

How to Cite:

Afreen, Y. N., & Reddy, P. B. (2022). CDLX: An Efficient novel approach for COVID detection lung x-rays throw transfer learning based on State of the art deep learning image classification models. *International Journal of Health Sciences*, 6(S5), 8439–8463.
<https://doi.org/10.53730/ijhs.v6nS5.10593>

CDLX: An Efficient novel approach for COVID detection lung x-rays throw transfer learning based on State of the art deep learning image classification models

Y Nuzhat Afreen

Research Scholar, School of Computer Science and Engineering, REVA University, Bangalore, India.

Corresponding author email: Afreenphd2020@gmail.com

P.V. Bhaskar Reddy

Professor, School of Computer Science and Engineering, REVA University, Bangalore, India.

Email: bhaskarreddy.pv@reva.edu.in

Abstract---When compared to the general population, lung cancer patients have a higher incidence of COVID-19 infection, pulmonary problems, and poorer survival results. As a reference for prioritising cancer care issues during the epidemic, the world's main professional organisations issued new recommendations for the diagnosis, treatment, and follow-up of lung cancer patients. In today's world, we are fighting one of the greatest pandemics in human history, known as COVID-2019, which is caused by a coronavirus. The patient can be treated promptly if the infection is detected early (before it enters the lower respiratory tract). To observe ground-glass opacity in the chest X-ray due to fibrosis in the lungs once the virus has reached the lungs. Artificial intelligence techniques can be utilised to detect the presence and degree of illness based on the major discrepancies between X-ray images of an infected and non-infected person. For this study, I employed feature extraction from Transfer Learning, which entails importing a pre-trained CNN model, such as Distributed Deep Convolutional VGGNet or Distributed Deep Convolutional with ResNet Model, and changing the last layer to meet my needs. Using Distributed Deep Convolutinal VGGNet, the model can get an F1-score of 0.88, which is the best among all pretrained models. Furthermore, the X-rays suggesting COVID-19 are divided into three categories based on severity: mild, medium, and severe. Because precision and recall are both essential factors in this study, the data are analysed using the F1-Score. To provide a thorough study of the Model

performance, the confusion matrix and results for the F1-Score, Precision, Recall, and total Accuracy are also presented. As a warning to society, the proposed approaches have had a considerable impact upon the country.

Keywords---Transger Learning, Resnet50, VGG16 COVID-19, x-rays, ML, DL, CT hadoop.

1 Introduction

Because COVID-19 data and study are still in their early stages, much remains unclear, particularly in terms of recovery, immunity, transmission, and linkages to other pre-existing illnesses. Experts do know that some people are more susceptible to catching a more severe strain of the virus. Due to the lack of a formal study linking the coronavirus to lung cancer, doctors can only make educated guesses regarding the best course of action. There is yet no conclusive evidence that the novel coronavirus causes lung cancer. In comparison to the general population, lung cancer patients had roughly twice the chance of COVID-19 infection, according to an early Wuhan study.

It's unclear whether this is because these patients, like many cancer patients, were immunocompromised as a result of their therapy. There may be an increased risk of COVID-19 in patients with lung cancer, according to Dr. Adil Akhtar, an expert in oncology and palliative care and an associate professor of medical oncology and haematology at the University of Auckland-William Beaumont School of Medicine. According to a Chinese study, lung cancer is the most frequent cancer.

"COVID-19 disproportionately impacts fragile people, especially the elderly, and persons with comorbidities, including immunocompromised people like cancer patients," according to Dr. Wasif M. Saif, head physician and medical director of Northwell Health Cancer Institute. "While lung cancer does not increase the risk of coronavirus complications, many lung cancer patients are older, have severe lung disease, and have decreased lung capacity," said Dr. Brendan Stiles, thoracic surgeon at New York Presbyterian Hospital and Associate Professor of Cardiothoracic Surgery at Weill Cornell Medicine. "If the coronavirus causes problems, these patients will almost probably acquire lung illness."

It's also probable that people with lung cancer who are immunocompromised by chemotherapy are more prone to contract the coronavirus and develop serious problems as a result of infection. Lung illnesses, often known as respiratory disorders, affect the respiratory tract as well as other organs in the lungs [1]. Lung diseases include pneumonia, TB, and coronavirus infections (COVID-19). According to the International Respiratory Society Forum, 334 million people have asthma, 1.4 million people die of tuberculosis each year, 1.6 million people die of lung cancer, and 1.6 million people die of pneumonia. Thousands have died as a result of the violence. COVID-19 has infected people all across the world [3], infecting millions of people and putting a pressure on health-care systems[4].

Lung illness is one of the world's most common causes of mortality and disability. For greater long-term survival and recovery, early detection is critical [5, 6]. Skin tests, blood tests, sputum samples [7], chest X-rays, and computed tomography (CT) [8] were once used to detect lung problems. In-depth training has recently showed promise in medical imaging for diagnosing disorders, particularly lung diseases. Deep learning is a branch of machine learning that focuses on processes inspired by the brain's structure and function. Medical visual patterns can now be identified, quantified, and classified more easily because to recent breakthroughs in machine learning, particularly deep learning [9].

In-depth training has enabled this advancement, allowing staff to learn features from data alone rather than manually developed characteristics based on unique field knowledge. Deep training is rapidly gaining traction as the gold standard for improving performance in a range of medical situations. As a result, these advancements aid clinicians in correctly diagnosing and classifying disorders [10] [11].

2 Related Works

“Because it allows for the creation of accurate models, transfer learning has become a popular technique in computer vision [12]”. With transfer learning, a model that has been learned in one domain can be applied to another. “Transfer learning can be done with or without the use of a previously acquired model [13]”. Many CNN architectures are trained using ImageNet [14]. “The photos were gathered from the internet and labelled by humans using Amazon's Mechanical Turk crowd-sourcing technique. ILSVRC uses a subset of ImageNet, which has approximately 1000 images in each of 1000 classes”.

There are 1.2 million training photos, 50,000 validation shots, and 150,000 testing photos available. [15]. “The weights of the model's upper layers are only re-trained, while the weights of the model's lower levels are usually frozen. [16]”. The practise of merging multiple classifiers to make a prediction is known as ensemble classification. Ensemble forecasting minimises prediction variance, resulting in more accurate forecasts than any single model. Popular polling, probability score averaging, and stacking are the ensemble methodologies used, according to the literature. [17].

COVID-19 has thirteen datasets that are publicly available in this section. Several datasets have been made public as a result of the COVID-19 outbreak. Several of these collections continue to grow in terms of the quantity of photos they hold. As a result, the number of pictures in the datasets may differ from what is stated in this paper [31-32]. It should be noted that some of the photographs can be found in a variety of databases. As a result, future studies should look for duplicate pictures [18].

The Covid is a new and exceedingly contagious respiratory disease caused by the severe acute respiratory syndrome coronavirus. The current COVID-19 virus has

spread over the world. As of June 7, 2020, there are 6,663,304 confirmed cases and 392,802 confirmed deaths worldwide. It spreads quickly, and some people succumb to respiratory failure before the disease has fully developed. The clinical symptoms, aetiology, and even treatment of COVID-19 are all dependent on research and analysis of the disease's critical phase. In China, the vast majority of patients have been successfully discharged. There have been no studies to far that have provided an early prognosis of the extent of lung injury and rehabilitation in COVID-19 patients.

According to a retrospective research, the chicken patients were discharged, many had radiological problems, and some even had lung fibrosis. In COVID-19 patients in the early phases of recovery, low lung function demands additional monitoring. A retrospective investigation of 57 approved but recovered COVID-19 patients was conducted to have a better understanding of the expected clinical results. Thirty days after discharge, serial lung function, lung imaging, and cardiovascular capacity were assessed. We also used performance measures to compare critically ill and non-critically ill persons.

This section describes the methodology used in a recent study that used in-depth training to diagnose lung illness. A block diagram is used to depict the procedure. The primary source of article links is cited for the first time as a suitable database. Scopus was chosen since it is one of the most well-known databases of peer-reviewed scientific literature. Many major publications are included, based on the number of citations received, and are indexed by Google Scholar but not by Scopus. Several pre-printed articles about COVID-19 have been given due to the recent epidemic of the disease.

"This study only includes the most recently publicised positions (2016-2020) to ensure that it only includes the most recent jobs". There are, however, some older but crucial components. All feasible documents for the detection of deep lung illnesses were found using relevant search terms. Among the terms used were "in-depth training," "screening," "classification," "CNN," "lung sickness," "tuberculosis," "pneumonia," "lung cancer," "COVID-19," and "coronavirus." The survey only includes English-language articles. At the end of this phase, we had found 366 articles. Second, for selection, only relevant works were reviewed. Only the right and abstract were examined throughout the screening.

The main assortment criterion for this study is papers using in-depth training methods to detect relevant disorders. Inappropriate members were expelled. Only 98 products are included in the short list, according to the inspection. The primary purpose of diagnosing lung illness is to divide a picture into two categories: healthy lungs and diseased lungs. To obtain the lung disease classifier, also known as a model, training is required. The process of training a neural network to recognise a set of photos is known as learning. Take in-depth training to learn how to build a model that can classify obsessed pictures according to their class labels.

As a result, the first stage in diagnosing lung disease using in-depth training is to image the lungs with the disease to be detected. The neural network is then trained until it can recognise the ailment in the second step. The final step is to

organise the new photographs into categories. When it receives fresh images that it has never seen before, the prototype estimates the class of photographs of individuals. Figure 1 depicts a high-level summary of the technique. The study was limited according to the explanation given in section 2, as the main source of work studied is people indexed in the Scopus database.

COVID-19-related posts were allowed as an exception, as most comments made it past the prepress stage at the time of this study. “In terms of publishing years, the most current papers that have been examined are those that were published before October 2020. Something is being done as a result of the conclusions reported in this evaluation report[19]”.

3. Methods and Materials

3.1 Dataset

The data sets used in the measured research are described in this section. The data sets used to diagnose tuberculosis, pneumonia, lung cancer, and COVID-19 are summarised. This is done to give readers with useful information about the data sets. Tables only include public data sets that may be viewed by everyone, whereas private data sets require authorization to view. Deep machine learning approaches that employ X-rays to diagnose COVID-19 pneumonia have recently been described by several organisations. However, the majority of these studies relied on a tiny data set with only a few COVID-19 samples. This makes summarising the results provided in these studies challenging, and there is no guarantee that the reported performance will be maintained when these models are tested on a bigger data set [13].

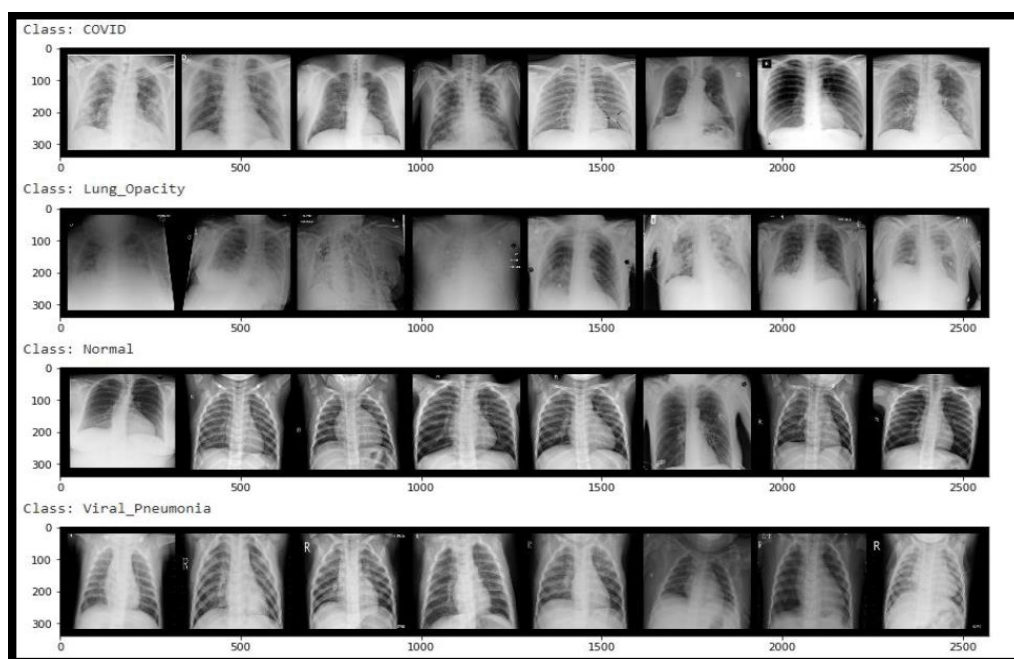


Fig: 1 Sample data set

COVID-19 comprises thirteen publicly available data sets, as indicated in Table 8. Several data sets were made available as a result of the COVID-19 outbreak. In terms of the amount of images in each of these collections, several of them are still growing. As a result, the number of pictures in the data sets may differ from what is stated in this text. It's worth noting that some of the photographs are available in a variety of databases. As a result, in the future, researchers will need to check for duplicate photographs. Summarize the research you've done using taxonomy. Customers can quickly obtain content that is relevant to their interests as a result of this. The next section looks at the distribution of functions based on taxonomic features that have been identified. There are a total of 21,165 samples, which are divided into four categories: COVID-19, opacity of the lungs, normal, and viral pneumonia are all examples of viral pneumonia. All of the photographs are saved as Portable Network Graphics (PNG) files and are 299x299 pixels in size. According to the most recent update, the Hadoop data security ecosystem now comprises 3,616 positive COVID-19 cases, 10,192 normal, 6,012 lung opacity (non-COVID lung infection), and 1,345 viral pneumonia images.

3.2 System Architecture

As illustrated in Figure 1, the input is a CXR image of a dataset with two subsections: COVID-19 patients and normal patients. Before training the model, this system underwent preprocessing, which included loading photos of a specified size, splitting the dataset, and data augmentation techniques. After the model was installed and fine-tuned, the accuracy improved. To demonstrate how loss and accuracy varied over time, the confusion matrix, model loss, and model accuracy were displayed. Finally, if a user offers a photo as an input model, we can predict if the image in the output component is of a COVID-19 patient.

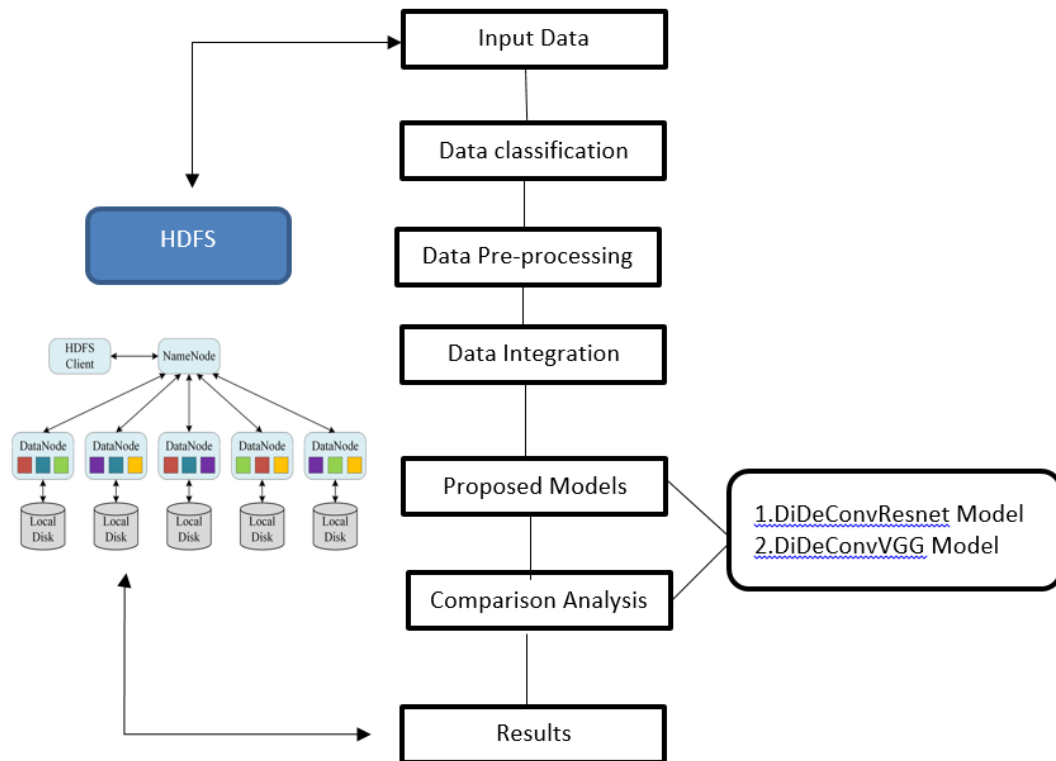


Fig 2: System Design

The block diagram depicts the entire system in the most simple manner possible. The decision-making component of this system is vital, and it plays an important part in this study. The decision is based on a model that was developed using a large amount of image data and stored in the Hadoop ecosystem for managing and controlling data transmission between nodes via name node, data node, and resource manager.

3.3 Methods

3.1 Convolutional Neural Networks (CNN)

The suggested method uses deep learning techniques based on a Hadoop Distributed Deep Convolutional Neural Network (HdiDConvNNet) to solve the problem of early lung diagnosis identification. The goal of our study is to apply a variety of distributed deep learning methods to identify between COVID-19 and non-COVID 19 CT scan pictures. Clinicians can employ automated COVID-19 diagnostics based on CT scan images to detect COVID -19 in a quick and efficient manner. Because precision and recall are both essential factors in this study, the data are analysed using the F1-Score. To provide a thorough study of the Model performance, the confusion matrix and results for the F1-Score, Precision, Recall, and total Accuracy are also presented. Hadoop Distributed Deep Convolutional Neural Network (HdiDConvNNet), which is a type of deep neural networks (DNN),

is widely used in image classification in Hadoop ecosystem and was originally presented by Fukushima in 1988 [8]. Due to computing hardware restriction. Feature extraction, one of the most important aspects of the networks, can be done effectively in CNN. Furthermore, the max pooling layer of CNN is powerful in reducing the dimension of the image without affecting the feature [8]. Firstly, there are three main types of layers in HdiDConvNNet, which are *convolutional layer*, *max pooling layer* and *fully connected layer*. Secondly, layers inside the network relate to each other and use the output of the previous as its input so as neuron behavior. Furthermore, the main task for the first two layers, convolutional layer, and max pooling layer, is to achieve feature extraction, while the last layer is considered as classification which normally has one or more fully connected layer. In particular, the last fully connected layer maps the features into an actual object with probability output. Lastly, like AlexNet, a classic CNN consists of blocks of layers, where each block contains several sub-layers such as convolutional layer, ReLU layer or max pooling layer.

Input Layer

In a general CNN, like AlexNet, the input to the first layer is not the image itself, but a so-called *channel*. *Channel* refers to a certain component of an image. For example, an RGB image has three channels red, green, and blue. There are $F_0 = 3$ channels to the first layer, where each channel corresponds to each of the RGB colors. Each of the RGB channel, together, forms the first input layer.

Convolutional Layer

Convolutional layer, one of the most important layers in CNN, provides the essential functionality to feature extraction. Like its name the operation convolutes a feature of an input layer with two learnable parameters, weight, and bias, to generate the output *feature map*. The operation is done by a mathematical linear operation, called convolution product. Let us recall the convolution of two integrable functions f and g , denoted as $f * g$, is defined as the following,

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(r)g(t-r)dr,$$

where t is an input variable in $\mathbb{R}$.

To obtain the most appropriate output *feature map*, the value of the weight and bias are what need to be discovered in the later optimization process. *Feature map* refers to the output feature for one given *filter* (see below). First, all the variables are under real number field ($\mathbb{R}$), and we define some of the important terminologies below. *Filter* (or *kernel*) is a three-dimensional matrix of size $K \times K \times F$, where $K \times K$ refers to the single aspect of a channel and F refers to the number of channels. The channel is also known as the *depth of the image* since the number of channels in a filter is equal to the number of channels for the input image. The *depth of the filters* is the number of filters, which also refers to the number of output feature maps from the convolutional operation. For example, in the first convolutional layer of the AlexNet, given the $227 \times 227 \times 3$ input image,

the filter is of size $11 \times 11 \times 3$ with the depth of the filters equal to 96. Note that the number of channels in the filter is equal to the number of channels for the input image.

Then, “the convolutional layer produces 96 output feature maps ($55 \times 55 \times 96$). The output dimension is computed via Equation in below”. It is worth mentioning that the convolutional operation is a combination of four summations. From inner to outer, the first two sums over one channel of a filter ($K \times K$), the third one sums over the *channel* (F), and the last one sums over the *depth of the filters* (D). Regarding how the operation applies to an image, we first consider $F = 1$, $D = 1$, and start from an image of size 5×5 and a filter (or kernel) of size 3×3 . The convolution is computed by sliding the filter over the image, starting from the top left corner. Then, the first value of the output matrix is obtained by summing the product of the corresponding pixel value and the filter value. Similarly, the final output is computed by moving the filter from left to right and from top to bottom.

I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}
I_{31}	I_{32}	I_{33}	I_{34}	I_{35}
I_{41}	I_{42}	I_{43}	I_{44}	I_{45}
I_{51}	I_{52}	I_{53}	I_{54}	I_{55}

K_{11}	K_{12}	K_{13}
K_{21}	K_{22}	K_{23}
K_{31}	K_{32}	K_{33}

Here, “Demonstration of input map and a filter: A 5×5 images with a single channel on the left and a 3×3 filter on the right. Note that each value from the image or filter is a pixel value. In other words, we can define the operation mathematically for one single channel is “

$$O(i, j) = \sum_{k=1}^K \sum_{l=1}^K I(i + k - 1, j + l - 1) K(k, l)$$

where , “ $K \times K$ is the size of the filter, O denotes output, and $O(i, j)$ is the value of the i^{th} row and j^{th} “column of the output matrix. Furthermore, i is running from 1 to $W - K + 1$ and j is running from 1 to $H - K + 1$ where $W \times H$ is the size of the image [7]. For example, the $O(1,1)$ value from the above example is “

$$\begin{aligned} O(1,1) &= \sum_{k=1}^3 \sum_{l=1}^3 I(1 + k - 1, 1 + l - 1) K(k, l) \\ &= I_{11} K_{11} + I_{12} K_{12} + I_{13} K_{13} + I_{21} K_{21} + I_{22} K_{22} \\ &+ I_{23} K_{23} + I_{31} K_{31} + I_{32} K_{32} + I_{33} K_{33} \end{aligned}$$

where , we have set “ $I_{ij} = I(i, j)$ for $1 \leq i, j \leq 5$ and $K_{kl} = K(k, l)$ for $1 \leq k, l \leq 3$ “ .

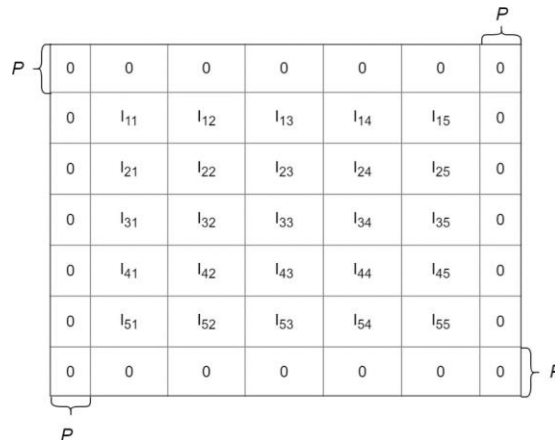
As a result, a 4×4 matrix is produced from the above example. Now, if we consider a normal image with RGB colors, which means $F = 3$, $D = 1$, the final matrix is formed by summing all the corresponding value from each of the channel output matrices. The result can be defined as

$$O'(i, j) = \sum_{v=1}^F O_v(i, j)$$

where O_v denotes the output matrix from the v^{th} channels, F is the number of channels and O' is the output feature map.

Padding and Stride

As we can notice, the outer pixel value of an image is only involved once in one of the operations, whereas the inner pixel value of an image is involved multiple times. Moreover, compared to the original image, the dimension of the output feature map has been reduced. For example, in the above example, the original image has size 5×5 , while the output has size only 4×4 . Padding (P) is added around the images with value 0 to preserve the width and height of the original image. If the original input has size $W \times H$, then, after adding the padding, the output has a size

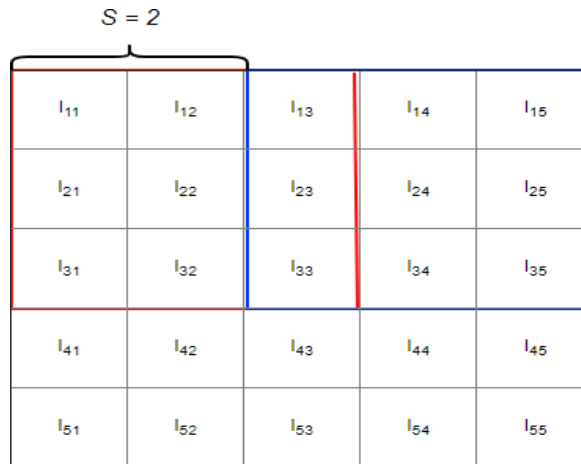


$$(P + W + P) \times (P + H + P)$$

where each P corresponds to the padding size in each direction of the input image. The formula is simplified as

$$(W + 2P) \times (H + 2P).$$

the Demonstration of padding: A matrix (Or image) with padding equal to 1 ($P = 1$). The zeros are added once to each direction of the image. Another important term to be introduced here is called *stride* (S), which is the distance between the movement of the filter. In general, the stride value for the convolutional layer is 1 and it is widely used in the max pooling layer.



The Demonstration of stride: The graph represents a movement of a filter with stride ($S = 2$) from the first position (red area) to the next position (blue area). In conclusion, if the size of the filter is $K \times K$, then the output width (W_o) and height (H_o) are

$$W_o = \text{Div}(W + 2P - K, S) + 1, H_o = \text{Div}(H + 2P - K, S) + 1$$

Also, in order to preserve the width and the height of the original image, common padding is

$$P = \text{Div}(K - 1, 2)$$

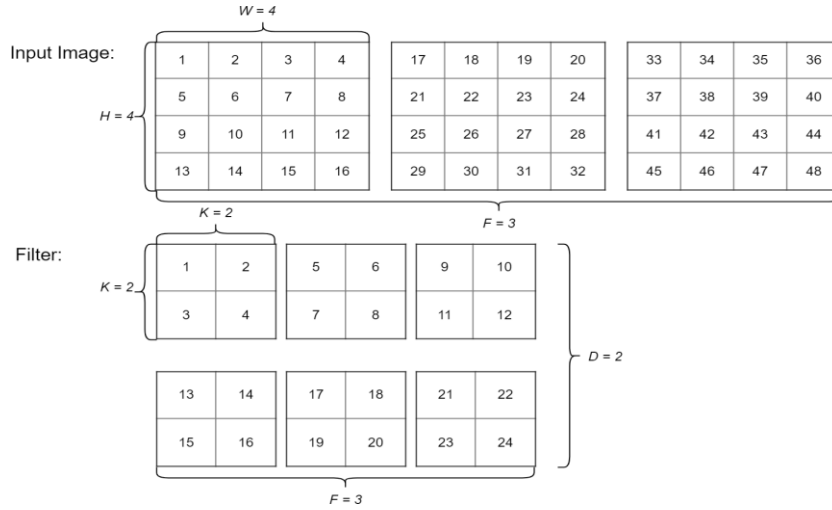
where K is the dimension of the filter. “a general convolution operation applied to a real image with size 32×32 , a filter of size 3×3 , no padding and 1 be the stride value. In realistic, there is more than one filter applied to the input to extract more feature. For example, in AlexNet, the first convolution layer applies 96 filters with size $11 \times 11 \times 3$ and a stride of 4 to the input image with size $227 \times 227 \times 3$, where the 3 represents the RGB colors”.

Parameters

In general, the convolutional operation is a matrix multiplication which can be expressed as a single neuron function without an activation function

$$f(x) = W'x + b$$

where $x \in \mathbb{R}^n$, $f(x) \in \mathbb{R}^m$, $W' \in \mathbb{R}^{m \times n}$ is a weight matrix and $b \in \mathbb{R}^m$ is a bias. The high dimensional filter is reshaped into a 2D weight matrix so as the image input. A simple example is used here to demonstrate the intuition behind it. “Consider an image with the following settings, $W = 4$, $H = 4$, $K = 2$, $F = 3$, $D = 2$, and $S = 1$. In other words, the input RGB image is of size $4 \times 4 \times 3$ ($W \times H \times F$), the filter is of size $2 \times 2 \times 3$ ($K \times K \times F$) with the depth of the filters (D) equal to 2, and the stride (S) is 1”.



Here , “Demonstration of input image and filter: The input image has size $4 \times 4 \times 3$, and the filter has size $2 \times 2 \times 3$ with the depth of the filter equal to 2. The total number of values from all the learnable parameters is a critical measure of the complexity of a CNN . The number of values in this learnable parameter, W' , is “.

$$\text{Number of values} = (K \times K \times F + 1) \times D$$

where K is the size of the individual channel, F is the number of channels, D is the depth of the filters and 1 refers to a single bias for the current filter [9]. For example, the first convolution layer in AlexNet contains

$$\text{Number of values} = (11 \times 11 \times 3 + 1) \times 96 = 34944.$$

Moreover, “AlexNet architecture has around 60 million values to be determined at the end of the networks, which is trained in two GPUs”.

Pooling Layer

The pooling operation, or sub-sampling layer, integrates a down sampled operation to the input feature map, which is also called dimension reduction operation . It is worth pointing out that the pooling operation does not require to train more parameters. In general, there are two kinds of pooling operations, one is called average pooling, and the other one is called max pooling. Similar to convolutional layer, the pooling layer also involves hyperparameters including filter size (R), stride (S) and padding (P). Max pooling, one of the most populate types of pooling operation, is essentially a max function, which filters the input map and takes the maximum value from the filtered region. The operation is defined by

$$O(i, j) = \max_{i \leq l \leq i+R-1, j \leq k \leq j+R-1} \{I(l, k)\}$$

where $O(i, j)$ is the value of the i^{th} row and j^{th} column of the output matrix. As an example (Figure 2.9), with a filter size equal to 2 ($R = 2$) and stride equal to 1 ($S = 1$), an output feature matrix of size 4×4 is computed by taking the maximum value inside the filtered space. The first value is computed by

$$O_{11} = \max \{I_{11}, I_{12}, I_{21}, I_{22}\}$$

where we have set $O_{ij} = O(i, j)$ and $I_{lk} = I(l, k)$. Compared to the input map, the dimension of the output map is reduced by 1. Demonstration of max pooling operation: A sample max pooling operation of an input feature map of size 5×5 with filter size ($R = 2$), stride ($S = 1$) and no padding resulted in an output of size 4×4 .

Activation Function (ReLU)

Rectified linear unit (ReLU), a nonlinear activation function, acts mathematically as a biological neuron [4]. Although the traditional activation functions, such as sigmoid or hyperbolic tangent (tanh), also perform the same task, it has been observed that a CNN with ReLU as an activation function trains much faster than the network with hyperbolic tangent as its. Furthermore, ReLU is a very simple function, which is defined by

$$f(x) = \max(0, x)$$

where the function is linear for $x > 0$ and 0 when x is negative (Figure 2.10). In other words, the function changes all the negative value with 0 and the positive value remains the same. Note that same as pooling layer, there are no parameters needed to be trained.

5	-2	7	ReLU	5	0	7
6	1	-1		6	1	0
-2	8	-9		0	8	0

The Demonstration of ReLU operation: A ReLU operation on a sample matrix with input on the left and output on the right.

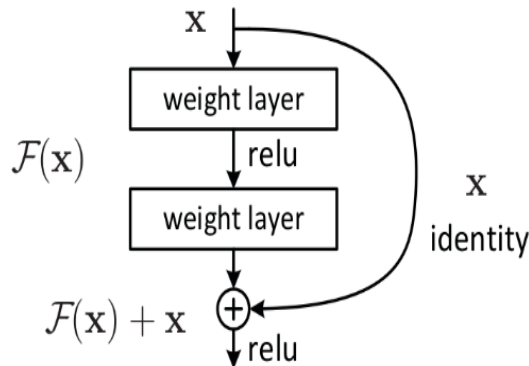
Fully Connected Layer

After the previous operation (pooling or convolution), fully connected layer, like its name, connect all the input feature maps from the previous layers to a larger feature map. In other words, the operation transforms the multi-dimensional input matrix, possibly of size $K \times K \times F$, from the previous layer to a vector (one-dimensional array). Note that there are more than one fully connected layers in general and each of the layers is followed by an activation function. In particular, the last fully connected layer inside a CNN connects all the features to the actual predicted objects (such as cat or dog), which will provide a score for each object. The operation inside this layer is relatively simple as well, which simply flattens the output (such as input map of size $6 \times 6 \times 256$) from the previous layer as input to a one-dimensional vector (a 9216×1 vector). Note that for the first fully connected layer in CNN, the input feature map is of size $K \times K \times F$, which is

transformed into one huge vector by taking it row by row. The transformation is similar to the conversion of filters into a weight matrix in the convolutional layer. Then, the vector is multiplied by a weight matrix (W) plus a bias (b). The operation is the same as a neuron function with a weight matrix, a bias, and an activation function. Considering the first fully connected layer in AlexNet with a transformed vector of size 9216×1 , we initialize a weight matrix of size 4096×9216 , which would have an output vector of size 4096×1 . Then, the output is treated as an input to the next fully connected layer, which is the last fully connected layer in AlexNet and maps to 1000 distinct classes.

Distributed Deep Convolutional with ResNet Model (DiDCResNet):

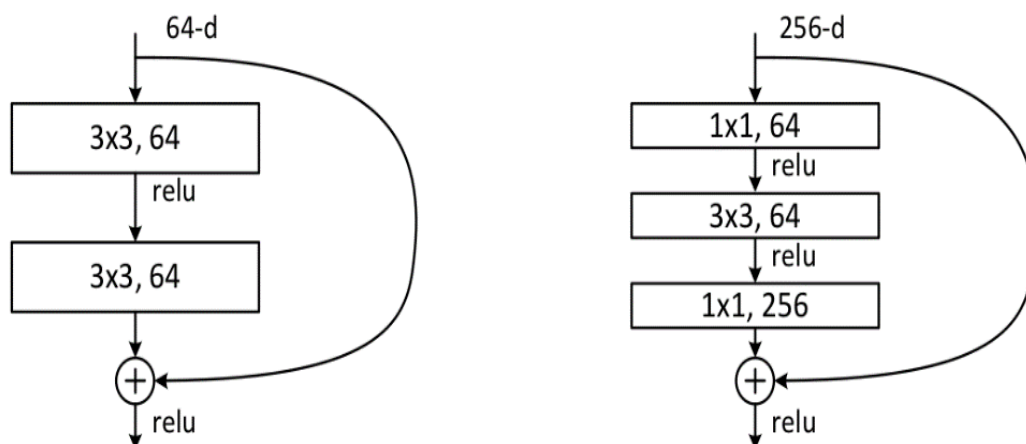
“ResNet50 is a variant of the ResNet model, which has 48 Convolution layers along with 1 Max-Pool and 1 Average Pool layer. It has 3.8×10^9 floating point operations. This is a widely used ResNet model and we have studied the architecture of ResNet50 in depth” [8]. Because we know that deep convolutional neural networks are really great at identifying low, mid, and high level features from images, and arranging more layers generally gives us better accuracy, so the question arises does the pattern work? better. more layers. This problem raises the gradient vanishing/exploding problem, these problems are largely solved in many ways and allow networks with dozens of layers to converge, but when deep neural networks start to converge, we see another problem that accuracy saturates and then it deteriorates. quickly and this is not caused by over-assembly, as might be expected, and adding more layers to a proper deep pattern just increases the learning error. The authors have addressed this problem by introducing a deep framework for residual learning, so they are introducing quick access links that simply perform identity matching.



“They explicitly let the layers fit a residual mapping and denoted that as $H(x)$ and they let the non-linear layers fit another mapping “ $F(x): =H(x)-x$ so the original mapping becomes $H(x): =F(x)+x$ as can be seen in Figure” [9]. And the benefit of these shortcut identity mapping was that there were no additional parameters added to the model and also the computational time was kept in check”.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

“Now we'll talk about Resnet 50, “as well as the architecture for the above-mentioned 18- and 34-layer ResNets, which is given residual mapping but not displayed for clarity. For the ResNet 50 and higher, a little adjustment was made in that previously, shortcut connections skipped two levels, but now they skip three layers, and there were also 1 * 1 convolution layers added, which we will go over in depth with the ResNet 50 Architecture”.



7 * 7 core size and 64 different cores, all with a step of size 2, which gives us 1 layer. Then we see a maximum pool also with step size 2. In the next convolution there is a 1*1.64 core, the next 3*3.64 core and finally a 1*1.256 core, these three layers are repeated a total of 3 times, so that gives us 9 layers in this step. Then we see a 1*1,128 core then a 3*3,128 core and finally a 1*1,512 core, this step is repeated 4 times, so it gives us 12 layers in this step.

Then there is a core of 1×1.256 and two more cores with 3×3.256 and 1×1.1024 and this repeats 6 times giving us a total of 18 layers. And then again, 1 core 1×1.512 with two more of 3×3.512 and 1×1.2048 and this repeats 3 times giving us a total of 9 layers. We then create an intermediate group and end it with a fully connected layer containing 1000 nodes, and finally a SoftMax function, giving us 1 layer. In fact, we do not take into account the activation functions and the maximum/mean unifying layers. In short, this gives us $1 + 9 + 12 + 18 + 9 + 1 = 50$ deep convolutional network layers.

Distributed Deep Concolutional with VGG Model (DiDeConvVGG)

Although AlexNet has shown that deep convolutional neural networks can work well, it does not offer a common template to guide future researchers in designing new networks. In the following sections, we will introduce some heuristic concepts that are commonly used to design deep networks. Progress in this area mirrors chip design, where engineers have moved from placing transistors to logic elements and logic blocks. Similarly, the design of neural network architectures has become increasingly abstract, with researchers moving from thinking in terms of individual neurons to entire layers and now to blocky, repeating layer patterns. The idea of using blocks first came from the Visual Geometry Group (VGG) at the University of Oxford, in its eponymous VGG network. It's easy to integrate these repetitive structures into code with any modern deep learning framework using loops and routines.

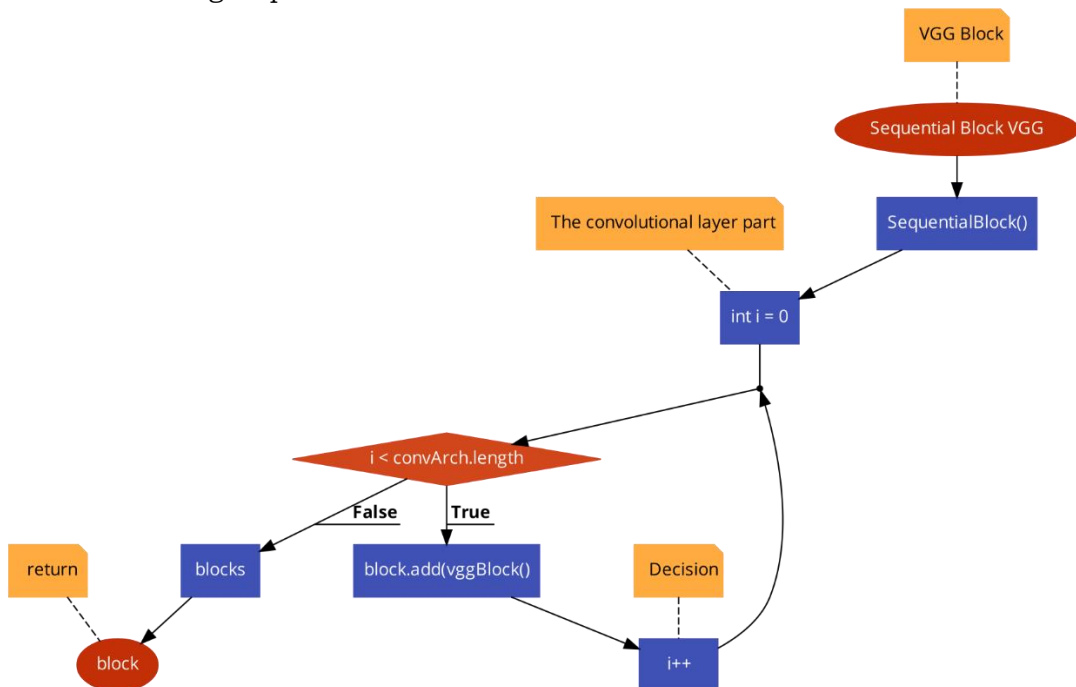


Fig: 3 Control Flow Graph Distributed Deep Concolutional with VGG Model

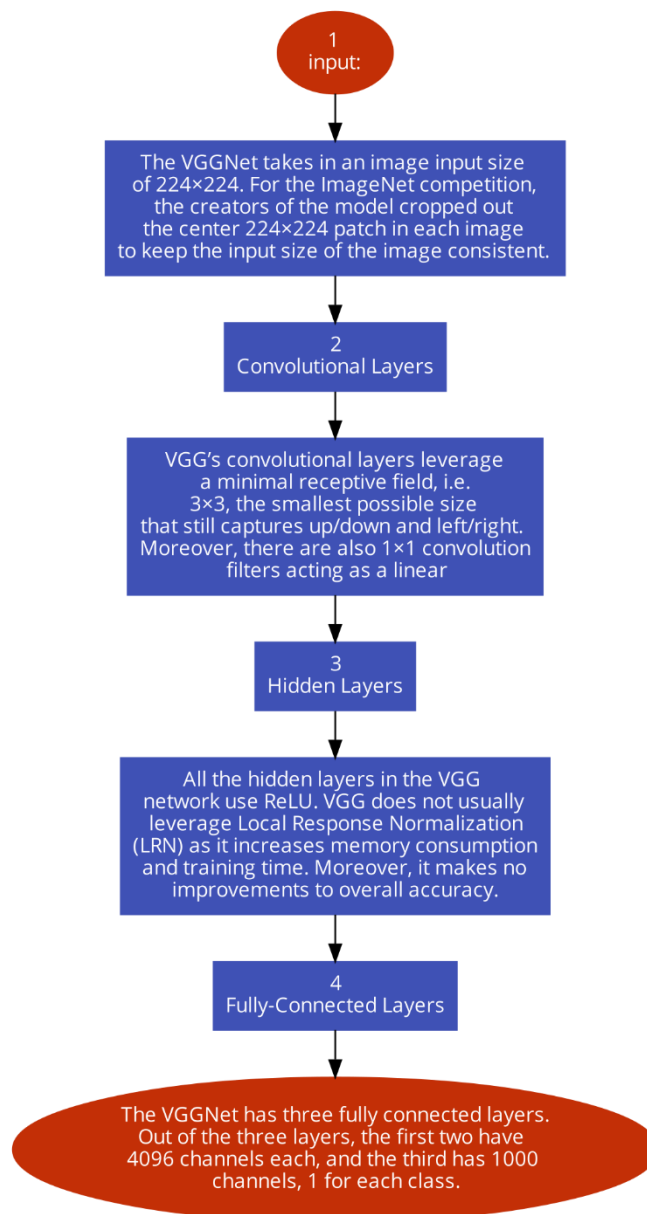


Fig: 4 Design steps for VGG

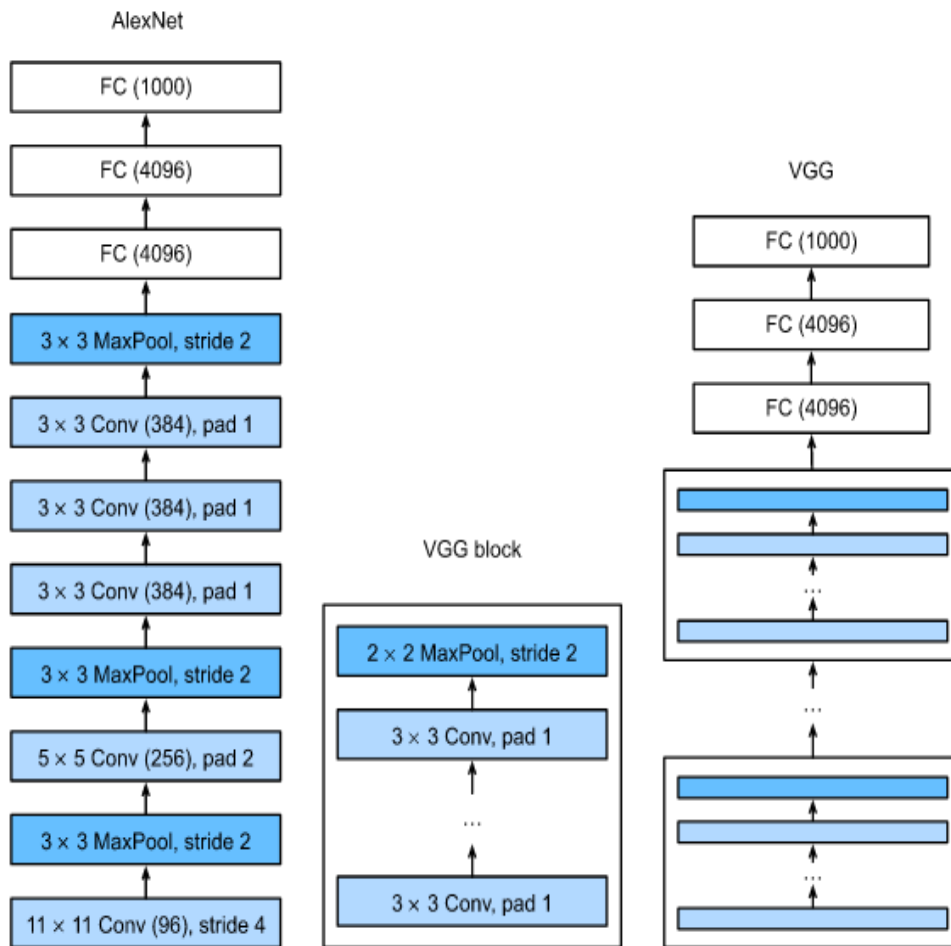


Fig: 5 VGG Network

The original VGG network included five convolutional blocks, the first two of which each have one convolutional layer and the latter three of which each have two convolutional layers. The first block has 64 output channels, and each consecutive block increases the number of output channels by a factor of two until the total number of output channels reaches 512. VGG-11 is the name given to this network because it has eight convolutional layers and three fully linked layers.

Complexity and challenges

The original VGG network included five convolutional blocks, the first two of which each have one convolutional layer and the latter three of which each have two convolutional layers. The first block has 64 output channels, and each consecutive block increases the number of output channels by a factor of two until the total number of output channels reaches 512. VGG-11 is the name given to this network because it has eight convolutional layers and three fully linked layers.

4 Experimental Results and Discussions

As a starting point, this section presents an architecture that consistently gives me with reasonable outcomes. There are four different sorts of X-ray images available in the model. For concluding comments, the confusion matrix, accuracy, precision, recall, and F-score are examined. The Covid class has an accuracy rate of 89% and a recall rate of 0.895%. As a result, the model is unable to correctly categorise all Covid-19 samples. It is, nonetheless, usually correct to do so (high precision - low FP). Precision and Recall (both better than 88 percent) are similar in the Lung Opacity and Normal classes, indicating that the model is good at recognising and categorising these samples. Here, we have to consider training models resnet50 with 1600 samples and maximum of 20 epochs and validating with 200 samples from the corpus data set.

(i): Training model resnet50 with 1600 samples and max of 20 epochs, and validating with 200 samples

<i>Epoch</i>	<i>train loss</i>	<i>val loss</i>	<i>time(sec)</i>
1	1.10196	0.91863	26.36
2	0.77953	0.64788	26.35
3	0.62232	0.58128	26.3
4	0.57838	0.64373	26.44
5	0.52742	0.52919	26.33
6	0.48673	0.53723	26.3
7	0.47148	0.5592	26.48
8	0.45234	0.46929	26.39
9	0.4299	0.54057	26.5
10	0.40807	0.45246	26.43
11	0.39559	0.50645	26.34
12	0.39706	0.45655	26.58
13	0.37875	0.43944	26.5
14	0.36924	0.42109	26.32
15	0.35774	0.42202	26.35

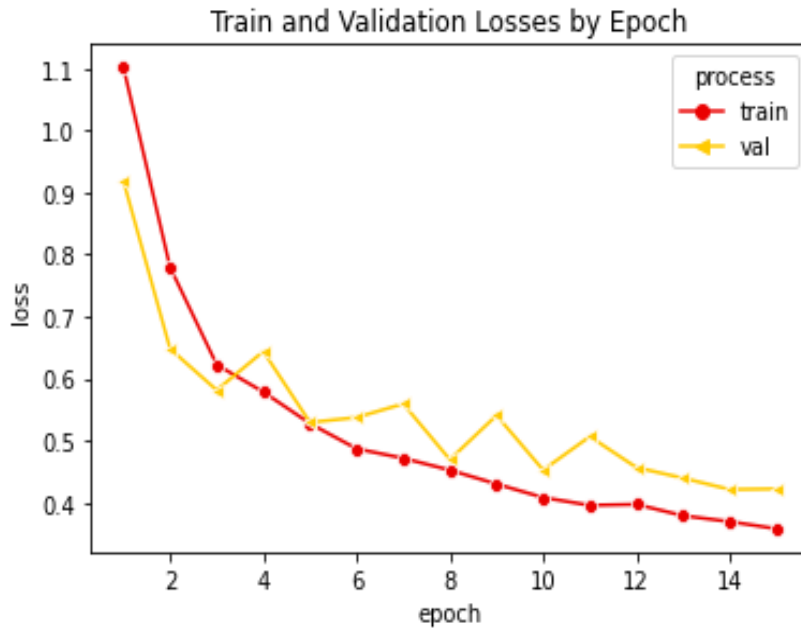


Fig: 6 train and validation losses by epoch

In Fig(6) ,we have to consider training models resnet50 with 1600 samples and maximum of 20 epochs and validating with 200 samples from the corpus data set. The relation between loss and epoch determination as show in fig (6).

Training model vgg16 with 1600 samples and max of 20 epochs, and validating with 200 samples:

<i>epoch</i>	<i>train loss</i>	<i>val loss</i>	<i>time(Sec)</i>
1	0.97049	0.77628	51.61
2	0.66439	0.6018	51.9
3	0.56597	0.54785	51.73
4	0.52116	0.55944	51.66
5	0.48218	0.49195	51.49
6	0.47613	0.47877	51.81
7	0.47394	0.46751	51.8
8	0.44823	0.44091	51.79
9	0.43496	0.43908	51.78
10	0.42257	0.43814	51.61
11	0.40768	0.42089	51.75
12	0.39885	0.43289	51.76
13	0.39214	0.42201	51.63
14	0.39148	0.39706	51.6
15	0.38929	0.41296	51.72

Here, we have to consider the Training model vgg16 with 1600 samples and max of 20 epochs, and validating with 200 samples and determines each epochs train loss and validation loss with respective time.

Briefly, our pre-planned primary analysis was to evaluate the impact of baseline characteristics of interest on the severity and recovery of COVID-19 in patients with lung cancer. We used literature review of COVID-19 publications and directed acyclic graphs to identify relationships between baseline features of interest and examined features that may directly impact COVID-19 infection severity and recovery.

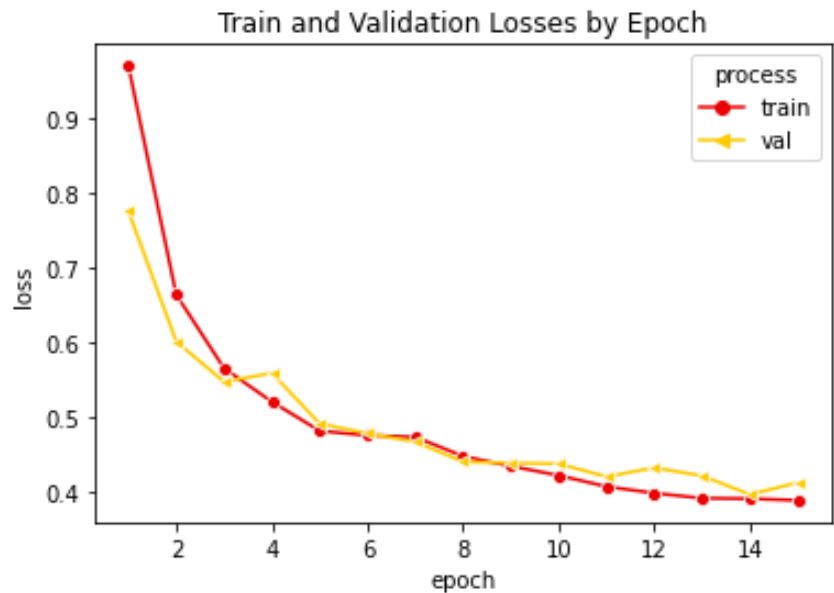
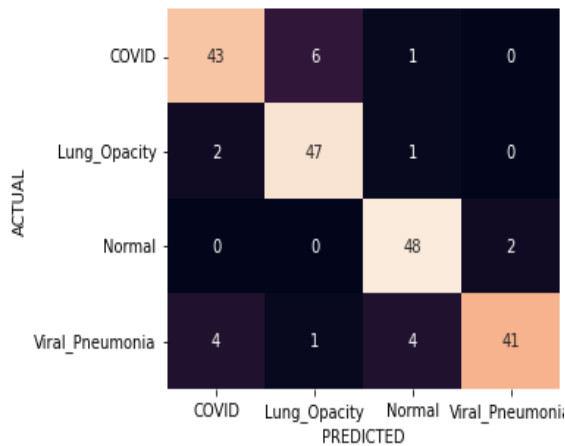


Fig: 7 Train and validation losses by epoch

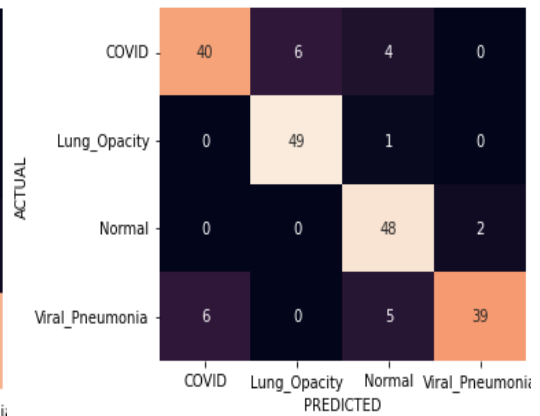
Here, we have to consider the Training model vgg16 with 1600 samples and max of 20 epochs, and validating with 200 samples and determines each epochs train loss and validation loss with respective time the relation between train and validation losses by epoch values shown in fig (7).

		Actual Values	
		0 (negative)	1 (positive)
Predicted Values	0 (negative)	TN	FN
	1 (positive)	FP	TP
		N (total negative)	P (total positive)

Fig 8: predicted and actual values



DiDeConResNet Model



DiDeConvVGG Model

In below table, “Overall, the model can recognise the samples, indicating that there is enough TP”. If Covid-19 is misclassified, samples of Normal or Lung Opacity can be expected. Viral Pneumonia is unlikely to be the cause. “Viral Pneumonia or Lung Opacity are more commonly misclassified as Normal than as Lung Opacity”. Viral Pneumonia or Lung Opacity are frequently misclassified as normal samples. It's less likely to be confused with COVID-19. The class with the fewest misclassifications is viral pneumonia”.

<i>Model</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
DiDeConResNet	0.895	0.897575	0.895	0.894333
DiDeConvVGG	0.88	0.88482	0.88	0.878175

The COVID class has an accuracy of 89% and a recall of 89%. The result indicates that the model is correctly categorise all Covid-19 samples. When it does, though, it is typically correct (high precision - low FP). Precision and Recall (above 80%) are similar in the Lung Opacity and Normal classes, indicating that the model is good at recognising and classifying these samples. In Covid-19, the Normal and Viral Pneumonia classes have the opposite outcome.

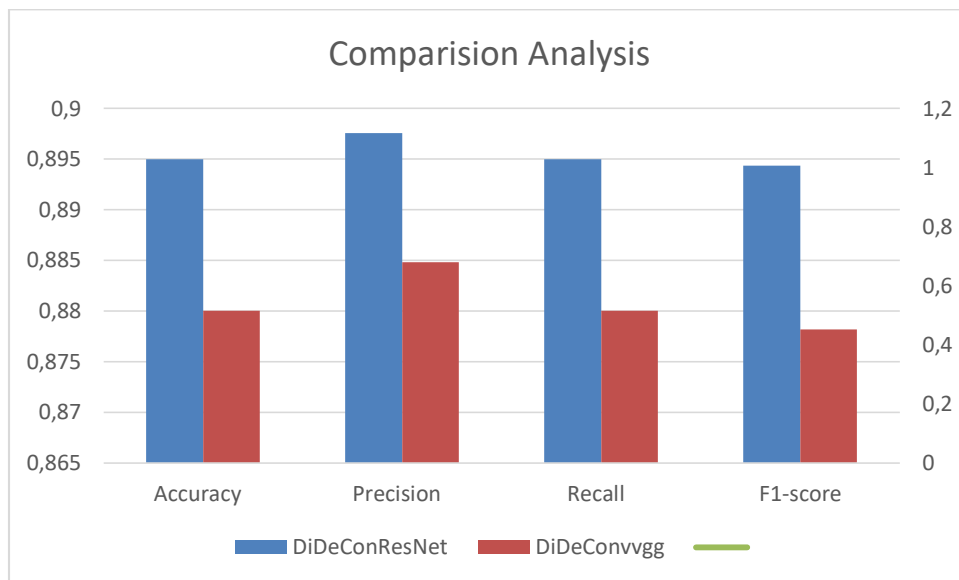


Fig: 9 Comparison Analysis

In Fig (9), They have a Recall that is higher than Precision. This indicates that the model can recognise these samples, as evidenced by the decreased number of FN. However, it causes FP, as seen in the Confusion Matrix, where the Normal class is frequently confused with Viral Pneumonia or Lung Opacity. The F-Score is a combination of Precision and Recall. Because the Precision and Recall criteria are similar, the Normal class has a high score. The Precision and Recall scores diverged more intensely for COVID, resulting in lower outcomes. Overall, the fact that all the general indicators are above 70% is a positive result. The quantity of COVID samples incorrectly labelled as NORMAL samples should be reduced, and the results per class should be improved.

In Fig (8)A confusion matrix is a matrix (table) that can be used to assess the effectiveness of a machine learning system, often one that employs supervised learning. The instances of an actual class are characterized in every single row of the confusion matrix, whereas the instances of a predicted class are represented in each column. This is how we keep it in this part of our course, but it can also be the other way around, with predicted classes in rows and real classes in columns. The name confusion matrix refers to the fact that it allows us to quickly understand what types of errors are present in our classification systems.

4. Conclusions and Future Scope

Deep learning algorithm can aid healthcare workers in detecting COVID-19 with minimal processing of chest X-ray images. More articles on diagnosing lung disease through deep learning have been published over time. There was, however, no comprehensive examination of the current state of research and implementation. As a result, this document was established to provide an exhaustive study for the diagnosis of lung diseases through deep learning, released from 2016 to September 2020, on tuberculosis, pneumonia, lung cancer, and COVID-19. Existing studies to detect COVID-19 affect lung cancer using

machine learning and deep learning algorithms with X-ray images of the lungs, using the necessary algorithms addressed with less likely results. This is a challenging situation for researchers, scientists, and healthcare professionals. Researchers are constantly working to find possible solutions to deal with this pandemic in their respective fields. We need to train each data set individually to make the prediction and test the effectiveness of the solution. The proposed methods have had a significant impact in the country as a warning to society. Therefore, the existence of such a hypothesis for innovative research will significantly reduce the incidence of lung cancer deaths affected by COVID-19 and will be considered for prior public advice. Precision and Recall are both important factor for this application, hence the results are analysed in terms of F1-Score. The confusion matrix and results for the F1-Score, Precision, Recall, and overall Accuracy are also presented to provide a complete analysis of the Model performance. Other challenges were acknowledged and explained, such as the future direction of lung disease screening with deep learning and transfer learning, which would be introduced with high-end cloud data servers. Other researchers could utilize the taxonomy to create their own investigations and activities. The proposed future path has the potential to improve the number of applications for in-depth lung disease diagnosis while simultaneously increasing efficiency.

References

1. Ankita Shelke, Madhura Inamdar, Vruddhi Shah, Amanshu Tiwari, Aafiya Hussain, Talha Chafekar, Ninad Mehendale. "Chest X-ray classification using Deep learning for automated COVID-19 screening" , Cold Spring Harbor Laboratory, 2020
2. Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach , Springer Science and Business Media LLC, 2020
3. Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, Springer Science and Business Media LLC, 2019
4. Gao, X.W.; James-reynolds, C.; Currie, E. Analysis of tuberculosis severity levels from CT pulmonary images based on enhanced residual deep learning architecture. *Neurocomputing* 2019, 392, 233–244. [CrossRef]
5. Gordienko, Y.; Gang, P.; Hui, J.; Zeng, W.; Kochura, Y.; Alienin, O.; Rokovyi, O.; Stirenko, S. Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer. *Adv. Intell. Syst. Comput.* 2019, 638–647. [CrossRef]
6. Haitz Sáez de Ocáriz Borde, David Sondak, Pavlos Protopapas. "Convolutional neural network models and interpretability for the anisotropic reynolds stress tensor in turbulent one-dimensional flows" , Journal of Turbulence, 2021
7. Hardianti, S., Syarif, S., Ahmad, M., Niswar, M., Stang, S., & Mastuti, N. L. P. H. (2021). The effect of web-based learning media towards the skills of the second stage of childbirth care practicum in D-III midwifery students. *International Journal of Health & Medical Sciences*, 4(2), 226-231. <https://doi.org/10.31295/ijhms.v4n2.1713>
8. Kabari, L.G.; Onwuka, U. Comparison of Bagging and Voting Ensemble Machine Learning Algorithm as a Classifier. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* 2019, 9, 1–6.
9. Kangle Zhu, Lian Wang, Zanhao Chen, Jianing Ding, Jixuan Dong, Jinliang

- Chen. "Analysis and construction of an exosome derived competing endogenous RNA network for small cell lung cancer in the exoRbase database" , Research Square Platform LLC, 2022
10. Ker, J.; Wang, L. Deep Learning Applications in Medical Image Analysis. IEEE Access 2018, 6, 9375–9389. [CrossRef]
 11. Kieu, S.T.H.; Hijazi, M.H.A.; Bade, A.; Yaakob, R.; Jeffree, S. Ensemble deep learning for tuberculosis detection using chest X-Ray and canny edge detected images. IAES Int. J. Artif. Intell. 2019, 8, 429–435. [CrossRef]
 12. Kushagra, Rajneesh Kumar, Shaveta Jain. "Recent Advances in Machine Learning for Diagnosis of Lung Disease: A Broad View" , 2022 IEEE Delhi Section Conference (DELCON), 2022
 13. M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" IEEE Access, Vol. 8, 2020, pp. 132665 - 132676.
 14. Mikołajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop, Swinoujscie, Poland, 9–12 May 2018; pp. 117–122. [CrossRef]
 15. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs . Traditional Computer Vision. Adv. Intell. Syst. Comput. 2020, 128–144. [CrossRef]
 16. Rahaman, M.M.; Li, C.; Yao, Y.; Kulwa, F.; Rahman, M.A.; Wang, Q.; Qi, S.; Kong, F.; Zhu, X.; Zhao, X. Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches. J. X-Ray Sci. Technol. 2020, 28, 821–839. [CrossRef]
 17. Salman, F.M.; Abu-naser, S.S.; Alajrami, E.; Abu-nasser, B.S.; Ashqar, B.A.M. COVID-19 Detection using Artificial Intelligence. Int. J. Acad. Eng. Res. 2020, 4, 18–25.
 18. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J. Big Data 2019, 6. [CrossRef]
 19. Suryasa, I. W., Rodríguez-Gámez, M., & Koldoris, T. (2022). Post-pandemic health and its sustainability: Educational situation. International Journal of Health Sciences, 6(1), i-v. <https://doi.org/10.53730/ijhs.v6n1.5949>
 20. Wang, C.; Chen, D.; Hao, L.; Liu, X.; Zeng, Y.; Chen, J.; Zhang, G. Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model. IEEE Access 2019, 7, 146533–146541. [CrossRef]
 21. World Health Organization. Coronavirus Disease 2019 (COVID-19) Situation Report; Technical Report March; World Health Organization: Geneva, Switzerland, 2020.