.

5.MobileNetV2

MobileNetV2 introduces a new module with an inverted residual structure. With MobileNetV2, state-of-the-art object recognition and semantic segmentation are accomplished. MobileNetV2's architecture begins with a fully convolutional layer with 32 filters and 19 residual bottleneck layers. Typically, the network requires 300 million multiply-add operations and utilizes 3.4 million parameters. Accuracy is increased by removing ReLU6 from the output of each bottleneck module.

6.AlexNet

AlexNet is made up of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. Each convolutional layer is composed of convolutional filters and a ReLU nonlinear activation function. Max pooling is accomplished using the pooling layers. Due to the existence of completely linked layers, the input size 224×224 ×3 is fixed. If the input picture is grayscale, it is converted to RGB by duplicating the single channel to create a three-channel RGB image. AlexNet's total parameter count is 60 million, with a batch size of 128.

7.EfficientNet B7

To enhance performance, a new baseline network was created using the AutoML MNAS framework, which improves both accuracy and efficiency (FLOPS). The resultant architecture is comparable to MobileNetV2 and MnasNet in that it utilizes mobile inverted bottleneck convolution (MBConv), but is somewhat bigger owing to an increased FLOP budget. The basic network is then scaled up to create a family of models called EfficientNets. EfficientNetB7 does not include any pre-trained weights.

8.DenseNet 121

Each layer in a DenseNet design is directly linked to every other layer, resulting in the term Densely Connected Convolutional Network. There are L(L + 1)/2 direct connections between ‘L’ levels. The feature maps from previous layers are not averaged, but concatenated and utilized as inputs in each layer. As a result, DenseNets need fewer parameters than a comparable conventional CNN, which enables feature reuse by discarding duplicate feature maps. Dense Blocks, in which the size of the feature maps stays constant inside a block but the number of filters varies. These layers in between are referred to as Transition Layers and are responsible for downsampling the image by using batch normalization, 1×1 convolution, and 2×2 pooling layers.

9.NFNet

NFNets is an abbreviation for Normalizer-Free Networks. NFNets are a subclass of modified ResNets that achieve competitive accuracy in the absence of batch normalization. NFNets scales the activations at the start and end of the residual branch using two scalers (α and β). Scaled Weight Standardization is used in NFNets to prevent mean shift. Additionally, Adaptive Gradient Clipping was used to train NFNets with larger batch sizes and learning rates.

10.Modified MobileNetV2

Modified MobileNetV2 is likewise a design suited for mobile as well as computer vision like MobileNetV2. To assist with computer vision, deep learning techniques are now being utilized in other areas including robotics, the Internet of Things (IoT), and Natural Language Processing. The modified MobileNetV2 model, as well as the CNN layers, are used to predict and categorize diseases in chest X-ray images in this study. The modified MobileNetV2 architecture includes a set of hidden layers based on a bottleneck residual block, as well as a depth-wise separable convolution that significantly lowers the number of parameters and results in a lightweight neural network that differs from typical convolution. The standard convolution is substituted with a depth-wise convolution with a single filter, followed by a depth-wise severable convolution with a pointwise convolution.

5.3 VGG-16 architecture

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual computer vision competition. Each year, teams compete on two tasks. The first is to detect objects within an image coming from 200 classes, which is called object localization. The second is to classify images, each labeled with one of 1000 categories, which is called image classification. VGG 16 was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University in 2014 in the paper "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION". This model won the 1st and 2nd place on the above categories in 2014 ILSVRC challenge.

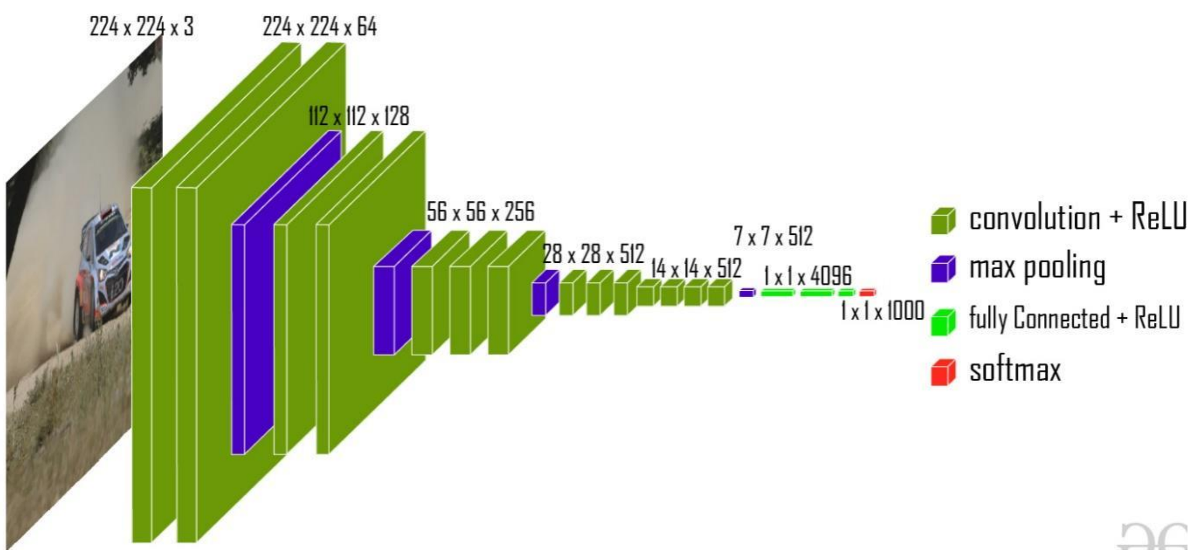


Fig 5.3 VGG-16 Architecture

CHAPTER 6 ARTIFICIAL NEURAL NETWORK

6.ARTIFICIAL NEURAL NETWORK

6.1 Introduction

Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions.

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

Artificial neural network tutorial covers all the aspects related to the artificial neural network. In this tutorial, we will discuss ANNs, Adaptive resonance theory, Kohonen self-organizing map, Building blocks, unsupervised learning, Genetic algorithm, etc.

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

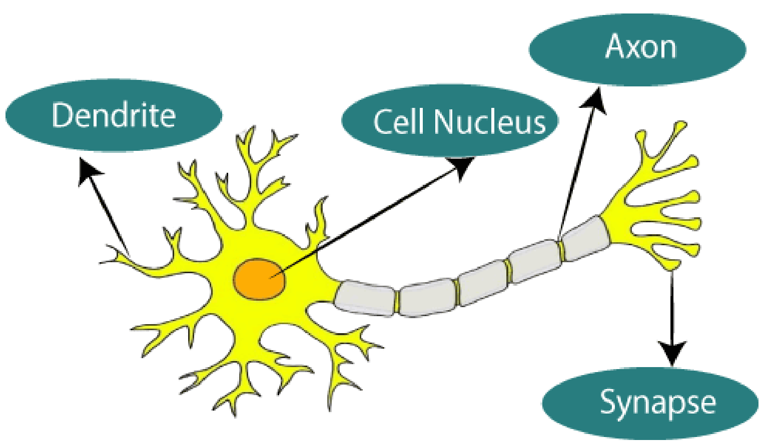


Fig 6.1 Biological Neural Network

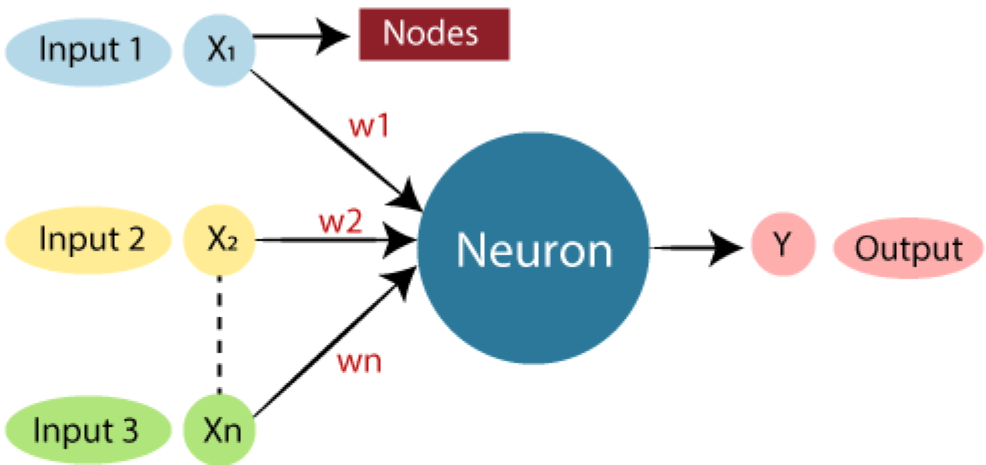


Fig 6.2 typical Artificial Neural Network

Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

Relationship between Biological neural network and artificial neural network:

Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

6.2 Architecture of an artificial neural network

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

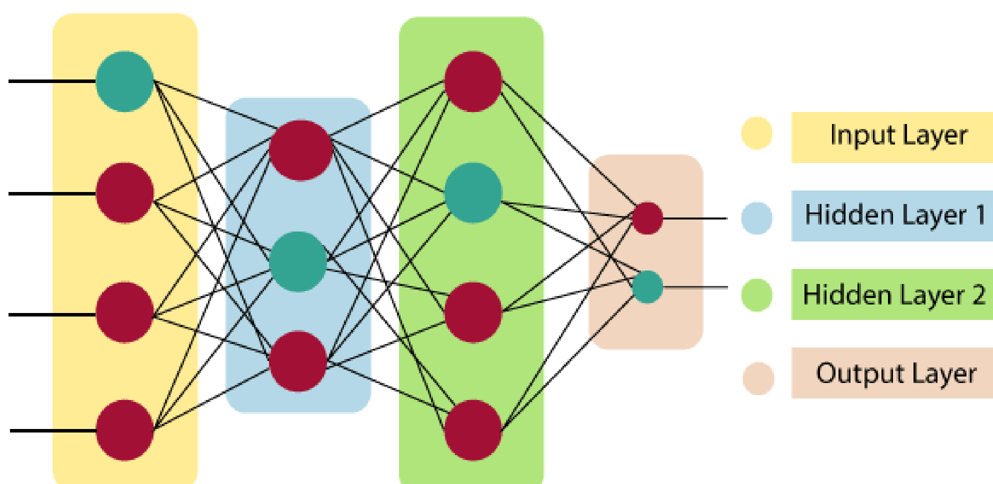


Fig 6.3 artificial neural network

Input Layer:

As the name suggests, it accepts inputs in several different formats provided by the programmer.

Hidden Layer:

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

6.3 Working of ANN

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.

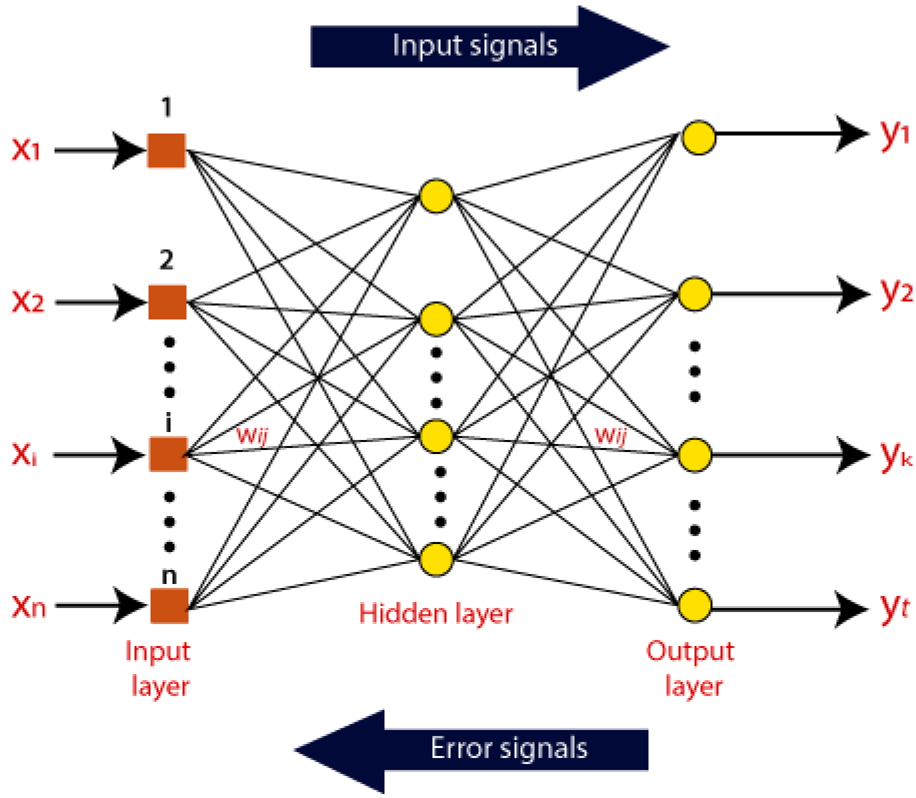


Fig 6.4 ANN

Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions.

6.4 Types of ANN

There are various types of Artificial Neural Networks (ANN) depending upon the human brain neuron and network functions, an artificial neural network similarly performs tasks. The majority of the artificial neural networks will have some similarities with a more complex biological partner and are very effective at their expected tasks. For example, segmentation or classification.

Feedback ANN:

In this type of ANN, the output returns into the network to accomplish the best-evolved results internally. As per the **University of Massachusetts**, Lowell Centre for Atmospheric Research. The feedback networks feed information back into itself and are well suited to solve optimization issues. The Internal system error corrections utilize feedback ANNs.

Feed-Forward ANN:

A feed-forward network is a basic neural network comprising of an input layer, an output layer, and at least one layer of a neuron. Through assessment of its output by reviewing its input, the intensity of the network can be noticed based on group behavior of the associated neurons, and the output is decided. The primary advantage of this network is that it figures out how to evaluate and recognize input patterns.

6.5 Advantages of Artificial Neural Network (ANN)

Parallel processing capability:

Artificial neural networks have a numerical value that can perform more than one task simultaneously.

Storing data on the entire network:

Data that is used in traditional programming is stored on the whole network, not on a database. The disappearance of a couple of pieces of data in one place doesn't prevent the network from working.

Capability to work with incomplete knowledge:

After ANN training, the information may produce output even with inadequate data. The loss of performance here relies upon the significance of missing data.

Having a memory distribution:

For ANN is to be able to adapt, it is important to determine the examples and to encourage the network according to the desired output by demonstrating these examples to the network. The succession of the network is directly proportional to the chosen instances, and if the event can't appear to the network in all its aspects, it can produce false output.

Having fault tolerance:

Extortion of one or more cells of ANN does not prohibit it from generating output, and this feature makes the network fault-tolerance

6.6 Disadvantages of Artificial Neural Network:

Assurance of proper network structure:

There is no particular guideline for determining the structure of artificial neural networks. The appropriate network structure is accomplished through experience, trial, and error.

Unrecognized behaviour of the network:

It is the most significant issue of ANN. When ANN produces a testing solution, it does not provide insight concerning why and how. It decreases trust in the network.

Hardware dependence:

Artificial neural networks need processors with parallel processing power, as per their structure. Therefore, the realization of the equipment is dependent.

Difficulty of showing the issue to the network:

ANNs can work with numerical data. Problems must be converted into numerical values before being introduced to ANN. The presentation mechanism to be resolved here will directly impact the performance of the network. It relies on the user's abilities.

The duration of the network is unknown:

The network is reduced to a specific value of the error, and this value does not give us optimum results.

CHAPTER 7 ACTIVATION FUNCTION

7.ACTIVATION FUNCTION

7.1 Linear Function :

Equation : Linear function has the equation similar to as of a straight line i.e. $y = ax$

No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer.

Range : -inf to +inf

Uses : Linear activation function is used at just one place i.e. output layer.

Issues : If we will differentiate linear function to bring non-linearity, result will no more depend on input “x” and function will become constant, it won’t introduce any ground-breaking behavior to our algorithm.

For example : Calculation of price of a house is a regression problem. House price may have any big/small value, so we can apply linear activation at output layer. Even in this case neural net must have any non-linear function at hidden layers.

7.2 Sigmoid Function:-

It is a function which is plotted as ‘S’ shaped graph.

Equation: A = 1/(1 + e-x)

Nature: Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.

Value Range: 0 to 1

Uses: Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.

7.3 Tanh Function:-

The activation that works almost always better than sigmoid function is Tanh function also knows as Tangent Hyperbolic function. It's actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.

Equation:-

f(x) = tanh(x) = 2/(1 + e-2x) - 1

OR

tanh(x) = 2 * sigmoid(2x) - 1

Value Range:- -1 to +1

Nature:- non-linear

Uses :- Usually used in hidden layers of a neural network as it's values lies between -1 to 1 hence the mean for the hidden layer comes out be 0 or very close to it, hence helps in centering the data by bringing mean close to 0. This makes learning for the next layer much easier.

7.4 RELU :-

Stands for Rectified linear unit. It is the most widely used activation function. Chiefly implemented in hidden layers of Neural network.

Equation :- A(x) = max(0,x). It gives an output x if x is positive and 0 otherwise.

Value Range :- [0, inf)

Nature :- non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.

Uses :- ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.

In simple words, RELU learns much faster than sigmoid and Tanh function.

7.5 Softmax Function :-

The softmax function is also a type of sigmoid function but is handy when we are trying to handle classification problems.

Nature :- non-linear

Uses :- Usually used when trying to handle multiple classes. The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.

Output:- The softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.

CHAPTER 8 GOOGLE COLABRATORY

8.GOOGLE COLABRATORY

8.1 Introduction

Google Colab was developed by Google to provide free access to GPU’s and TPU’s to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.

8.2 Google Colab Features

Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

- Interactive tutorials to learn machine learning and neural networks.
- Write and execute Python 3 code without having a local setup.
- Execute terminal commands from the Notebook.
- Import datasets from external sources such as Kaggle.
- Save your Notebooks to Google Drive.
- Import Notebooks from Google Drive.
- Free cloud service, GPUs and TPUs.
- Integrate with PyTorch, Tensor Flow, Open CV.
- Import or publish directly from/to GitHub.

8.3 Uses of Google Colab

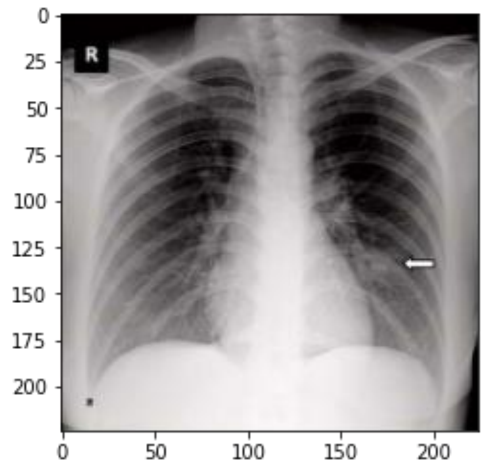
As a programmer, you can perform the following using Google Colab.

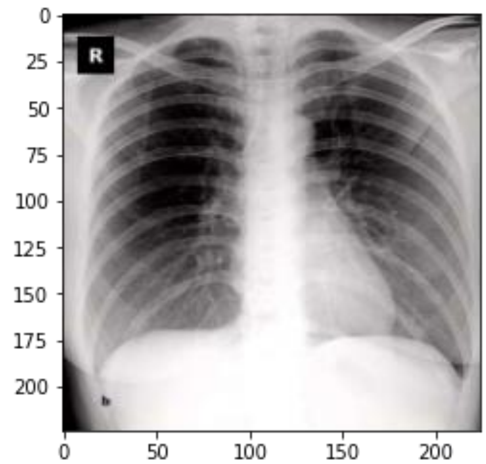
- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive

- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

CHAPTER 9 RESULTS AND DISCUSSION

9.RESULTS AND DISCUSSION



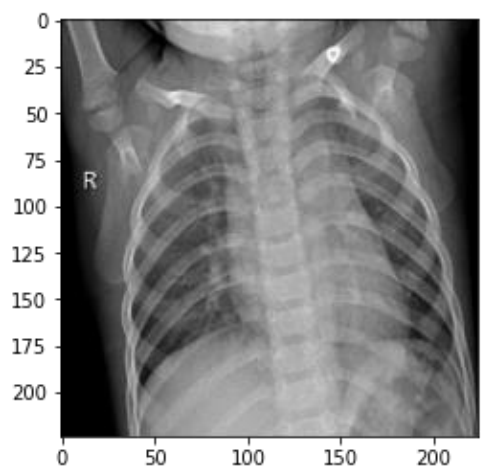


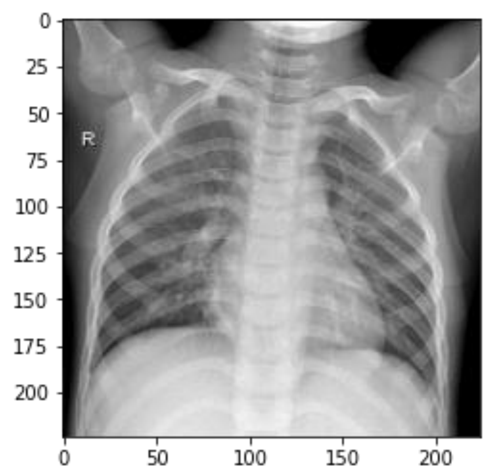
Result: binary 0

It indicates Covid positive

Result: binary 0

It indicates Covid positive





Result: binary 1

It indicates Covid negative

Result: binary 1

It indicates Covid negative

Declaring batch size 32

The Fig 9.1 represents the graph of Epochs versus Accuracy. Accuracies generated by the model on both the sets – training set and validation set – are marked on the graph during the 50 epochs time period.

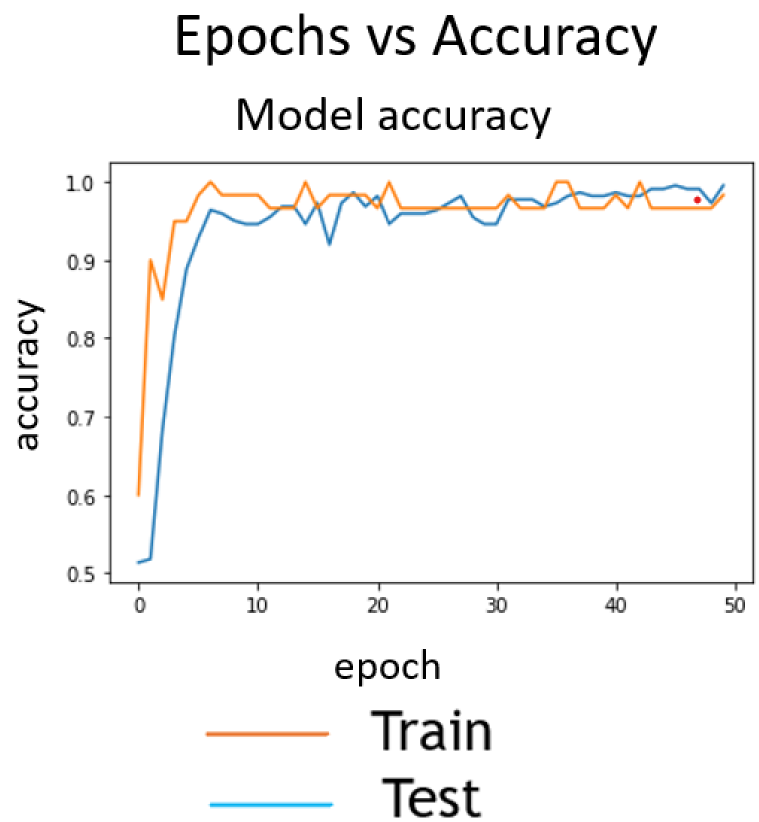


Fig 9.1: Accuracy obtained during 50 epochs

The Fig 9.2 represents the graph of Epochs versus Loss. Losses generated by the model on both the sets – training set and validation set – are marked on the graph during the 50 epochs time period.

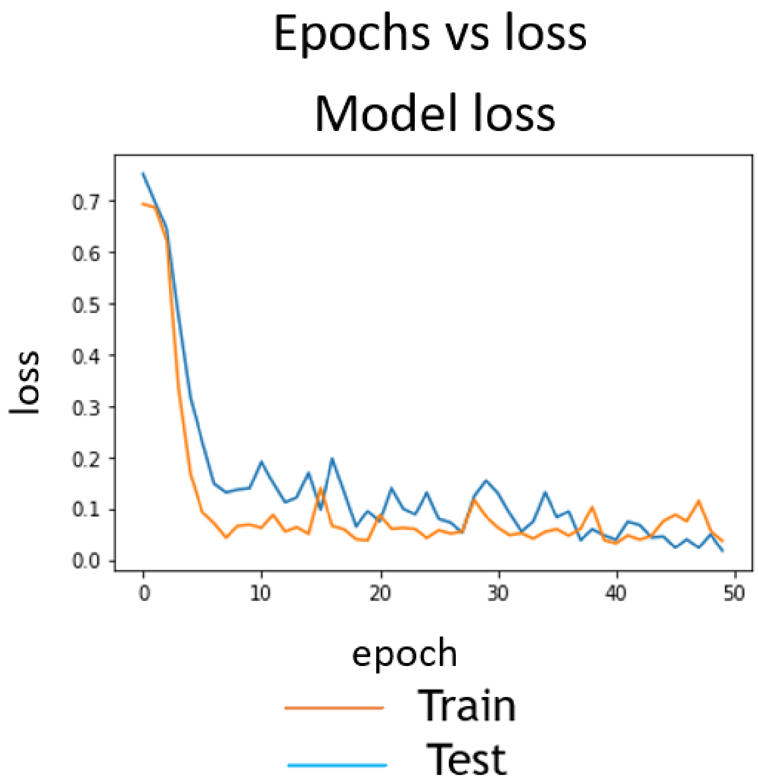


Fig 9.2: Loss obtained during 50 epochs

The Fig 9.3 represents the graph of Epochs versus Accuracy. Accuracies generated by the model on both the sets – training set and validation set – are marked on the graph during the 100 epochs time period.

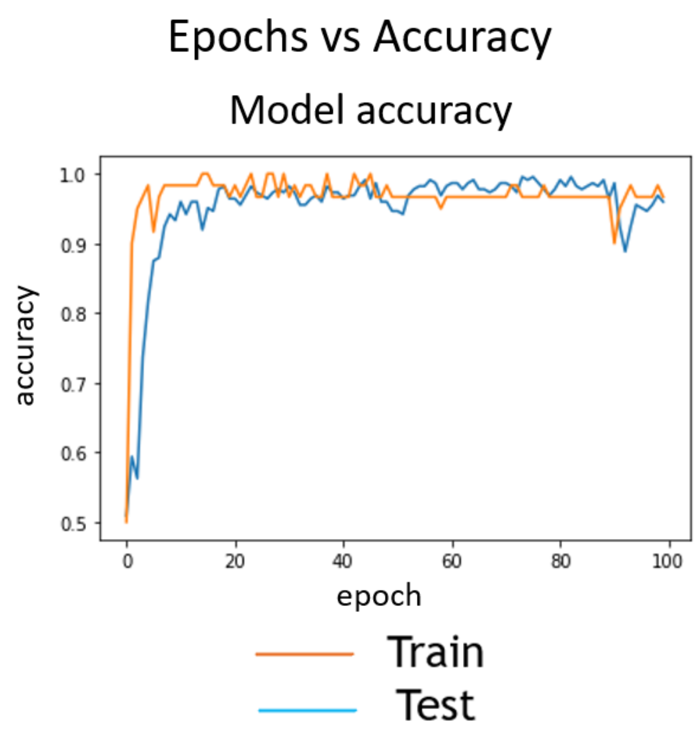


Fig 9.3: Accuracy obtained during 100 epochs

The Fig 9.4 represents the graph of Epochs versus Loss. Losses generated by the model on both the sets – training set and validation set – are marked on the graph during the 100 epochs time period.

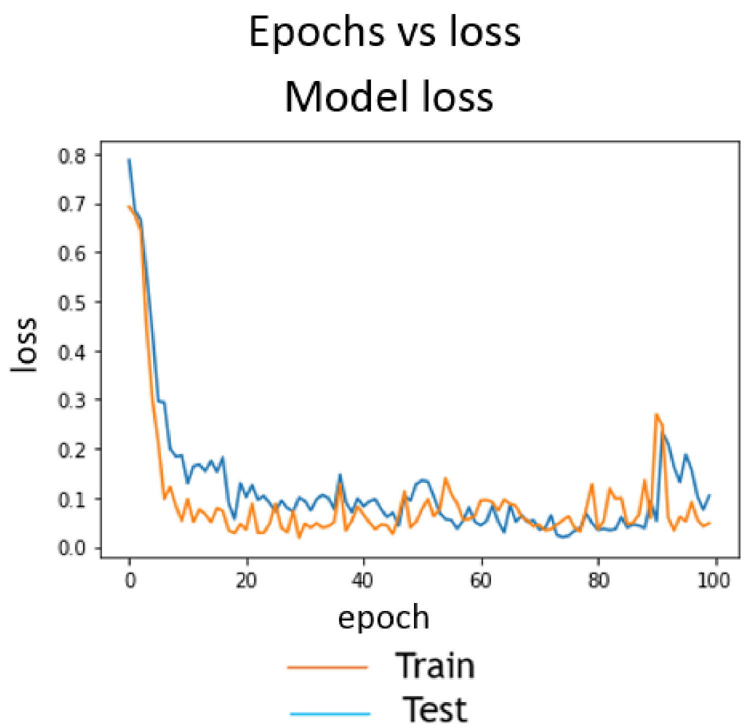


Fig 9.4: Loss obtained during 100 epochs

Declaring batch size 64

The Fig 9.5 represents the graph of Epochs versus Accuracy. Accuracies generated by the model on both the sets – training set and validation set – are marked on the graph during the 50 epochs time period.

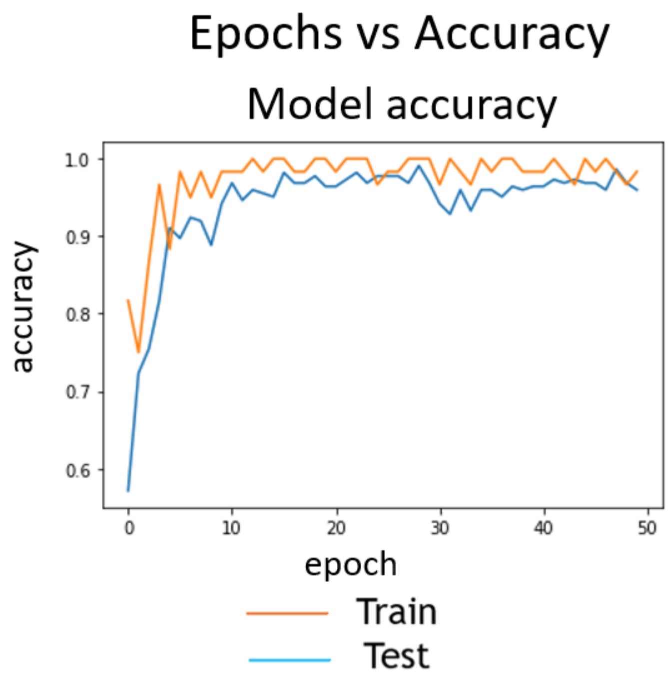


Fig 9.5: Accuracy obtained during 50 epochs

The Fig 9.6 represents the graph of Epochs versus Loss. Losses generated by the model on both the sets – training set and validation set – are marked on the graph during the 50 epochs time period.

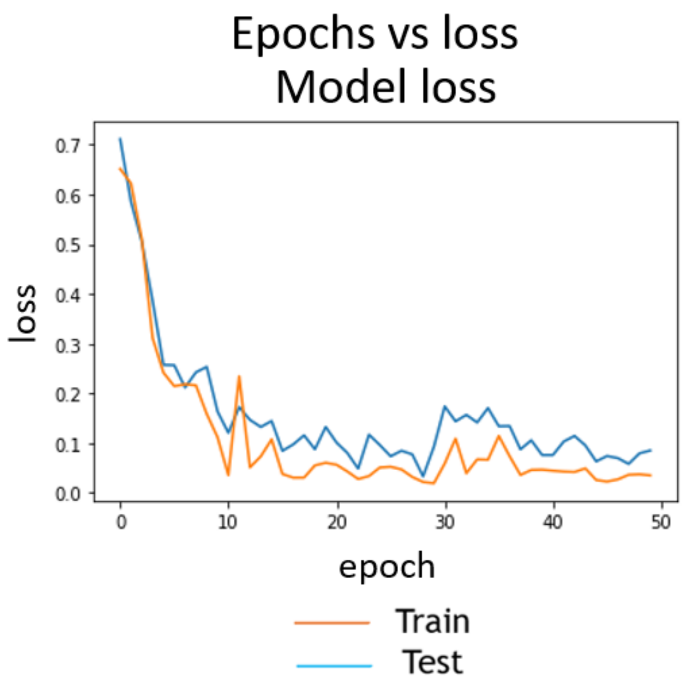


Fig 9.6: Loss obtained during 50 epochs

The Fig 9.7 represents the graph of Epochs versus Accuracy. Accuracies generated by the model on both the sets – training set and validation set – are marked on the graph during the 100 epochs time period.

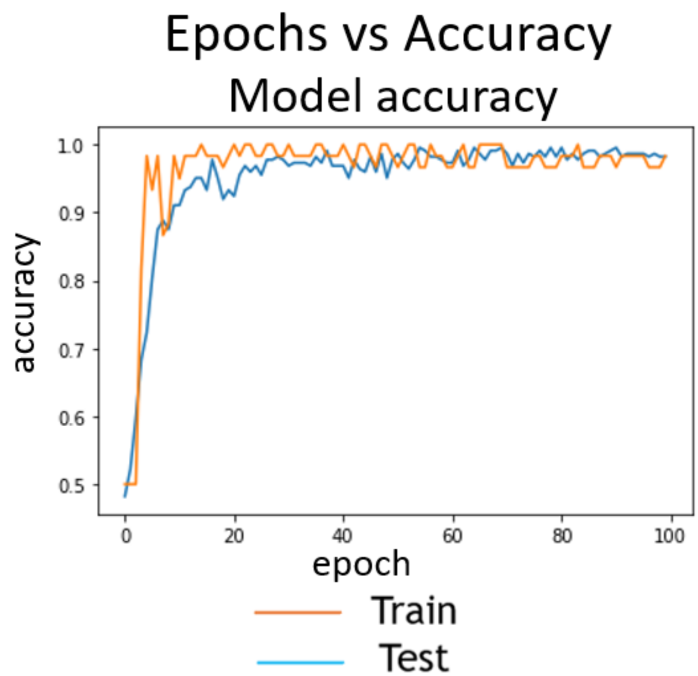


Fig 9.7: Accuracy obtained during 100 epochs

The Fig 9.8 represents the graph of Epochs versus Loss. Losses generated by the model on both the sets – training set and validation set – are marked on the graph during the 50 epochs time period.

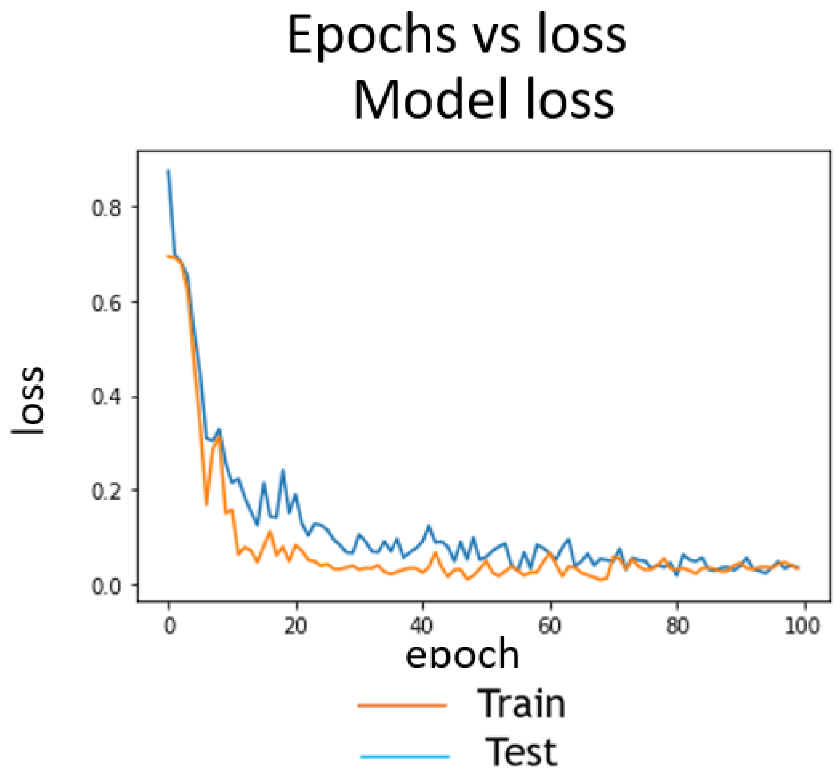


Fig 9.8: Accuracy obtained during 100 epochs

CHAPTER 10 CONCLUSION

10.CONCLUSION

The new COVID-19 virus has caused thousands of deaths, especially in elders and patients with health conditions. The standard method for detection and diagnosis of COVID-19 is the reverse-transcription polymerase chain reaction (RT-PCR) test after collection of proper respiratory tract specimen, which is time-consuming and in many cases not affordable thus the development of new low-cost rapid tests of diagnostic tools to support clinical assessment is needed. We presented an evaluation of transfer learning using pretrained deep convolutional neural network models for COVID-19 identification using chest X-ray images. Two publicly available datasets were used in different experimental setups. In specific, we tested the binary COVID-19 identification performance of several convolutional neural network models using 10-fold cross validation on each dataset separately, then we tested the transferability of the models by using one dataset for training and the other for testing and vice versa. Finally, we merged the two datasets and performed 10- fold cross validation to investigate the effect of the size of available data in accuracy, precision and recall. The experimental evaluation demonstrated the potential of building diagnostic tools for automatic detection of COVID19 positive cases from chest X-ray images and deep convolutional neural networks and the development of larger and clinically standardized datasets would further help in this direction.

REFERENCE

[1] Roosa K, Lee Y, Luo R, Kirpich A, Rothenberg R, Hyman JM, and et al. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. Infectious Disease Modelling, 5:256-263, 2020

[2] V. M. Corman et al., “Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr,” Eurosurveillance, vol. 25, no. 3, p. 2000045, 2020.

[3] J. Watson, P. F. Whiting, and J. E. Brush, “Interpreting a covid-19 test result,” British Medical Journal, vol. 369, pp. 1–7, 2020.

[4] W. H. Self, D. M. Courtney, C. D. McNaughton, R. G. Wunderink, and J. A. Kline, “High discordance of chest x-ray and computed tomography for detection of pulmonary opacities in ed patients: implications for diagnosing pneumonia,” American Journal of Emergency Medicine, vol. 31, no. 2, pp. 401–405, 2013.

[5] Jaiswal AK, Tiwari P, Kumar S, Gupta D, Khanna A, and Rodrigues JJ. Identifying pneumonia in chest X-rays: A deep learning approach. Measurement, 145:511-518, 2019.

[6] B. Yan, X. Tang, B. Liu, J. Wang, Y. Zhou, G. Zheng, Q. Zou, Y. Lu, and W. Tu, “An improved method of covid-19 case fitting and prediction based on lstm,” arXiv preprint arXiv:2005.03446, 2020.

[7] Dandıl, E., C¸ akiro˘glu, M., Ek¸si, Z., Ozkan, M., Kurt, ¨ O.K., Canan, A.: Artificial neural ¨ network-based classification system for lung nodules on computed tomography scans. In: 2014 6th International conference of soft computing and pattern recognition (SoCPaR), pp. 382–386. IEEE (2014)

[8] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature 521(7553), 436 (2015)

[9] L. Luo et al., “Deep mining external imperfect data for chest x-ray disease screening,” arXiv preprint arXiv:2006.03796, 2020.

[10] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2097–2106.

[11] Aboul Ella Hassanien Sr., Lamia Nabil Mahdy Jr., Kadry Ali Ezzat Jr., Haytham H. Elmousalami Jr., Hassan Aboul Ella Jr. (2020), Automatic X-ray COVID-19 Lung Image Classification System based on MultiLevel Thresholding and Support Vector Machine, medRxiv, doi: https://doi.org/10.1101/2020.03.30.20047787.

[12] Abbas A , Abdelsamea M M, Medhat Gaber M, (2020) Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network, medRxiv, doi: https://doi.org/10.1101/2020.03.30.20047456.

[13] Sethy PK, Behera SK. (2020) Detection of Coronavirus Disease (COVID-19) Based on Deep Features, International Journal of Mathematical, Engineering and Management Sciences Vol. 5, No. 4, 643-651.

[14] Wang L, Wong A. (2020), COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images; arXiv: 2003.09871.

[15] Li et al. (2020), Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. RSNA Radiology, https://doi.org/10.1148/radiol.2020200905.

[16] Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). [online] Switzerland: Springer, Cham,159- 168.

[17] Gozes O et. Al (2020), Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis, Radiology: Artificial Intelligence.

[18] Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Al-Emadi, N. (2020). Can AI help in screening Viral and COVID-19 pneumonia? arXiv preprint arXiv:2003.13145.

[19] Darshan. (2020). COVID-19 Detection X-Ray Dataset. Retrieved from https://www.kaggle.com/darshan1504/covid19-detection-xray-dataset.

[20] Bradski, G. (2000). The opencv library. Dr Dobb's J. Software Tools, 25, 120-125.