COVID DETECTION USING DEEP LEARNING MODEL WITH X-RAY

A

Project Report

Submitted to

Jawaharlal Nehru Technological University Hyderabad

In partial fulfilment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY
in

ELECTRONICS & COMMUNICATION ENGINEERING

By

KAASOJU SAI TEJA (18VE1A04E1)

ALIGIREDDY ANIRUDH REDDY (18VE1A04C2)

SAI KRUTHI KANAKUNTLA (18VE1A04G7)

Under the Guidance of

Mr. S. SUBRAHMANYAM

Assistant Professor

SREYAS Institute of Engineering & Technology
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NBA & NAAC)
Hyderabad | PIN: 500068

(2018 – 2022)

SREYAS Institute of Engineering & Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NBA & NAAC)
Hyderabad | PIN: 500068

Certificate

This is to certify that the Project Report on **"COVID DETECTION USING DEEP LEARNING MODEL WITH X-RAY"** submitted by **Kaasoju Sai Teja, Aligireddy Anirudh Reddy, Sai Kruthi Kanakuntla** bearing Hall Ticket No. **18VE1A04E1, 18VE1A04C2, 18VE1A04G7**respectively in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad forthe academic year 2021-22 is a record of bonafide work carried out by them under our guidance and Supervision.

Guide	**Head of the Department**
Mr. S. SUBRAHMANYAM	**Prof. B. SREENIVASU**

Project Coordinator	**Signature of the External Examiner**
Prof. CHINNAM S V MARUTHI RAO	

SREYAS Institute of Engineering & Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NBA & NAAC)
Hyderabad | PIN: 500068

Declaration

We, **KAASOJU SAI TEJA, ALIGIREDDY ANIRUDH REDDY, SAI KRUTHI KANAKUNTLA** bearing Hall Ticket No. **18VE1A04E1, 18VE1A04C2, 18VE1A04G7** hereby declare that the mini project titled "**COVID DETECTION USING DEEP LEARNING MODEL WITH X-RAY**" done by us under the guidance of **MR.S.SUBRAHMANYAM**, which is submitted in the partial fulfilment of the requirement for the award of the B.Tech degree in **Electronics & Communication Engineering** at **Sreyas Institute of Engineering & Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

KAASOJU SAI TEJA (18VE1A04E1)

ALIGIREDDY ANIRUDH REDDY (18VE1A04C2)

SAI KRUTHI KANAKUNTLA (18VE1A04G7)

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the effortswith success.

We take this opportunity to acknowledge with thanks and deep sense of gratitudeto **Mr. S. Subrahmanyam (Assistant Professor, Department of ECE)** for her constant encouragement and valuable guidance during the Project work.

A Special note of Thanks to **Mr. B. Sreenivasu (Associate Professor and Head of the Department - ECE),** who has been a source of Continuous motivation and support. He had taken time and effort to guide and correct us all through the span of this work.

We owe very much to the **Department Faculty, Principal** and the **Management** who made our term at Sreyas a Steppingstone for our career. We treasure every moment we had spent in the college.

Last but not least, my heartiest gratitude to my parents and friends for their continuous encouragement and blessings. Without their support this work would not have been possible.

KAASOJU SAI TEJA (18VE1A04E1)

ALIGIREDDY ANIRUDH REDDY (18VE1A04C2)

SAI KRUTHI KANAKUNTLA (18VE1A04G7)

CONTENTS

ABSTRACT

Currently, the detection of coronavirus disease 2019 (COVID-19) is one of the main challenges in the world, given the rapid spread of the disease. Recent statistics indicate that the number of people diagnosed with COVID-19 is increasing exponentially, with more than 1.6 million confirmed cases; the disease is spreading to many countries across the world. In this study, we analyze the incidence of COVID-19 distribution across the world. We present an artificial-intelligence technique based on a deep convolutional neural network to detect COVID-19 patients using real-world datasets. Our system examines chest X-ray images to identify such patients. Our findings indicate that such an analysis is valuable in COVID-19 diagnosis as X-rays is conveniently available quickly and at low costs. Additionally, three forecasting methods—the prophet algorithm, auto regressive integrated moving average model, and long short-term memory neural network —we read opted to predict the numbers of COVID-19 confirmations, recoveries, and deaths over thenext7days. The prediction results exhibit promising performance and offer an average accuracy of 94.80% and 88.43% in Australia and Jordan, respectively. Our proposed system can significantly help identify the most infected cities, and it has revealed that coastal areas are heavily impacted by the COVID-19 spread as the number of cases is significantly higher in those areas than in non-coastal areas.

TABLE OF FIGURES

CHAPTER-1

1.1 INTRODUCTION

The novel coronavirus disease, named COVID-19 by the World Health Organization—is caused by a new coronavirus class known as SARS-CoV2 (Severe Acute Respiratory Syndrome Coronavirus. It is a single-stranded RNA (ribonucleic acid) virus that causes severe respiratory infections. The first COVID-19 cases were reported in December 2019, in Wuhan, Hubei province, China. As the virus has since spread worldwide, it has been given the status of pandemic by the World Health Organization. As of 16th February, 2021, 110 million people have been infected, and 2.4 million people have died due to COVID-19 . One of the best solutions has been detecting the virus in its early stages and then isolating the infected individuals by quarantining them, thus preventing healthy people from becoming infected.

In many cases, real-time reverse transcriptase-polymerase chain reaction (RRT-PCR) of nasopharyngeal swabs has been used for diagnosis. The RT-PCR throat swabs are collected from patients with COVID-19, and the RNA is then extracted. This process takes over two hours to complete and has a long turnaround time with limited sensitivity. The best alternative is to detect images of COVID-19 from radiology scans (chest X-ray images and chest computed tomography

(CT) images). The advantages of using chest X-rays over CT images are as follows: X-ray imaging systems are much more widely available than CT imaging systems, they are cost-effective, and digital X-ray images can be analysed at the point of acquisition, thus making the diagnosis process extremely quick.

X-ray images are grayscale. In medical imaging terms, these are images with values ranging from 0 to 255, where 0 corresponds to the completely dark pixels, and 255 corresponds to the completely white pixels. Different values on the X-ray image correlate to different areas of density. The different values are—dark: locations in the body which are filled with air are going to appear black, dark grey: subcutaneous tissues or fat, light grey: soft tissues like the heart and blood vessels, off white: bones such as the ribs, bright white: presence of metallic objects such as pacemakers or defibrillators. The way that physicians interpret an image is by looking at the borders between the different densities. The ribs appear off-white because they are dense tissues, but since the lungs are filled with air, the lungs appear dark. Similarly, below the lung is the hemidiaphragm, which is a soft tissue and hence appears light grey. This helps to find the location and extent of the lungs. If two objects with different densities are close to each other, they can be demarcated in an X-ray image. If something happens in the lungs, such as pneumonia, the air-dense lungs change into water-dense lungs. This causes the demarcation lines to fade since the pixel densities start closing in on the grayscale bar.

About 20% of patients infected with COVID-19 develop pulmonary infiltrates and some develop very serious abnormalities. The virus reaches the lungs' gas exchange units and infects alveolar type 2 cells. The most frequent CT abnormalities observed are ground-glass opacity, consolidation, and interlobular septal thickening in both lungs.

However, due to infection control issues related to patient transport to CT rooms, problems encountered with CT room decontamination, and the lack of CT scanner availability in different parts of the world, portable chest X-rays are likely to be one of the most common modalities for the identification and follow-up of COVID-19 lung abnormalities. Hence, a significant number of expert radiologists who can interpret these images are needed. Due to the ever-increasing number of cases of COVID-19 infection, it is becoming more difficult for radiologists to keep up with this demand. In this scenario, Deep Learning techniques have proven to be beneficial in both classifying abnormalities from lung X-ray images and aiding the radiologists to accurately predict COVID-19 cases in a reduced time frame.

While many studies have demonstrated success in detecting images of COVID-19 using Deep Learning with both CT scans and X-rays, most of the Deep Learning architectures require extensive programming.

What does it do in our lungs ?

Bronchi

The bronchi are the two large tubes that carry air from your windpipe to your lungs. You have a left and right main bronchus in each lung. After the main bronchi, these tubes branch out into segments that look like tree branches.

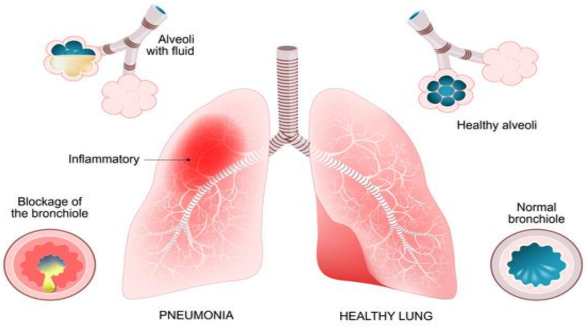


FIGURE 1. 1:BROCHI

Bronchioles

Bronchioles are air passages inside the lungs that branch off like tree limbs from the bronchi.

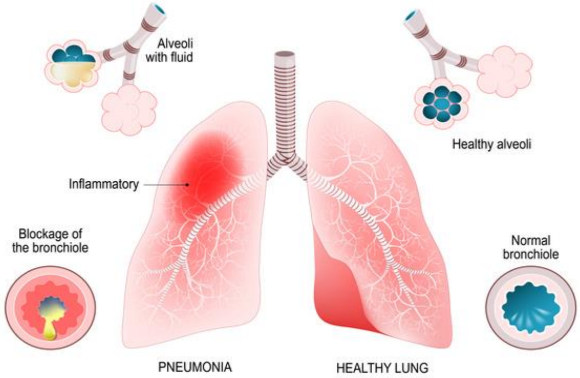


FIGURE 1. 2:BRONCHIOLES

CHAPTER-2

2.1 CONVOLUTIONAL NEURAL NETWORKS (CNN)

In deeplearning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution.

Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

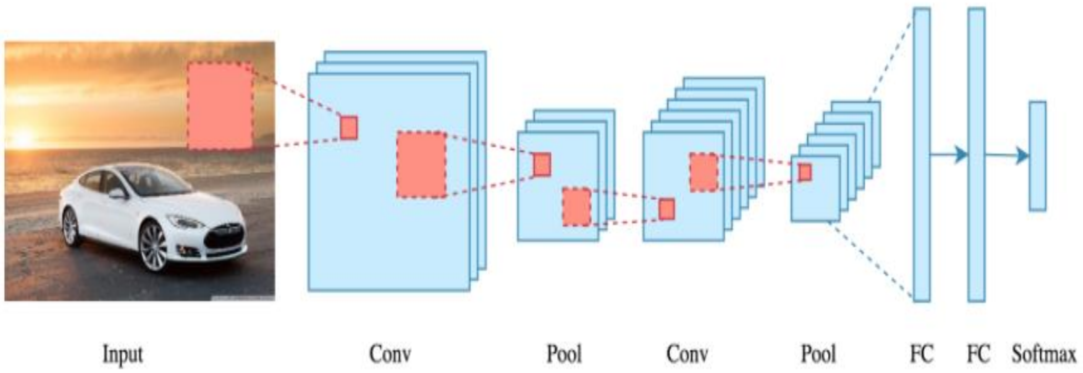


FIGURE 2. 1: CNN

An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same, but it has a single plane. Look at this image to understand more.

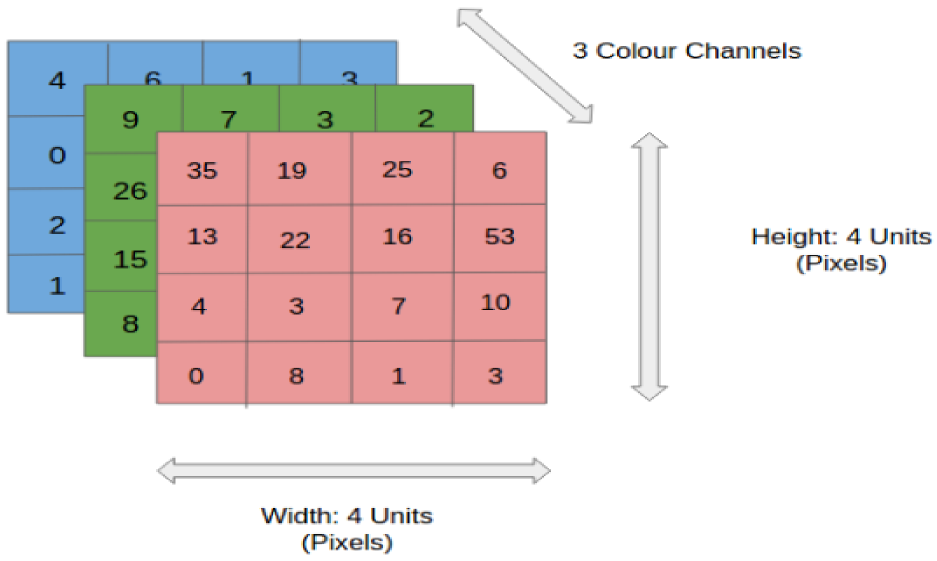


FIGURE 2. 2:RGB MATRIX

For simplicity, let's stick with grayscale images as we try to understand how CNNs work.

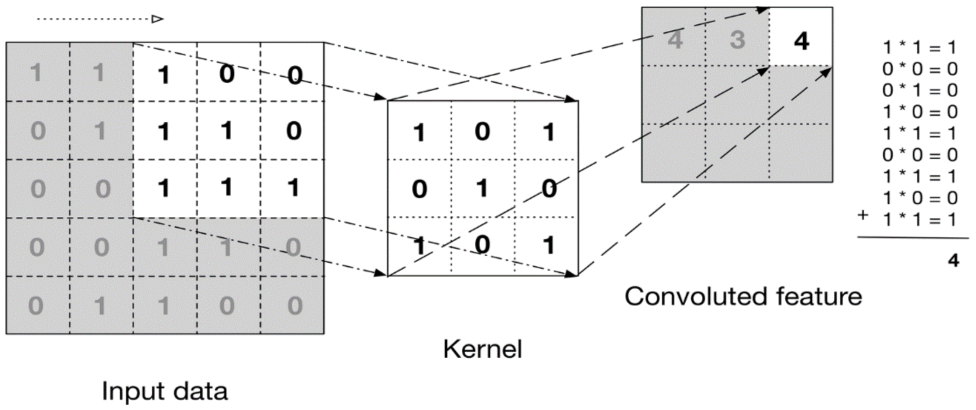


FIGURE 2. 3:CNN WORKING

The above image shows what a convolution is. We take a filter/kernel(3×3 matrix) and apply it to the input image to get the convolved feature. This convolved feature is passed on to the next layer.

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

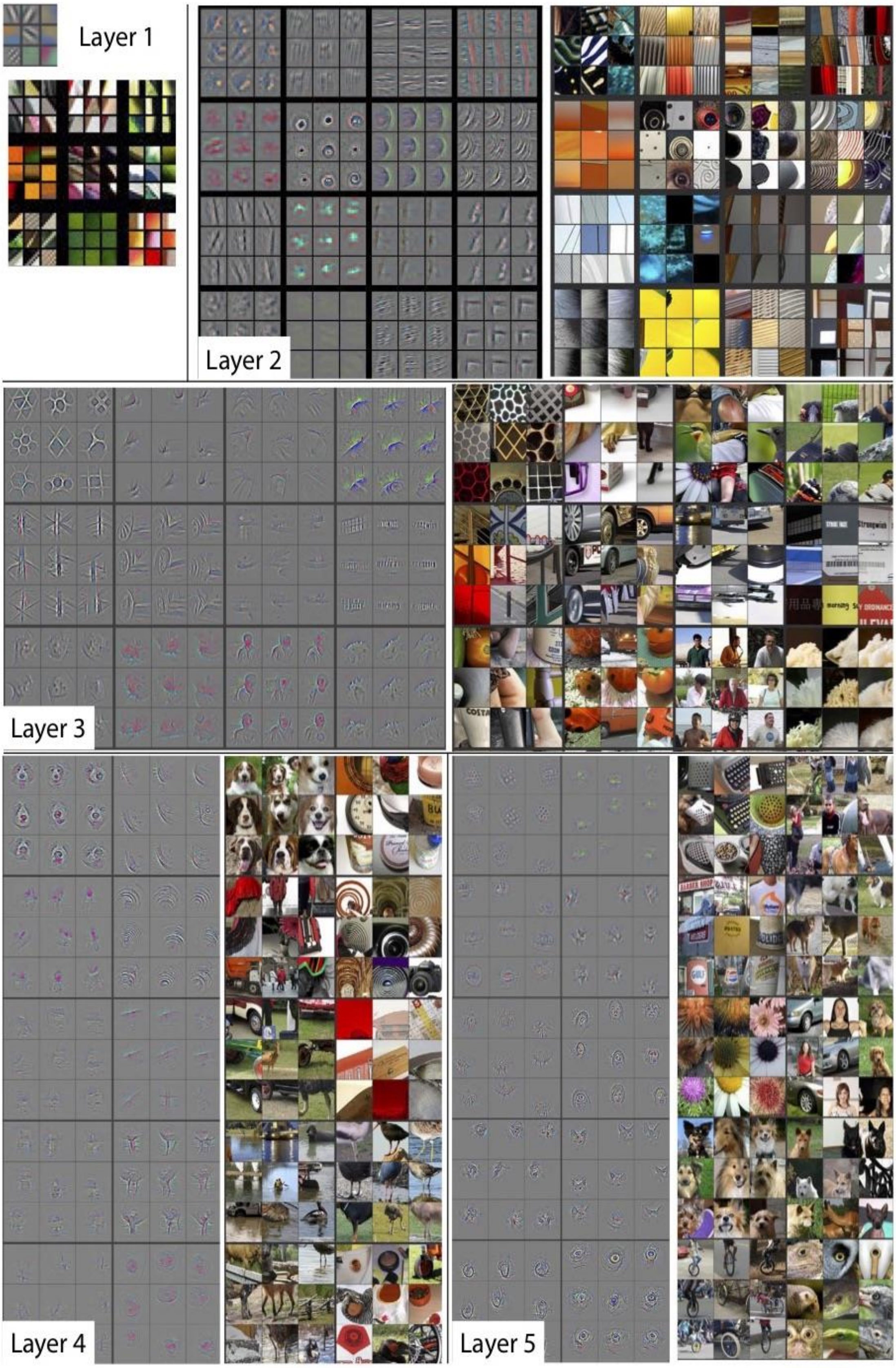


FIGURE 2. 4: CNN LAYERS

Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the image is to belong to a "class." For instance, if you have a ConvNet that detects cats, dogs, and horses, the output of the final layer is the possibility that the input image contains any of those animals.

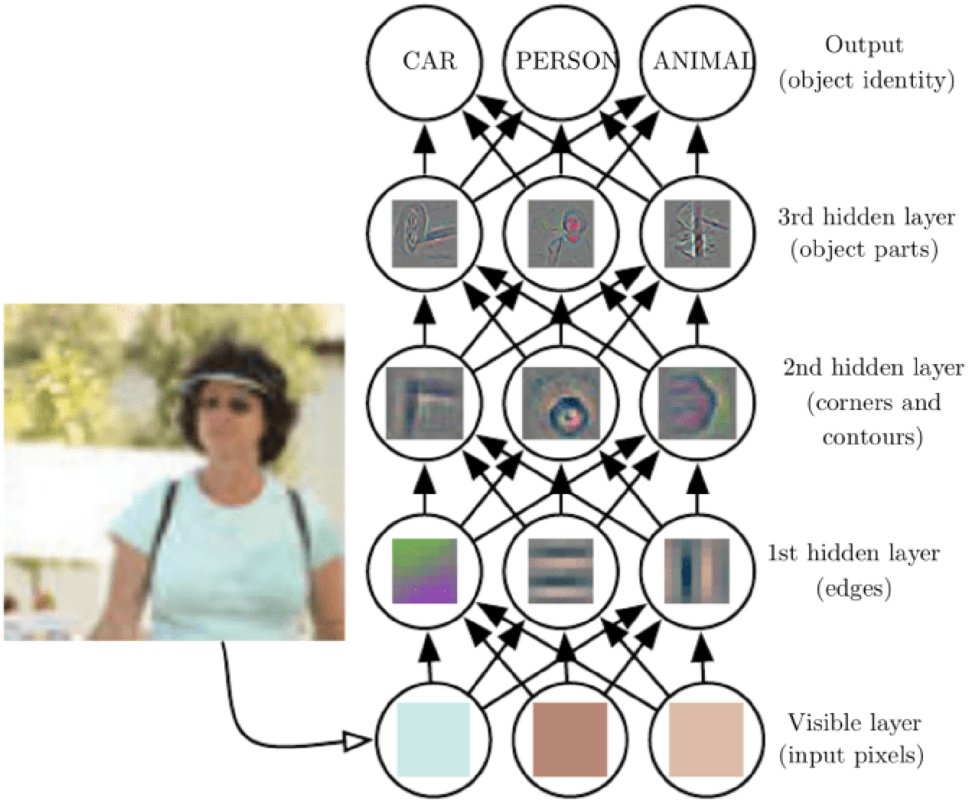
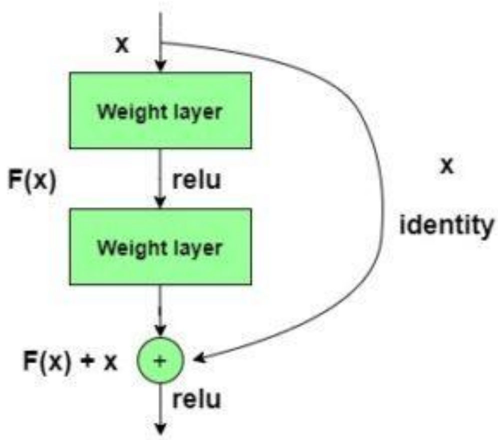


FIGURE 2. 5:DETECTION MAP

2.2 RESIDUAL NETWORKS

Recent years have seen tremendous progress in the field of Image Processing and Recognition. Deep Neural Networks are becoming deeper and more complex. It has been proved that adding more layers to a Neural Network can make it more robust for image-related tasks. But it can also cause them to lose accuracy. That's where Residual Networks come into place.

The tendency to add so many layers by deep learning practitioners is to extract important features from complex images. So, the first layers may detect edges, and the subsequent layers at the end may detect recognizable shapes, like tires of a car. But if we add more than *30* layers to the network, then its performance suffers, and it attains a low accuracy. This is contrary to the thinking that the addition of layers will make a neural network better. This is not due to overfitting, because in that case, one may use *dropout* and *regularization* techniques to solve the issue altogether. It's mainly present because of the popular **vanishing gradient** problem.

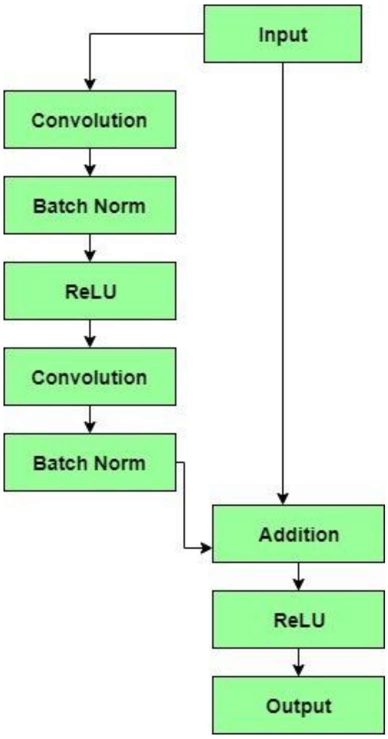


$y = F(x) + x$

The **ResNet152** model with 152 layers won the ILSVRC Imagenet 2015 test while having lesser parameters than the **VGG19** network, which was very popular at that time. A residual network consists of residual units or blocks which have *skip connections*, also called *identity connections*.

The skip connections are shown below:

The output of the previous layer is added to the output of the layer after it in the residual block. The hop or skip could be *1, 2 or even 3*. When adding, the dimensions of *x* may be different than *F(x)* due to the convolution process, resulting in a reduction of its dimensions. Thus, we add an additional *1 x 1* convolution layer to change the dimensions of *x*.

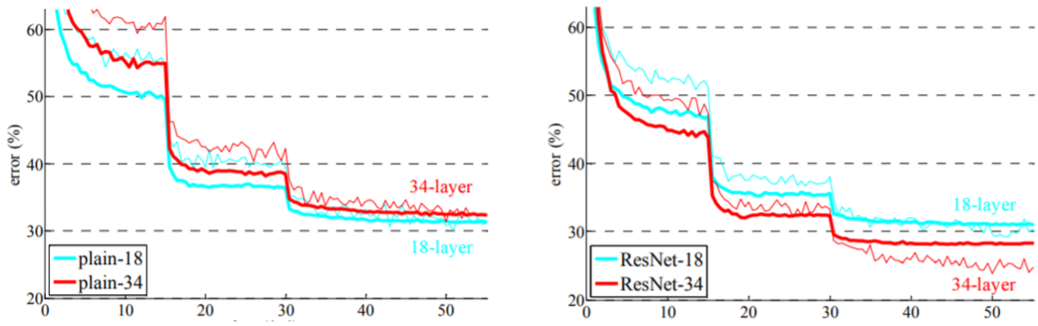


A residual block has a *3 x 3* convolution layer followed by a batch normalization layer and a ReLU activation function. This is again continued by a *3 x 3* convolution layer and a batch normalization layer. The skip connection basically skips both these layers and adds directly before the ReLU activation function. Such residual blocks are repeated to form a residual network.

After an in-depth comparison of all the present CNN architectures was done, the ResNet stood out by holding the lowest top 5% error rate at *3.57%* for classification tasks, overtaking all the other architectures. Even humans do not have much lower error rates. Comparison of *34* layer ResNet with VGG19 and a *34* layer plain network:

To conclude, it can be said that residual networks have become quite popular for image recognition and classification tasks because of their ability to solve vanishing and exploding gradients when adding more layers to an already deep neural network. A ResNet with thousand layers has not much practical use as of now.

The below graphs compare the accuracies of a plain network with that of a residual network. Note that with increasing layers a *34-layer* plain network's accuracy starts to saturate earlier than ResNet's accuracy.



2.3 GOOGLENET MODEL – CNN ARCHITECTURE

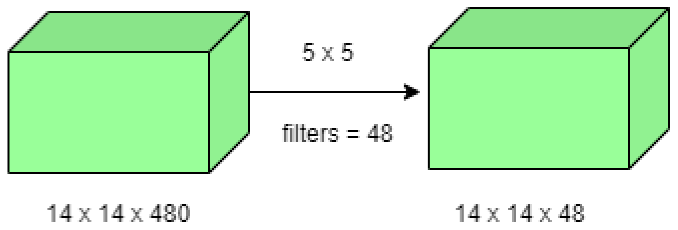
Google Net (or Inception V1) was proposed by research at Google (with the collaboration of various universities) in 2014 in the research paper titled "Going Deeper with Convolutions". This architecture was the winner at the ILSVRC 2014 image classification challenge. It has provided a significant decrease in error rate as compared to previous winners AlexNet (Winner of ILSVRC 2012) and ZF-Net (Winner of ILSVRC 2013) and significantly less error rate than VGG (2014 runner up). This architecture uses techniques such as *1×1* convolutions in the middle of the architecture and global average pooling.

Features of GoogleNet:

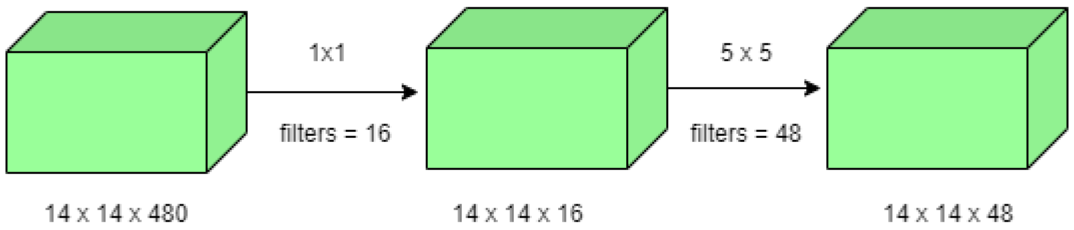
The GoogLeNet architecture is very different from previous state-of-the-art architectures such as AlexNet and ZF-Net. It uses many different kinds of methods such as *1×1* convolution and global average pooling that enables it to create deeper architecture. In the architecture, we will discuss some of these methods:

- ***1×1* convolution:** The inception architecture uses *1×1* convolution in its architecture. These convolutions used to decrease the number of parameters (weights and biases) of the architecture. By reducing the parameters, we also increase the depth of the architecture. Let's look at an example of a *1×1* convolution below:

    - For Example, If we want to perform *5×5* convolution having 48 filters without using *1×1* convolution as intermediate:



- Total Number of operations : *(14 x 14 x 48) x (5 x 5 x 480) = 112.9M*
    - With 1×1 convolution :



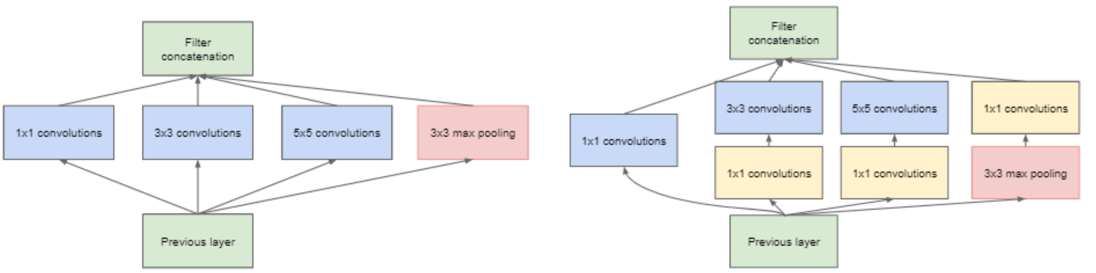
- *(14 x 14 x 16) x (1 x 1 x 480) + (14 x 14 x 48) x (5 x 5 x 16) = 1.5M + 3.8M = 5.3M* which is much smaller than 112.9M.

- **Global Average Pooling:** In the previous architecture such as AlexNet, the fully connected layers are used at the end of the network. These fully connected layers contain the majority of parameters of many architectures that causes an increase in computation cost.

  In GoogLeNet architecture, there is a method called global average pooling is used at the end of the network. This layer takes a feature map of *7×7* and averages it to *1×1*. This also decreases the number of trainable parameters to 0 and improves the top-1 accuracy by 0.6%

- **InceptionModule:** The inception module is different from previous architectures such as AlexNet, ZF-Net.

In this architecture, there is a fixed convolution size for each layer. In the Inception module *1×1, 3×3, 5×5* convolution and *3×3* max pooling performed in a parallel way at the input and the output of these are stacked together to generated final output. The idea behind that convolution filters of different sizes will handle objects at multiple scale better.



(a) Inception module, naïve version (b) Inception module with dimension reductions

FIGURE 2. 6:INCEPTION MODULE

- **Auxiliary Classifier for Training:** Inception architecture used some intermediate classifier branches in the middle of the architecture, these branches are used during training only. These branches consist of a 5×5 average pooling layer with a stride of 3, a *1×1* convolutions with *128* filters, two fully connected layers of 1024 outputs and 1000 outputs and a SoftMax classification layer. The generated loss of these layers added to total loss with a weight of 0.3. These layers help in combating gradient vanishing problem and provide regularization.

MODEL ARCHITECTURE:

Below is Layer by Layer architectural details of GoogLeNet.

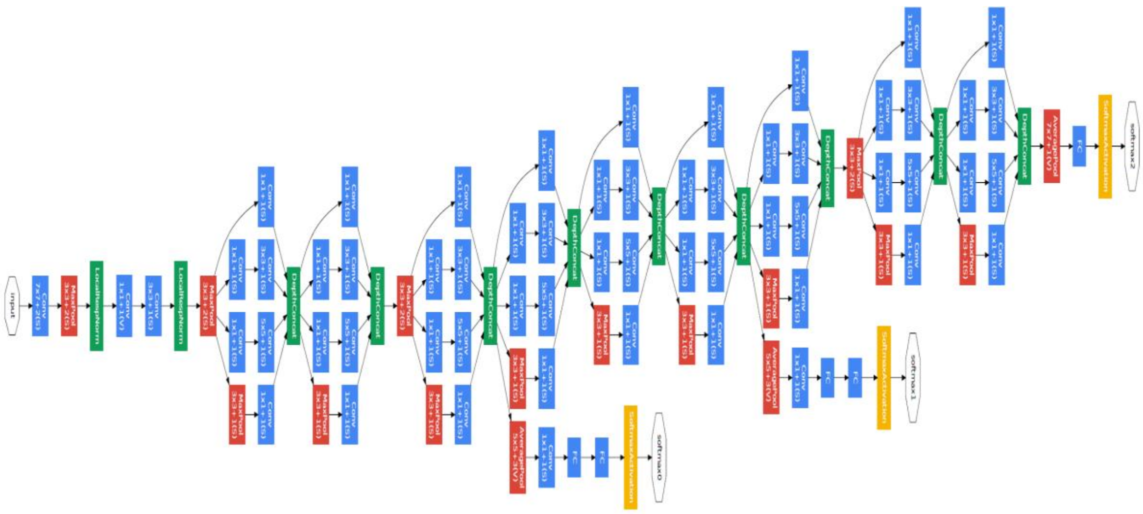
type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

FIGURE 2. 7:GOOGLE NET LAYERS

The overall architecture is 22 layers deep. The architecture was designed to keep computational efficiency in mind. The idea behind that the architecture can be run on individual devices even with low computational resources. The architecture also contains two auxiliary classifier layer connected to the output of Inception (4a) and Inception (4d) layers.

The architectural details of auxiliary classifiers as follows:

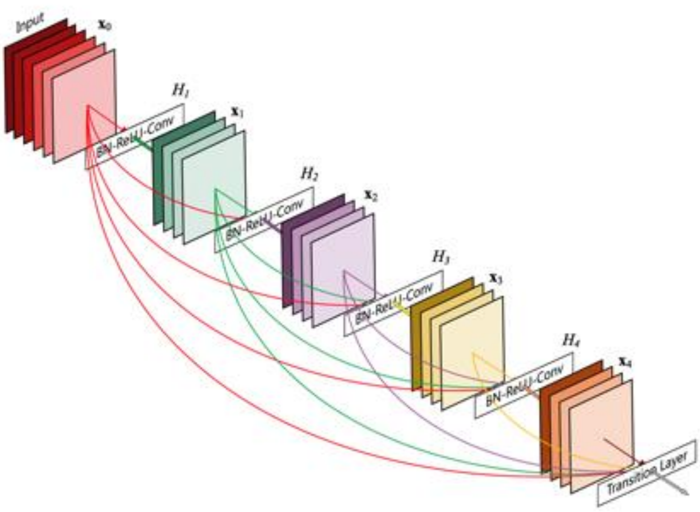
- An average pooling layer of filter size 5×5 and stride 3.
- A 1×1 convolution with 128 filters for dimension reduction and ReLU activation.
- A fully connected layer with 1025 outputs and ReLU activation
- Dropout Regularization with dropout ratio = 0.7
- A softmax classifier with 1000 classes output similar to the main softmax classifier.



This architecture takes image of size *224 x 224* with RGB colour channels. All the convolutions inside this architecture uses Rectified Linear Units (ReLU) as their activation functions.

2.4 DENSENET

The idea behind dense convolutional networks is simple: it may be useful to reference feature maps from earlier in the network. Thus, each layer's feature map is concatenated to the input of every successive layer within a dense block. This allows later layers within the network to directly leverage the features from earlier layers, encouraging feature reuse within the network. The authors state, "concatenating feature-maps learned by different layers increases variation in the input of subsequent layers and improves efficiency."



When I first came across this model, I figured that it would have an absurd number of parameters to support the dense connections between layers. However, because the network is capable of directly using any previous feature map, the authors found that they could work with very small output channel depths (ie. 12 filters per layer), vastly *reducing* the total number of parameters needed. The authors refer to the number of filters used in each convolutional layer as a "growth rate", kk, since each successive layer will have kk more channels than the last (as a result of accumulating and concatenating all previous layers to the input).

ARCHITECTURE

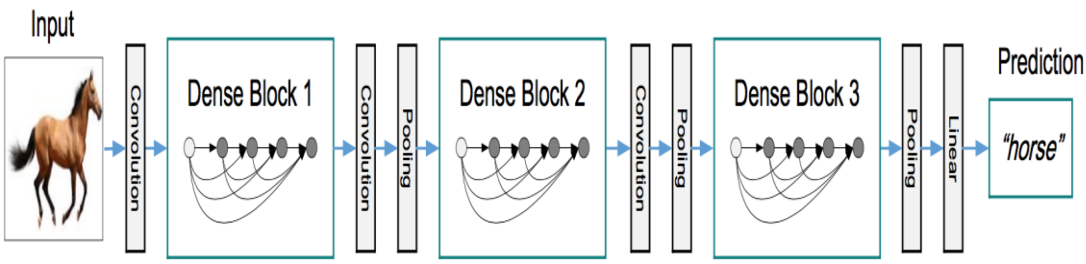


FIGURE 2. 8:ARCHITECTURE OF DENSENET

CHAPTER-3

3.1 COMPONENTS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

FUNCTIONAL REQUIREMENTS

- Graphical User interface with the User.

  The graphical user interface (GUI) is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

SOFTWARE REQUIREMENTS

For developing the application, the following are the Software Requirements:

1. Python 3.10:

    Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

    Python is dynamically-typed and garbage-collected. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.10 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 3.10 was discontinued with version 2.7.18 in 2020.

    Python consistently ranks as one of the most popular programming languages.

2. Cloud Deployment tools:

Software deployment tools make the process of distributing software and updates as easy as possible. Often, these tasks are automatic or scheduled to enable software developers to focus on what they do best – writing code. And the best tools work with a variety of platforms and types of infrastructures, making it easy to streamline your workflow in your preferred environment.

Software deployment tools also allow developers to collaborate on their projects, track progress, and manage changes.

- AWS CloudFormation.
- Puppet.
- Ansible.
- Chef.
- Kubernetes
- Terraform.
- Google Cloud Deployment Manager.
- Microsoft Azure Automation.

3. Tensorflow:

   TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

4. OpenCV

   OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel[2]). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

HARDWARE REQUIREMENTS

For developing the application the following are the Hardware Requirements:

- Processor: Intel i3
- RAM: 4 GB
- Space on Hard Disk: minimum 1 TB

INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1) Input Design is the process of converting a user-oriented description of the input into a computer-based system. This

design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2) It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3) When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1) Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2) Select methods for presenting information.
3) Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

3.2 SYSTEM ARCHITECTURE

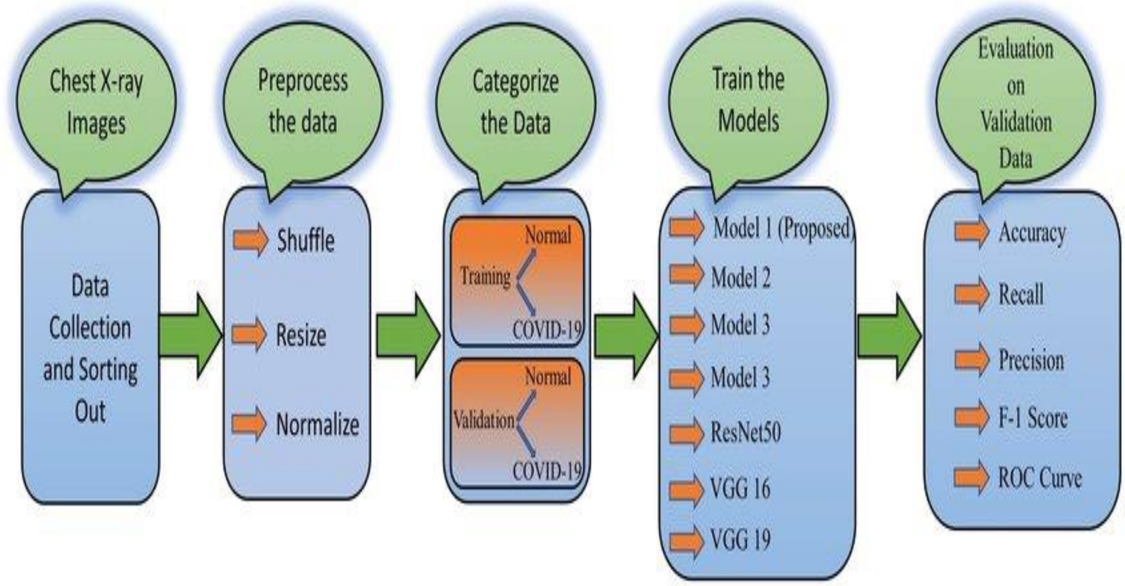


FIGURE 3. 1:SYSTEM ARCHITECTURE

Deep Learning has improved multi-fold in recent years and it has been playing a great role in image classification which also includes medical imaging. Convolutional Neural Networks (CNNs) have been performing well in detecting many diseases including coronary artery disease, malaria, Alzheimer's disease, different dental diseases, and Parkinson's disease. Like other cases, CNN has a substantial prospect in detecting COVID-19 patients with medical images like chest X-rays and CTs. Coronavirus or COVID-19 has been declared a global pandemic by the World Health Organization (WHO). As of 8 August 2020, the total COVID-19 confirmed cases are 19.18 M and deaths are 0.716 M worldwide. Detecting Coronavirus positive patients is very important in preventing the spread of this virus. On this conquest, a CNN model is proposed to detect COVID-19 patients from chest X-ray images. Two more CNN models with different number of convolution layers and three other models based on pretrained ResNet50, VGG-16 and VGG-19 are evaluated with comparative analytical analysis. All six models are trained and validated with Dataset 1 and Dataset 2. Dataset 1 has 201 normal and 201 COVID-19 chest X-rays whereas Dataset 2 is comparatively larger with 659 normal and 295 COVID-19 chest X-ray images. The proposed model performs with an accuracy of 98.3% and a precision of 96.72% with Dataset 2. This model gives the Receiver Operating Characteristic (ROC) curve area of 0.983 and F1-score of 98.3 with Dataset 2. Moreover, this work shows a comparative analysis of how change in convolutional layers and increase in dataset affect classifying performances.

CHAPTER-4

4.1 HISTORY

AI algorithms moved from heuristics-based techniques to manual, handcrafted feature extraction techniques. and then to supervised learning techniques. Unsupervised machine learning methods are also being researched, but most of the algorithms from 2015-2017 in the published literature have employed supervised learning methods, namely Convolutional Neural Networks (CNN). Aside from the availability of large, labelled data sets being available, hardware advancements in Graphical Processing Units (GPUs) have also led to improvements in CNN performance, and their widespread use in medical image analysis. McCulloch and Pitts described the first artificial neuron in 1943, which developed into the perceptron posited by Rosenblatt in 1958. In essence, an artificial neural network is a layer of connected perceptron linking inputs and outputs, and deep neural networks are multiple layers of artificial neural networks. The advantage of a deep neural network is its ability to automatically learn significant low-level features (such as lines or edges) and amalgamate them to higher level features (such as shapes) in the subsequent layers. Interestingly, this is how the mammalian and human visual cortices are thought to process visual information and recognize objects. CNNs may have their origins in the Noncognition concept proposed by Fukushima in 1982, but it was

Lecun who formalized CNNs and used the error backpropagation described by Rumelhart *et al* to successfully perform the automatic recognition of handwritten digits. The widespread use of CNNs in image recognition came about after Krizhevsky *et al.* the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a CNN that had a 15% error rate. CNNs are the most popular machine learning algorithm in image recognition and visual learning tasks, due to its unique characteristic of preserving local image relations, while performing dimensionality reduction. This captures important feature relationships in an image (such as how pixels on an edge join to form a line) and reduces the number of parameters the algorithm must compute, increasing computational efficiency. CNNs can take as inputs and process both 2-dimensional images, as well as 3-dimensional images with minor modifications. This is a useful advantage in designing a system for hospital use, as some modalities like X-rays are 2-dimensional while others like CT or MRI scans are 3-dimensional volumes.

4.2 REFERENCES

1. Worldometers Website. [Online]. Available: https://www.worldometers. info/coronavirus/ 2021.
2. West, C. P., Montori, V. M., & Sampath Kumar, P. (2020). "COVID19 Testing: The Threat of False-Negative Results," 95(6), 1127-1129. https://doi.org/10.1016/ j. mayocp.2020.04.004.
3. M. E. H. Chowdhury et al., "Can AI Help in Screening Viral and COVID-19 Pneumonia?" IEEE Access, Vol. 8, pp. 132665-132676, 2020, doi: 10.1109/ACCESS.2020.3010287.
4. Y. Li, X. Zhang and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 1091-1100, doi: 10.1109/CVPR.2018.00120.
5. Yasin, R., Gouda, W.," Chest X-ray findings monitoring COVID-19 disease course and severity," Egypt J Radiol Nucl Med 51, 193 (2020). https://doi.org/10.1186/s43055-020-00296-x.
6. Xu X, Yu C, Qu J, Zhang L, Jiang S, Huang D, Chen B, Zhang Z, Guan W, Ling Z, Jiang R, Hu T, Ding Y, Lin L, Gan Q, Luo L, Tang X, Liu J., "Imaging and clinical features of patients with 2019 novel coronavirus SARS-CoV-2," Eur J Nucl Med Mol Imaging, 47(5):1275- 1280, 2020.
7. K. Zuiderveld, "Contrast limited adaptive histogram equalization," in Graphics Gems IV. New York, NY, USA: Academic, 1994.