COVID-19 Disease Detection using Pre-Trained Deep Convolutional Neural Network (GoogleNet) on Cloud Platform

[1]Sachin Kumar*
Department of Computer Application V.B.S P.U, Jaunpur, U.P, INDIA
Email: jaiswalsachin009@gmail.com

[2]Saurabh Pal
Department of Computer Application V.B.S P.U, Jaunpur, U.P, INDIA
Email: drsaurabhpal@vbspu.ac.in

[3]Vijendra Pratap Singh
Department of Computer Science & Application M.G.K.V.P, Varanasi, U.P,INDIA
Email: vijendra@mgkvp.ac.in

[4]Sanjay Kumar Singh
Department of Computer Science & Applications, K.I.T, Varanasi, U.P,INDIA
Email: Sanjayvns1810@gmail.com

[5]Sandeep Kumar Singh
Department of Computer Science & Engineering K.I.T, Varanasi,U.P,INDIA
Email: sandeep007viet@gmail.com

[6]Priya Jaiswal
V.B.S P.U, Jaunpur, U.P, INDIA
Email: jaiswalsachin001@gmail.com

Abstract:

Background and Purpose: The COVID -19 epidemics are causing the main rash in more than 151 countries around the whole world.Covid-19 has a bad effect on human life worldwide. One of the critical steps in fighting COVID-19 is finding the contaminated patients early enough and putting these infected people under special care. Our main aim is to separate COVID-19 patients from other patients.

Materials and Methods: In this research article, we used GoogleNet as a learning network. GoogleNet is a deep convolutional neural network of 22 layers deep. We have used a pre-trained version of the GoogleNet trained on ImageNet. The pre-trained GoogleNet image input size is 224 x 224.GoogleNet; the deep convolutional neural network model can analyze X-ray images to classify the patient's condition of the affected disease.

Result: Experiments and evaluation of the GoogleNet have been effectively done based on 80% of X-ray pictures for training and 20% of X-ray pictures for testing phases respectively. GoogleNet shows a good result for disease classification with 91.40% of accuracy in 2.49 minutes.

Conclusion: In this research paper, we have used the deep CNN model to classify COVID-19 disease using X-ray images based on the projected GoogleNet. Scientific studies will be the next goal of this research article.

Keywords: Chest diseases, Classification, Convolutional neural network (CNN), GoogleNet, COVID-19

DOI Number: 10.14704/nq.2022.20.9.NQ44231 **Neuro Quantology 2022; 20(9):1989-2008**

1. Introduction: Now days, a deep CNN model can attain a better precision in segmenting and analyzing an image [1].Deep CNN can play an important role for image classification. With the help of deep learning, we are able to recognize lesions and tumors in medical images [2, 3]. Diagnostic through computer [4, 5], the investigation of health related data through electronic tools [6], the development of drug intake and treatment [7], atmosphere identification [8], and computer-human brain interface [9],aiming to provide assessment of the human's health. The main factor of the accomplishment of deep CNN is based on the deep CNN to learn complex-level of abstraction [10].Experts can take support of deep learning for disease recognition like COVID-19 and other chest diseases.

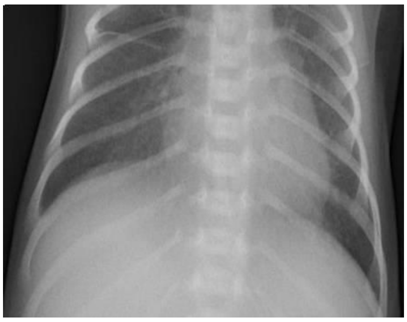
1.1. **Infection in chest:** A chest disease is a contamination that infects the lower part of respiratory tract. The respiratory tract includes the bronchi, lungs and windpipe. Some common type of infection in chest is viral pneumonia, bacterial pneumonia, tuberculosis. The infection in chest can vary from mild to severe. We have also taken an image dataset of viral pneumonia, bacterial pneumonia, tuberculosis, and COVID-19 and normal chest [11].Figure 1 represents different types of chest images. **1.2.**

A.BACTERIAL PNEMONIA

B.COVID-19

C.NORMAL CHEST

D. TUBERCULOSIS



E. VIRAL PNEMONIA

FIG 1: DIFFERENT TYPES OF INFECTED CHEST IMAGES

- **COVID-19:** The novel corona virus epidemic come into view in China in December 2019 and it has turn into a severe public health problem worldwide [12, 13].Corona virus belongs to the family of perilous virus [14].The name of COVID-19 virus is due to their crown-like structure observed by using electron microscope [15].COVID-19 may cause rigorous and contagious syndrome. The eruption of the COVID-19 in Wuhan, China has been quickly extended to the all over the world since December 2019 [16-19].This virus is named as COVID-19 on Feb 11, 2020 by the world health organization. There have been 80,894 confirmed cases in china to date March 18, 2020 and 204,037 confirmed cases worldwide [20].

- **Pneumonia:** It is a disease of the human lungs that may be caused by pathogens, like bacteria, viruses, and fungi [21]. It can happen to anybody, young or healthy people. It becomes life severe for infants, people with the impaired immune system, people suffering from any other disease, hospitalized people, elderly people, people positioned on a ventilator, people suffering from persistent diseases like people who smoke cigarettes, and asthma.

- **Viral Pneumonia**: The symptoms of viral pneumonia occur slowly and it is milder. Viral pneumonia becomes difficult to detect if bacterial pneumonia occurs at the same time with viral pneumonia.

- **Bacterial Pneumonia:** Symptoms of bacterial pneumonia can occur rapidly or even gradually among kids [22].Bacterial pneumonia infects a big fraction of the human lungs and can direct to contaminating many lobes of the human lung. When any person's numerous lobes of lungs are infected then that person needs to be hospitalized [23]. Another type of pneumonia is fungal pneumonia. It can occur in individuals with feeble immune systems. Fungal pneumonia can be hazardous also and require time for the patient to recover health [24].

- **Tuberculosis:** Tuberculosis is a contagious syndrome generally caused by Mycobacterium tuberculosis (MTB) bacteria. It distresses the human lungs. Tuberculosis can also have an effect on other parts of the human body. Most contaminations show no warning sign, in which case it is recognized as latent tuberculosis. The bacteria that cause tuberculosis are spread when a contaminated person coughs or sneeze. Treatment is not always required for those without symptoms. Patients with active symptoms will need a long course of cure involving numerous antibiotics [25].

The central goal of this paper is a classification of COVID -19 patient among several other patients suffering from chest diseases and trying to find the efficiency of GoogleNet for recognition of patients suffering from COVID-19 among several other patients with a deep convolutional neural network and to project CNN as a model for classification chest diseases, concentrating on images of chest disease. This research paper proposed two main contributions of chest disease classification:

i. **Enhancement in the classification of chest disease using deep CNN model:** Deep learning techniques demonstrates better result in image classification. A deep CNN learning technique has the ability to make use of unprocessed data directly exclusive of feature engineering. Besides, deep learning models recommend the choice of transfer learning from another job using pre-trained deep learning models on bigger image datasets.

ii. **Recognition of chest infection signs from the contaminated chest:** The searching of the contaminated chest helps out the client of the designed sculpt by providing them knowledge about the chest ailment. Also, this natural information is pulled out exclusive of the interference of chest specialists.

In this proposed paper, we have used GoogleNet to identify the infection in the human chest. Our attempt will resolve the human chest disease identification problem. Our method will assist doctors to take out the infection in the human chest in a timely fashion and thus raise the quality of image classification for human chest disease. We have collected the image dataset of human chest disease from Kaggle. After the collection of the human chest disease image dataset, we have classified the human chest disease by using a deep CNN model. The effectiveness and performance of our projected model have been checked on GoogleNet on various parameters such as F1 score, sensitivity, and specificity with respect to the pre-trained deep CNN model.

TABLE 1: COMPARISON BETWEEN TECHNIQUES FOR PLANT DISEASE CLASSIFICATION

Sr.no	Authors	Year	No. of classes	No. of images	CNN Architecture	Measures	Accuracy %
1	Kawasaki,Y.et al [26]	2K15	Three	Eight Hundred	Customized	Sensitivity,Accuracy, Specificity	94.90
2	Sladojevic S.et al.[27]	2K16	Fifteen	4K483	CaffeNet	Accuracy	96.30
3	Mohanty et al.[28]	2K16	Thirty Eight	54K306	GoogleNet ,AlexNet	Mean Position, F1Score, Mean Recall	99.34
4	Nachtigall.L.G et al.[29]	2K16	Five	14K50	AlexNet	Precision,Recall, Accuracy	97.30
5	Fujita,E.et al.[30]	2K16	Seven	14K840	Customized	Specificity,Sensitivity, Accuracy	82.30
6	Brahmini,M.et al.[31]	2K17	Nine	14K828	GoogLeNet ,AlexNet	Macro Precision,Accuracy, Macro Recall	99.18
7	DeChant C.et al.[32]	2K17	Two	1K796	Customized	Precision,Accuracy, Recall	96.70
8	Lu,Y.et al.[33]	2K17	Ten	Five Hundred	AlexNet	Accuracy	95.48
9	Wang,G.et al.[34]	2K17	Four	2K086	VGG 19,VGG 16, Inception V3,ResNet-50	Accuracy	90.4
10	Brahimi,M.et al.[35]	2K18	Thirty Eight	54K323	DenseNet169,Alex Net, Inception V3, SqueezeNet1.1, ResNet-34, VGG 13	Accuracy	99.76
11.	Wang,J.et al.[36]	2K18	Five	2K430	Customized	Accuracy	90.84
12.	Rangarajan,A.K.et al.[37]	2K18	Seven	13K262	VGG 16,AlexNet	Accuracy	97.49
13	Khandelwal,I.et al.[38]	2K18	Fifty Seven	86K198	ResNet-50,Inception V3	Accuracy	99.374
14	This Work	2K20	Seven	6K594	GoogleNet	Precision, Recall, F1 Score, Accuracy,	97.11

The table 1 given below corresponds to the comparison between techniques of numerous plants disease.

The remaining part of this proposed manuscript planned as follows:

ii. The literature review about the proposed technology is explained in this part.
iii. Proposed Methodology is for GoogleNet is explained in this section.
iv. Discussion about the chest disease dataset collection and chest disease image dataset preparation is discussed in this section.
v. Proposed model and training the model is on GoogleNet discussed in this part.
vi. Performance evaluation of GoogleNet is discussed in this section.
vii. The result analysis and result discussion of GoogleNet are explained in this section.
viii. Conclusion and Future work about the proposed work is discussed in this section.

2. Literature Review: Nowadays, the Deep CNN method is used for image classification. It is an influential deep learning architecture. It has been extensively used in machine learning tasks. Table 2 represents the training time and accuracy on ImageNet.

Table 2: Training Time and Accuracy on ImageNet

Sr.No	Authors	Mini batch Size	Processor	DL Library	Time	Accuracy
1	He et al.[39]	256	Tesla P100 x 8	Caffe	29 Hours	75.3%
2	Goyal et al.[40]	8192	Tesla P100 x 256	Caffe2	1 Hour	76.3%
3	Smith et al.[41]	8192 to 16384	Full TPU Pod	TensorFLow	30 min	76.1 %
4	Akiba et al.[42]	32768	Tesla P100 x 1024	Chainer	15 mins	74.9 %
5	Jia et al.[43]	65536	Tesla P40 x 2048	TensorFlow	6.6 mins	75.8%
6	Ying et al.[44]	65536	TPU v3 x 1024	TensorFlow	1.8 mins	75.2 %
7	Mikami et al.[45]	55296	Tesla V100 x 3456	NNL	2 mins	75.29 %
8	This Work	32	Tesla P100-1024	MXNet	2.49 min	91.40%

- Several research articles already available for COVID-19 recognition. In [46] a deep CNN model is capable to forecast the coronavirus disease from a chest X-ray image. We have projected a novel CNN model that is GoogleNet. In this research paper, the images are classified into five classes are Bacterial Pneumonia, COVID-19, Normal Chest, Tuberculosis, and Viral Pneumonia.

- Abbas et al [47] introduce a new architecture of CNN to increase the performance of CNN models in classifying X-ray images. In this research paper, they have projected an architecture that is DeTrac. There are three stages. In the first stage, a pre-trained convolutional neural network for feature mining is applied. In the second stage, an optimization technique is used for training and finally, the class composition layer is adapted for the final classification of the images using error correction criteria applied to the softmax layer. For this research they used GoogleNet and results showed 95.12% accuracy on CXE images.

- Zhang et al [48] projected a new deep CNN model for fast and trustworthy COVID-19 disease recognition based on chest X-ray images. In this research paper, the researcher has proposed a model consisting of three components namely a variance of detection head, a classification head, and a backbone head. The variance detection head and classification head have the same architecture. The variance detection produces the scalar irregularities which in turn recognize COVID-19 patients. The high-level features images are taken out by backbone networks. These images are used as input into the image classification convolution layer. The projected model attained to decrease the false-positive rate. More, especially, the result demonstrates specificity and sensitivity of 70.65% and 96% respectively.

- Deep learning-based techniques are previously being used in a variety of areas [49-53]. M.I Razaak et al [54] proposed the difficulties and opportunities of medical image processing. Many research articles have already been proposed for the recognition of various infections by using deep learning techniques, as proposed by Dinggang Shen et al [55]. Andre et al [56] proposed a research article on dermatologist-level classification of skin cancer by using deep learning methods and F.Milletari et al [57] projected an article on the depiction of prostrate in MRI volumes using a convolutional neural network.

- Grewal et al [58] projected a research article on brain hemorrhage recognition in CT scans by using deep learning technique and Varun et al [59] presented a technique for recognizing diabetic retinopathy in retinal photographs. Y.Bar et al [60] also used deep learning techniques for chest pathology recognition. Techniques concerning the inspection of the recognition of infection by chest X-ray images have also been worked on numerous assessment techniques [61-63].

3. Proposed Methodology: In this section, we have discussed the course of action to classify human chest disease. We have developed an algorithm for convolutional neural networks using GoogleNet. There are a small number of steps to perform a deep convolutional neural network. The first step is to load the RGB image of the infected chest, the second step is to resize the image of the infected chest in 224 x 224 x 3 size, the Third step is feature extraction of the infected chest, the fourth step is chest disease classification and in 5th step, we will get chest disease name as output. Fig 2 describes the steps for human chest disease recognition.

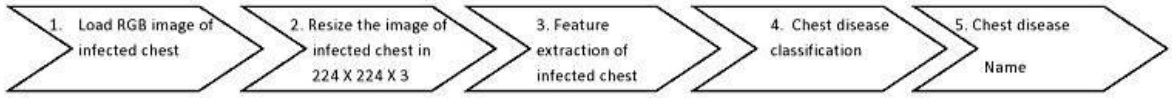
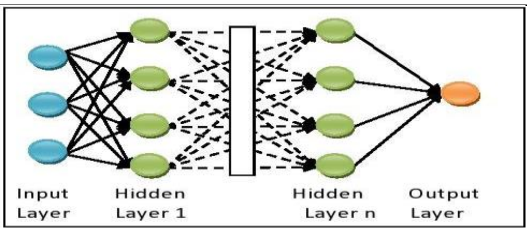
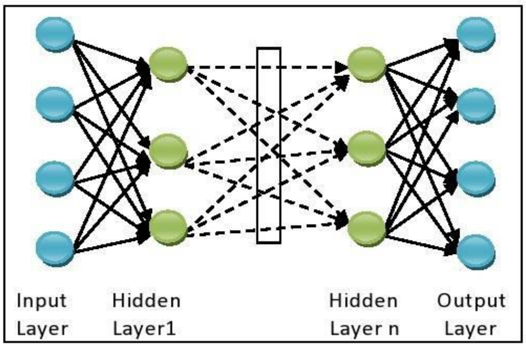
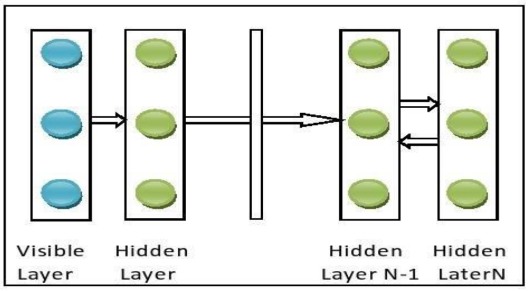
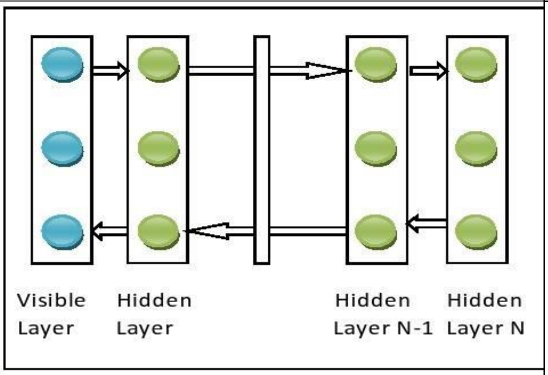
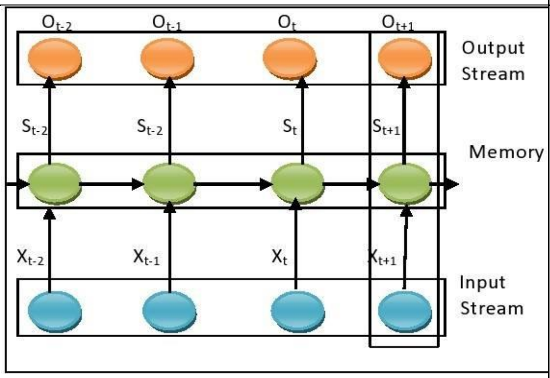
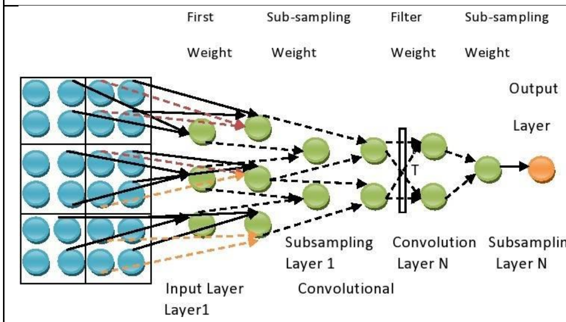


Fig 2: Steps for Chest disease recognition

Table 3 represents different learning architecture of different networks. **TABLE 3: DIFFERENT DEEP LEARNING ARCHITECTURE**

ARCHITECTURE	DESCRIPTION	KEY POINTS
	**Deep Neural Network** • Deep structure generally used for classification or regression. • Made of more than 2 hidden layers. • Allows Nin-linear hypothesis to be expressed.	**Positives:** • Successfully used in many fields. **Negatives:** • Training is big because once the error is passes to the first layers it become small.
	**Deep Auto encoder** • Mainly designed for dimensionality reduction or feature extraction [64]. • It has the equal number of input and output. • Main aim to reconstruct the input and vectors. • It is unsupervised learning method.	**Positives:** • No requirement of labeled data for training. • Several variations were expected to make the representation more powerful: Sparse AutEnc.[65] ,Denoising AutEnc [66] ,Contractive AutEnc [67],Convolutional AutEnc.[68] . **Negatives:** • Pre-training stage is required.
	**Deep Belief Network** • It is aggregation of RBM where each subnet hidden layer acts as the visible layer for the next [69]. • It has unguided connections only in the upper two layers. • Supervised and unsupervised training of the network is permitted.	**Positives :** • Recommending a greedy learning approach layer – by- layer to initiate the network. • Traceable interfaces to directly maximize the potential. **Negatives:** • Training process is computationally costly due to the initialization procedure and sampling.

Architecture	Description	Key Points
	**Deep Boltzmann Machine** • It is another approach based on the Boltzmann family [70]. • Handles undirected links between all layers. • Uses a random probability algorithm [71] to maximize the minimum likelihood.	**Positives:** • Includes a top-down view for more powerful interventions with the ambiguous component. **Negatives:** • Time complexity for the interferences is higher than DBN. • The time complexity of the interventions is higher than that of DBN. • Unrealistic optimization factors for a large data set.
	**Recurrent Neural Network** • It is a neural network capable of investing the flow of data [72]. • It is useful in applications where the output depends on the previous calculations. • It shares the similar weight across all steps.	**Positives:** • It can remember sequential events. • It can model time reliance. • It has shown great achievement in many applications of natural language processing. **Negatives:** • It may need many layers to find a complete hierarchy of visual features. • May require many layers to find a complete hierarchy of visual features. • It usually needs a big dataset of categorized images.
	**Convolutional Neural Network** • Good compatibility with 2D data such as images [73]. • All hidden convolutional filter convert its output to a 3D output volume for neurons activations. • Motivated by the neurobiological model of the visual cortex [74].	**Positives:** • Few neurons connections are mandatory with respect to a typical NN. • Several variables were expected; AlexNet[75], Clarifai [76] and GoogleNet [77]. **Negatives:** • It may need many layers to find a complete hierarchy of visual features. • It is usually need a large dataset of categorized images.

A. **Convolutional Neural Network (CNN):** CNN is an efficient machine learning method. We will require a vast amount of images to train CNN. The bigger the collection of images, the richer the features that CNN learns. Training any CNN from a large collection of different images is not a simple task. However there is a simple way, we can train our dataset by using deep CNN. The figure given below represents the overall structure of CNN. Fig 3 represents the structure of CNN.

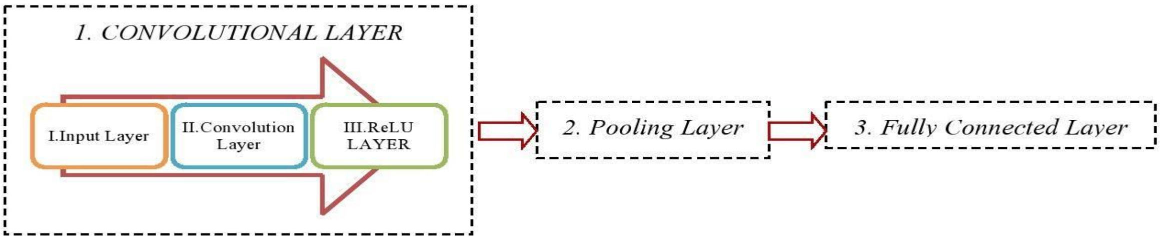


Fig 3: Convolutional Neural Network

Types of CNN layers: There are a number of layers of convolutional neural networks. Different layers of the CNN given below:

I. **Input Layer:** Convolutional neural network accepts images as input. The dimension of the images is 224 x 224 x 3.

II. **Convolution Layer:** The convolution layer measures the dot product between the image patch and all filters.

III. **ReLU Layer:** The Rectified Linear Unit (ReLU) is an activation function that has been adopted in the design of most neural networks, especially convolutional neural networks. ReLU is the identity function, f(x) =x, for all positive values and zeros for all negative values of input 'x'. ReLU is lightly activated; it helps to transcribe the inactivity of the organic neuron to confirmed impulses. Table 4 represents the value of the X-axis and Y-axis for a rectified linear unit.

TABLE 4:Rectified Linear Unit

x	-4	-3	-2	-1	0	1	2	3	4
f (x)	f (-4)	f (-3)	f (-2)	f (-1)	f (0)	f (1)	f (2)	f (3)	f (4)
F (x)	0	0	0	0	0	1	2	3	4

For the above given table, we have find out the graph given below in figure 4.

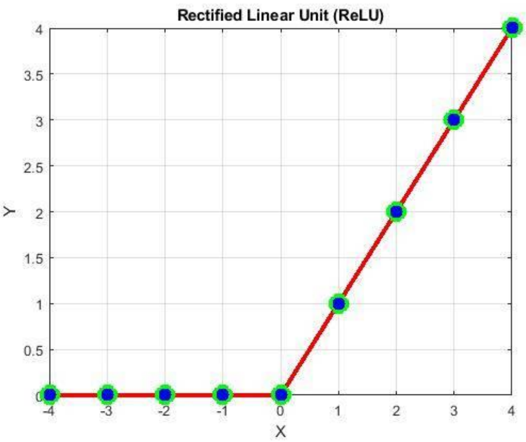


Fig 4: Rectified Linear Unit

B. **Max Pooling Layer:** It generally comes after the activation layer. The max-pooling layer maximally activates only a group of neurons from the feature map. Max pooling layer used with a stride factor of '2' on a '2 x 2' window. The max-pooling layer effectively reduces the height and the width of the feature maps while preserving the number of channels. We have restricted the X and Y axis by 150 and 170 regions. Figure 5 represents the function of the max pool layer.

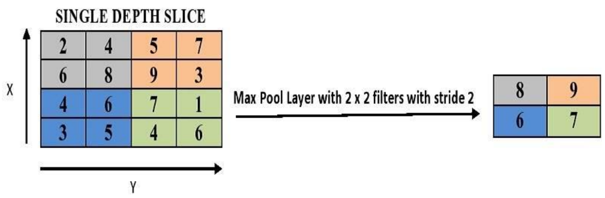


Fig 5: Max Pool Layer

C. **Fully Connected Layer:** Any given array is converted into a one-dimensional array by using a fully connected array. This one-dimensional array is the same as the dimension of the classes.

3. Dataset Collection & Dataset Preparation:

In this segment, we have discussed image dataset collection of human chest diseases and dataset preparation for human chest disease classification.

3.1 Dataset Collection: We have collected 1030 images of the image dataset from Kaggle. The Kaggle is an open repository that contains 54323 images of 38 types of chest disease. The collected images of the chest are infected by six diseases. We have applied the training process for 65% - 90% data for training of infected chest images and 35 % -10 % data for testing of infected chest disease. We have done training and testing on the following dataset given below in table 5.

TABLE 5: INFECTIONS IN HUMAN CHEST

SR.NO	CLASS	CHEST DISEASE
1	A	BACTERIAL PNEMONIA
2	B	COVID-19
3	C	NORMAL CHEST
4	D	TUBERCULOSIS
5	E	VIRAL PNEMONIA

3.2 Dataset Preparation: We have used a deep learning algorithm to train our model on the available dataset. We have collected enough chest images to train our model. It is always a tedious job to have accurate data. If we will not process the image data correctly then we will not able to get the desired outcome. So before training the model, we have to organize the data properly. We have resized the image into 224 x 224 x 3 dimensions and convert them into the grayscale images from the RGB image as shown in figure 6.

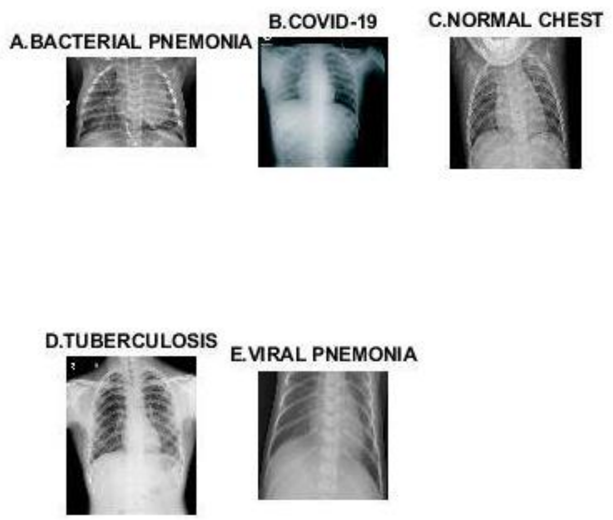


Fig 6: Resized Infected Chest

4. Proposed Model: In this research paper, we have proposed a model namely a pre-trained deep convolutional neural network with GoogleNet. Our center of attention is to classify COVID-19 disease among several chest diseases. Our concentration is to classify COVID-19 disease by using a pre-trained deep convolutional neural network with GoogLeNet. In our proposed model, the architecture of deep convolutional neural networks has designed for image recognition tasks on the ImageNet dataset.

5. Training of the model: For the training of our chosen model, we have laden a pre-trained deep convolutional neural network. There are numerous pre-trained deep convolutional neural networks available. We have selected GoogleNet for training purposes.

GoogleNet: Szegedy et al [78] used GoogleNet and was the winner of ILSVRC in 2014.GoogleNet encloses 9 inception units and owns 7 million parameters, four max-pooling layers, four convolutional layers, five fully-connected layers,3 softmax layers, and three average pooling layers for the main secondary classifier in the network [79].

5.1 Loss Type: We have set up a loss-type function namely "cross-entropyex".Cross entropyex is mostly required for multiclass classification. To split training and testing image data from the same image dataset. We have used a function namely splitEachLabel ( ).we have used a diverse range of image data for the training and testing process individually. By using a randomized ( ) function we have selected the images of the infected human chest.

Output of Loss function:

ClassificationOutputLayer with properties:

Name: 'output'

Classes: [1000×1 categorical]

OutputSize: 1000

Hyperparameters

LossFunction: 'crossentropyex'

Figure 7 represents the architecture of GoogleNet which is formed by using a MATLAB function.

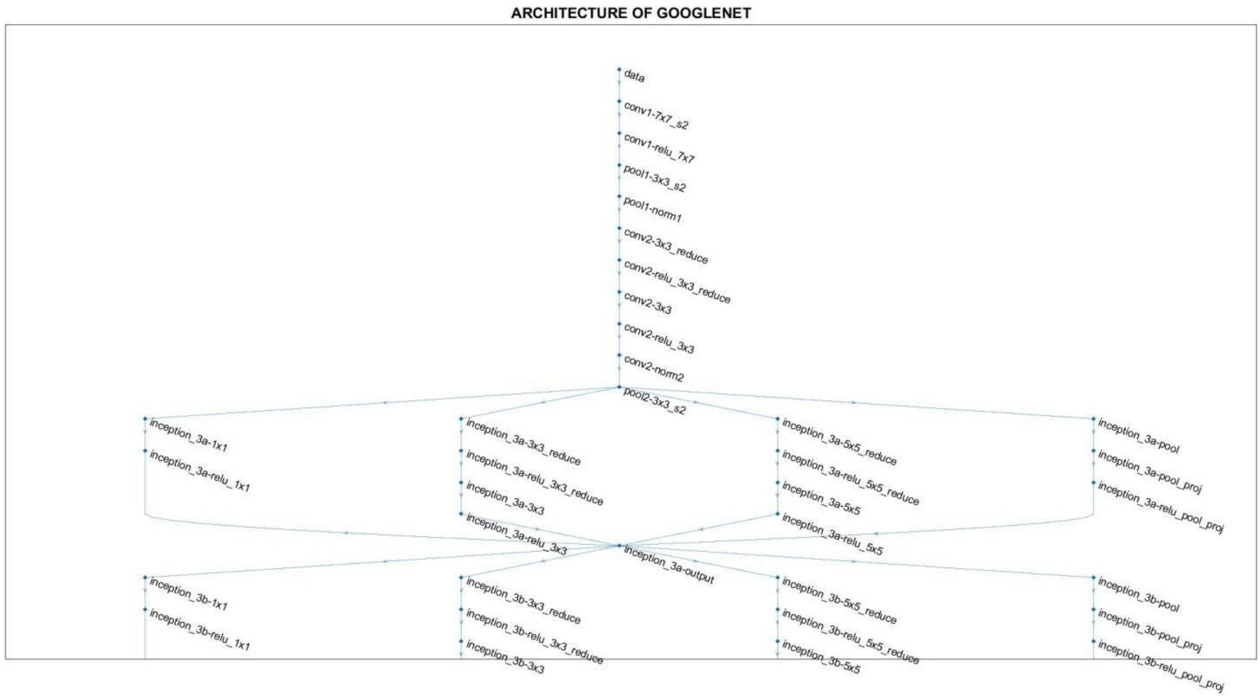


Fig7:Architecture of GoogleNet Performance Evaluation: Our projected model is functional in testing image datasets from different classes and training image datasets. We have used the net.Layer(1) function to depict the first convolutional layer weight. We can observe the weight of the first convolutional layer. We processed the features by using deep CNN. It combines the features to sketch the higher-level image features. A higher-level feature is suitable for image recognition tasks. Figure 8 given below represents the first convolution layer weight.

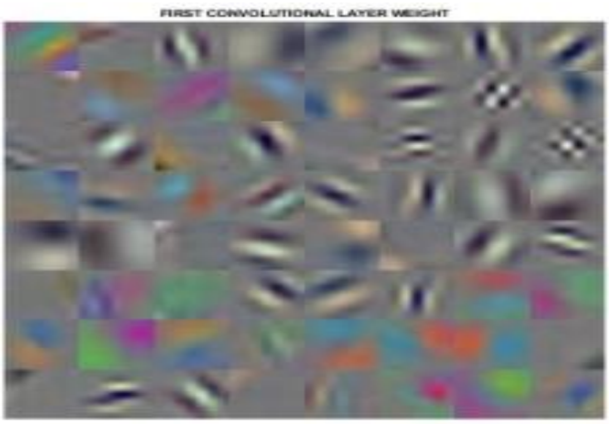


Fig 8: First Convolutional Neural Network

The execution of the projected technique is assessed by calculating the convolutional neural network models with dissimilar metrics. The performance of any learning algorithm is calculated by inspecting how better they perform on test data.

- **Receiver Operating Characteristic curve (ROC):** ROC is normally used for performance computation in the supervised image classification. It entirely depends on the relationship between the false positive rate and the true positive rate. We have used the area under the curve (AUC) for multiple classes to execute this task. It is the mean of the area under the curve values.

- **Sensitivity (Recall):** Sensitivity (Recall) of any classifier is the ratio between how many were correctly labeled as positive to how many were actually positive. Sensitivity can be calculated by using equation number 1.In equation number 1. A full form of TP is true positive. True positive cases are the number of occurrences, which are correctly accepted as positive. In equation number 1 full form of FN is a false negative. False-negative is the number of positive cases that are wrongly classified as negative.

$$\textbf{Sensitivity} = \frac{\boldsymbol{TP}}{\textbf{TP} + \textbf{FN}} \qquad (\mathbf{1})$$

- **True Negative Rate (Specificity):** True negative rate is the ratio of true negative cases that were classified as negative and the sum of true negative and false positive cases thus true negative rate is a measure of how correctly our classifier recognizes negative cases. The true negative rate is also known as specificity. The true negative rate is calculated by equation number 2, Here TN stands for true negatives and FP stands for false positive.

$$\textbf{True Negative Rate (Specificity)} = \frac{\textbf{TN}}{\textbf{TN} + \textbf{FP}} \qquad (2)$$

- **Accuracy:** Normally, Specificity, and sensitivity compute the usefulness of the algorithm on a class, positive and negative respectively. In general, for the assessment of classification performance, we compute the accuracy of classification. In the assessment stage, the accurateness of classification was computed every twenty iterations. This computation counts the percentage of the model that is correctly classified and it is obviously expressed in equation 3:

$$\textbf{Accuracy} = \frac{(\boldsymbol{TP} + \boldsymbol{TN})}{(\boldsymbol{TP} + \ \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN})} \qquad (3)$$

- **Precision:** For the evaluation of the analytical power of an algorithm, we have computed the precision value. Precision is the ratio of true positives and summation of false positives and true positives. Precision is expressed by equation 4.

$$\textbf{Precision} = \frac{(\textbf{TP})}{(\boldsymbol{TP} + \boldsymbol{FP}\ )} \qquad (4)$$

- **F_β Measure (F1 Score):** F1 score determines test's correctness. It is calculated from the sensitivity and precision of the test. F1 score is the harmonic mean of precision and sensitivity. The lowest probable value of the F1 score is 0 and the highest probable value of the F1 score is 1.When we will assume the value of β =0.5 then Fβ will be said as F1 score. It is calculated by equation number 5.

$$\boldsymbol{F_{\beta^1}} = (\mathbf{1} + \boldsymbol{\beta^2}\ ) * (\textbf{Precision} * \textbf{Recall}) /\{(\boldsymbol{\beta}^2 * \textbf{Precision}) + (\textbf{recall}) \qquad (5)$$

Result Analysis: We have chosen a 1030 image of the infected human chest for training and testing the model. We have used a function confusionmat ( ) to plot the confusion matrix. We have plotted confusion matrices for 80% training and 20 % testing data. The following figure represents the confusion matrix generated by GoogleNet for 80 % training and 20 % testing data.

Table 7 represents confusion matrix for different training % and testing %.

TABLE 7 :CONFUSION MATRIX FOR DIFFERENT TRAINING % & TESTING %

CM FOR 65 % TRAINING AND 35 % TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	51	0	0	0	3
B	1	51	0	0	2
C	3	1	47	0	3
D	0	0	0	54	0
E	17	1	2	1	33

CM FOR 70% TRAINING AND 30% TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	39	0	1	0	6
B	0	44	0	1	1
C	1	1	44	0	0
D	0	0	0	46	0
E	7	1	3	1	33

CM FOR 75% TRAINING AND 25% TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	25	0	4	0	9
B	0	37	0	0	1
C	0	0	38	0	0
D	0	0	0	38	0
E	2	5	2	0	29

CM FOR 80% TRAINING AND 20%TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	29	0	1	0	1
B	0	29	0	0	1
C	4	0	26	0	1
D	0	0	0	31	0
E	5	0	2	0	24

CM FOR 85 % TRAINING AND 15 % TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	18	0	2	0	3
B	0	22	0	0	1
C	0	1	21	0	1
D	0	0	0	23	0
E	1	2	1	0	19

CM FOR 90% TRAINING AND 10% TESTING					
CLASS	**A**	**B**	**C**	**D**	**E**
A	14	0	0	0	1
B	0	15	0	0	0
C	0	0	15	0	0
D	0	0	0	15	0
E	3	2	1	0	9

Performance of GoogleNet using confusion matrix: In this research paper, we have used a pre-trained deep convolutional neural network for the classification of COVID-19 disease among several other human chest diseases. The central objective of this research paper is to categorize COVID-19 infection using a deep CNN model with GoogleNet for measurement of the accuracy, sensitivity, F1-score, precision, specificity, and empiric ROC (Reciever operating characteristic curve) by fine-tuning. The performance of GoogleNet for different training 65% - 90% % and testing 35 % -10% is represented by table 8.

TABLE 8: PERFORMANCE CHART OF GoogleNet for DIFFERENT TRAINING % & TESTING %

65% TRAINING AND 35 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1SCORE	ACCURACY
A	0.9444	0.9222	0.7083	0.0778	0.8095	0.9259
B	0.9444	0.9815	0.9107	0.0185	0.9273	0.9753
C	0.9444	0.9815	0.9107	0.0185	0.9273	0.9753
D	1	0.9963	0.9818	0.0037	0.9908	0.9969
E	0.6111	0.9704	0.8049	0.0296	0.6947	0.9105

70% TRAINING AND 30 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1SCORE	ACCURACY
A	0.8298	0.9652	0.8298	0.0348	0.8298	0.9422
B	0.9565	0.9913	0.9565	0.0087	0.9565	0.9855
C	0.9167	0.9826	0.9167	0.0174	0.9167	0.9712
D	0.9583	0.9913	0.9583	0.0087	0.9583	0.9856
E	0.825	0.9696	0.825	0.0304	0.825	0.9481

75% TRAINING AND 25 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1 SCORE	ACCURACY
A	0.6579	0.9895	0.9259	0.0105	0.7692	0.9342
B	0.9737	0.9737	0.881	0.0263	0.925	0.9737
C	1	0.9684	0.8636	0.0316	0.9268	0.9737
D	1	1	1	0	1	1
E	0.7632	0.9474	0.7436	0.0526	0.7532	0.9167

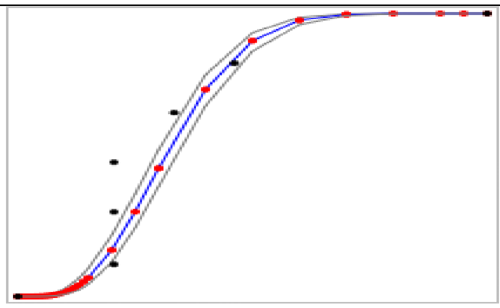
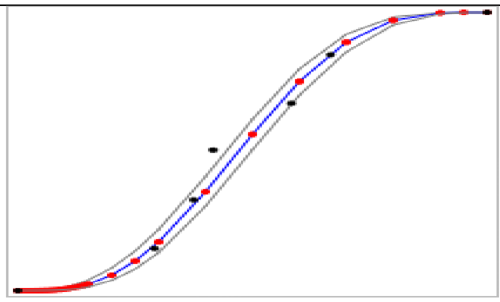
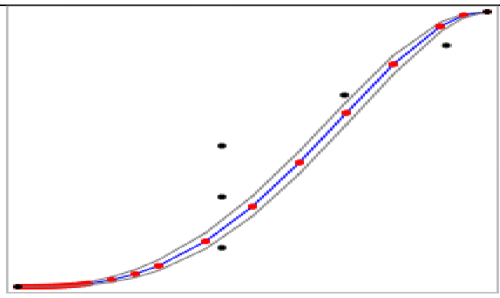
80% TRAINING AND 20 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1 SCORE	ACCURACY
A	0.9355	0.9427	0.7632	0.0573	0.8406	0.9415
B	0.9355	1	1	0	0.9667	0.9892
C	0.8387	0.9806	0.8966	0.0194	0.8667	0.957
D	1	1	1	0	1	1
E	0.7742	0.9806	0.8889	0.0194	0.8276	0.9462

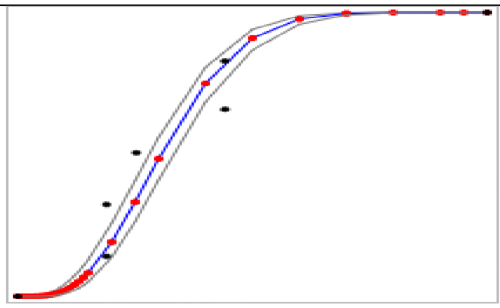
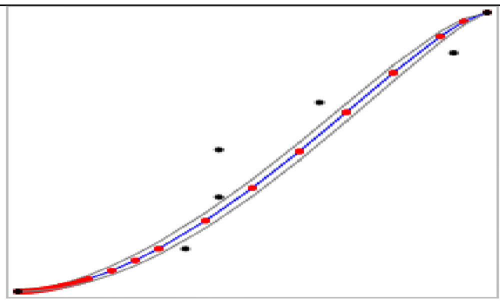
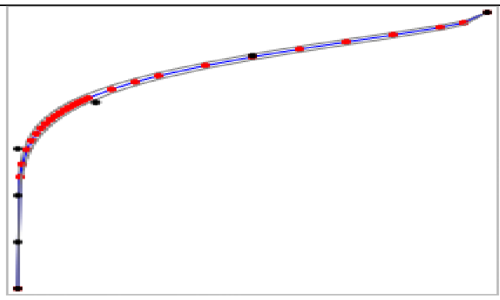
85% TRAINING AND 15 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1 SCORE	ACCURACY
A	0.7826	0.9913	0.9474	0.0087	0.8571	0.9565
B	0.9565	0.9652	0.8462	0.0348	0.898	0.9638
C	0.913	0.9739	0.875	0.0261	0.8936	0.9638
D	1	0.9913	0.9583	0.0087	0.9787	0.9928
E	0.8261	0.9565	0.7917	0.0435	0.8085	0.9348

90% TRAINING AND 10 % OF TESTING						
CLASS	SENSITIVITY	SPECIFICITY	PRECISION	FPR	F1 SCORE	ACCURACY
A	0.9333	0.96	0.8235	0.04	0.875	0.9556
B	1	0.9733	0.8824	0.0267	0.9375	0.9778
C	1	0.9867	0.9375	0.0133	0.9677	0.9889
D	1	1	1	0	1	1
E	1	0.9867	0.9	0.0133	0.9881	0.9881

Performance Graph of GoogleNet: The graph given below corresponds to the performance of GoogleNet with different training and different testing image datasets. In the graph represented below, we have explains several parameters which are specificity, sensitivity, precision, false-positive rate (FPR), F1 score, and ROC area. Table 7 given below represents the ROC curve for GoogleNet for 65%-90% training and 35%-10% testing.

TABLE 9 :EMPIRIC ROC AREA & ROC CURVE FOR DIFFERENT TAINING % & TESTING %

<table>
<tr><th>SR.NO</th><th>EMPIRIC ROC AREA</th><th colspan="3">FPR & TPR VALUE FOR DIFFENT CLASSES</th><th>ROC CURVE
(x axis =FPR and Y axis =TPR)</th></tr>
<tr><td rowspan="8">1</td><td rowspan="8">0.67</td><td colspan="3">FOR 65 %</td><td rowspan="8"></td></tr>
<tr><td>CLASS</td><td>FPR</td><td>TPR</td></tr>
<tr><td>A</td><td>0.0778</td><td>0.9444</td></tr>
<tr><td>B</td><td>0.0185</td><td>0.9444</td></tr>
<tr><td>C</td><td>0.0185</td><td>0.9444</td></tr>
<tr><td>D</td><td>0</td><td>0.9444</td></tr>
<tr><td>E</td><td>0</td><td>1</td></tr>
<tr><td>F</td><td>0.0296</td><td>0.6111</td></tr>
<tr><td rowspan="8">2</td><td rowspan="8">0.529</td><td colspan="3">FOR 70 %</td><td rowspan="8"></td></tr>
<tr><td>CLASS</td><td>FPR</td><td>TPR</td></tr>
<tr><td>A</td><td>0.0348</td><td>0.8298</td></tr>
<tr><td>B</td><td>0.0087</td><td>0.9565</td></tr>
<tr><td>C</td><td>0.0174</td><td>0.9167</td></tr>
<tr><td>D</td><td>0.0043</td><td>0.9787</td></tr>
<tr><td>E</td><td>0.0087</td><td>0.9583</td></tr>
<tr><td>F</td><td>0.0304</td><td>0.825</td></tr>
<tr><td rowspan="8">3</td><td rowspan="8">0.441</td><td colspan="3">FOR 75 %</td><td rowspan="8"></td></tr>
<tr><td>CLASS</td><td>FPR</td><td>TPR</td></tr>
<tr><td>A</td><td>0.0105</td><td>0.6579</td></tr>
<tr><td>B</td><td>0.0263</td><td>0.9737</td></tr>
<tr><td>C</td><td>0.0316</td><td>1</td></tr>
<tr><td>D</td><td>0</td><td>1</td></tr>
<tr><td>E</td><td>0</td><td>1</td></tr>
<tr><td>F</td><td>0.0526</td><td>0.7632</td></tr>
</table>

4	**0.661**	**FOR 80 %**			
		CLASS	**FPR**	**TPR**	
		A	0.0573	0.9355	
		B	0	0.9355	
		C	0.0194	0.8387	
		D	0.0065	1	
		E	0	1	
		F	0.0194	0.7742	
5	**0.457**	**FOR 85 %**			
		CLASS	**FPR**	**TPR**	
		A	0.0087	0.7826	
		B	0.0348	0.9565	
		C	0.0261	0.913	
		D	0	0.913	
		E	0.0087	1	
		F	0.0435	0.8261	
6	**0.821**	**FOR 90 %**			
		CLASS	**FPR**	**TPR**	
		A	0.04	0.9333	
		B	0.0267	1	
		C	0.0133	1	
		D	0	1	
		E	0	1	
		F	0	1	

The figure given in table 9 represents the different ROC curves for different training and different testing data. In the table given above FPR stands for false-positive rate and TPR stands for true positive rate. In the above empiric ROC graph, the minimum area covered by GoogleNet with the area under ROC is 0.441 on 75 % training and 25 % testing data, and the maximum area covered by GoogleNet with the area under ROC 0.821 on 90% training and 10 % testing.

Accuracy: The GoogleNet model showed alike and statistically considerable performance. Starting with accuracy, if we focus on the accuracy metric by considering 65% image data for training and 35 % for testing data, we have obtained the lower result for GoogleNet with an accuracy of 88.58 % in 2.16 minutes followed by 85 % of training and 15% of testing data with accuracy 89.86 % in 2.26 minutes. The maximum result is obtained, when there was 90 % of training and 10 % of testing the data with an accuracy of 92.22 % in 2.44 minutes only. Table 10 given below represents accuracy % different training % and different testing % of GoogleNet with total time elapsed. Figure 9 given below represents the overall accuracy and time elapsed (in second) different training or testing %.

TABLE 10:ACCURACY % & TIME ELAPSED BY GoogleNet FOR DIFFERENT TRAINING % AND TESTING %

SR.NO	TRAINING %	TESTING %	ACCURACY %	TIME ELAPSED (SEC)
1	65	35	88.58	129.6
2	70	30	91.30	155.4
3	75	25	89.91	135.6
4	80	20	91.40	149.4
5	85	15	89.86	135.6
6	90	10	92.22	146.4

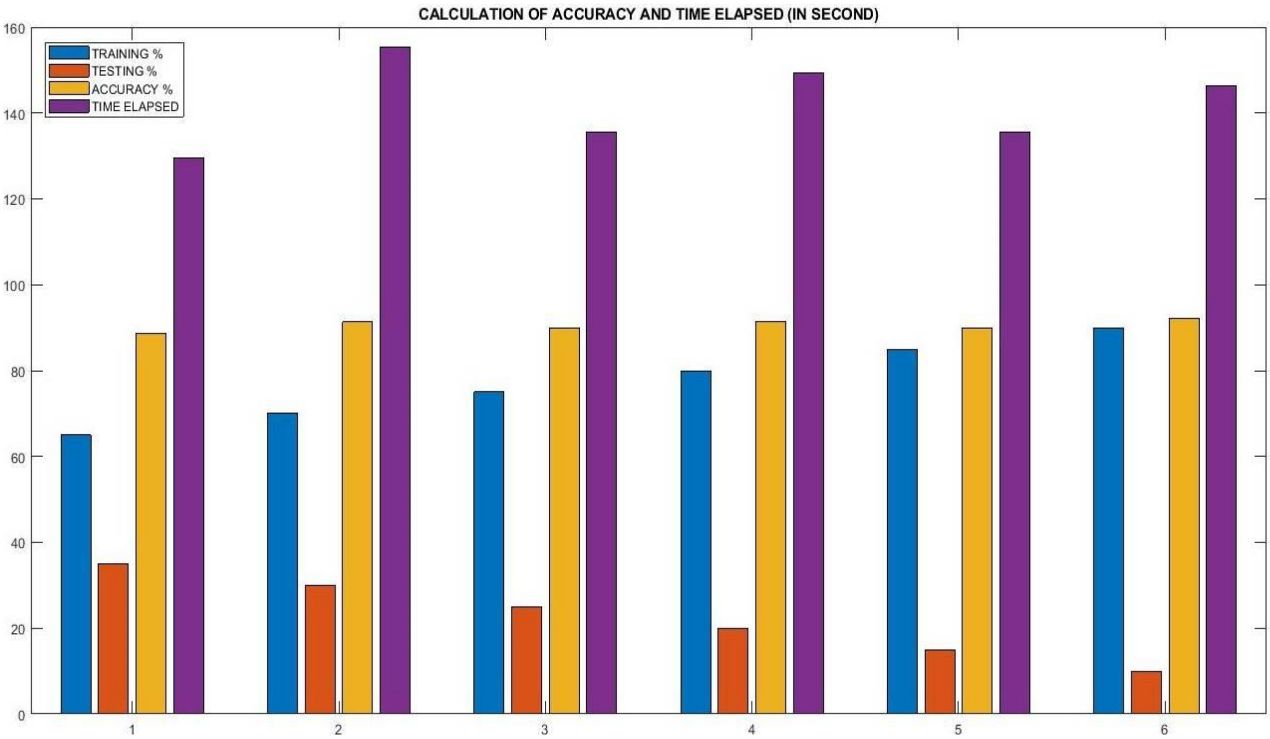


Fig 9: Graph for Accuracy & Time elapsed (in sec) for different training % and testing %

Speed: In addition, if we look for performance on behalf of the shortest time, then GoogleNet for 65 % training and 35 % testing showed a better speed in comparison to other training % and testing %. Note that it does not show a better result in the case of accuracy. So it is considered to scarify a greater processing efficiency in exchange for the minimum decrease. Table 8 represents the total time elapsed for speed comparison.

Conclusion and Future Work: In our projected research paper, we have proposed an algorithm for a convolutional neural network to build a classifier for chest disease categorization. The outcomes demonstrate that CNN drastically overcame the state of the art of this field. In addition, our experimentation reveals the advantage of using a CNN model. As we know, Chest infection is a serious disease. So we need to recognize the chest infection at its early stage to overcome this problem. We have to present a better solution by using novel methods like chest disease prevention and chest disease detection. These days, many researchers attain better accuracy by convolutional neural network models. They have faced a lot of problems to solve the chest disease. To resolve such type of difficulty, there are several classification techniques but convolutional neural networks are useful in the theoretical area as well as application area both. A new technique is used for the recognition and classification method to increase the power of deep convolutional neural networks. Our central objective is to identify the six most frequent diseases of the chest. It is more difficult and challenging to work with different chest disease images with different orientations, pose sizes, and resolutions. We have developed a method for better results after completing the training and testing of chest disease images. We can state from every part of the above experiences and experiments that with the help of our technique anyone can able to differentiate exaggerated chest disease correctly.

We have projected a technique in which we have classified six types of chest disease. For chest disease classification, we have used an algorithm that is CNN to construct a model chest disease dataset. In the future, our main aim will be to get more accurate results. We would like to provide a good answer for chest disease recognition by using an image dataset.

References:

1. Liu, N., Wan, L., Zhang, Y., Zhou, T., Huo, H., & Fang, T. (2018). Exploiting convolutional neural networks with deeply local description for remote sensing image classification. *IEEE access*, *6*, 11215-11228.
2. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.
3. Brunetti, A., Carnimeo, L., Trotta, G. F., & Bevilacqua, V. (2019). Computer-assisted frameworks for classification of liver, breast and blood neoplasias via neural networks: A survey based on medical images. Neurocomputing, 335, 274-298.
4. Asiri, N., Hussain, M., Al Adel, F., & Alzaidi, N. (2019). Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. Artificial intelligence in medicine, 99, 101701.
5. Zhou, T., Thung, K. H., Zhu, X., & Shen, D. (2019). Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis. Human brain mapping, 40(3), 1001-1016.
6. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2017). Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE journal of biomedical and health informatics, 22(5), 1589-1604.
7. Chouhan, V., Singh, S. K., Khamparia, A., Gupta, D., Tiwari, P., Moreira, C., ... & De Albuquerque, V. H. C. (2020). A novel transfer learning based approach for pneumonia detection in chest X-ray images. Applied Sciences, 10(2), 559.
8. Malūkas, U., Maskeliūnas, R., Damaševičius, R., & Woźniak, M. (2018). Real time path finding for assisted living using deep learning. J. Univ. Comput. Sci, 24, 475-487.
9. Zhang, X., Yao, L., Wang, X., Monaghan, J., Mcalpine, D., & Zhang, Y. (2019). A survey on deep learning based brain computer interface: Recent advances and new frontiers. arXiv preprint arXiv:1905.04149.
10. Bakator, M., & Radosav, D. (2018). Deep learning and medical diagnosis: A review of literature. Multimodal Technologies and Interaction, 2(3), 47.
11. Roosa, K., Lee, Y., Luo, R., Kirpich, A., Rothenberg, R., Hyman, J. M., ... & Chowell, G. (2020). Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. Infectious Disease Modelling, 5, 256-263.
12. Yan, L., Zhang, H. T., Goncalves, J., Xiao, Y., Wang, M., Guo, Y., ... & Huang, X. (2020). An interpretable mortality prediction model for COVID-19 patients. Nature Machine Intelligence, 1-6.
13. Paules, C. I., Marston, H. D., & Fauci, A. S. (2020). Coronavirus infections—more than just the common cold. Jama, 323(8), 707-708.
14. Chen, Y., Liu, Q., & Guo, D. (2020). Emerging coronaviruses: genome structure, replication, and pathogenesis. Journal of medical virology, 92(4), 418-423.

15. Zhu, N., Zhang, D., Wang, W., Li, X., Yang, B., Song, J., ... & Niu, P. (2020). A novel coronavirus from patients with pneumonia in China, 2019. New England Journal of Medicine.

16. Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., ... & Xing, X. (2020). Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. New England Journal of Medicine.

17. Cohen, J., & Normile, D. (2020). New SARS-like virus in China triggers alarm.

18. Hemdan, E. E. D., Shouman, M. A., & Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. arXiv preprint arXiv:2003.11055.

19. Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., ... & Agha, R. (2020). World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). International Journal of Surgery.

20. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/media-resources/press-briefings.

21. Gilani, Z., Kwong, Y. D., Levine, O. S., Deloria-Knoll, M., Scott, J. A. G., O'Brien, K. L., & Feikin, D. R. (2012). A literature review and survey of childhood pneumonia etiology studies: 2000–2010. Clinical Infectious Diseases, 54(suppl_2), S102-S108.

22. Bouch, C., & Williams, G. (2006). Recently published papers: pneumonia, hypothermia and the elderly. Critical Care, 10(5), 167.

23. Scott, J. A. G., Brooks, W. A., Peiris, J. M., Holtzman, D., & Mulholland, E. K. (2008). Pneumonia research to reduce childhood mortality in the developing world. The Journal of clinical investigation, 118(4), 1291-1300.

24. Wunderink, R. G., & Waterer, G. (2017). Advances in the causes and management of community acquired pneumonia in adults. Bmj, 358, j2471.

25. https://www.kaggle.com/saife245/tuberculosis-image-datasets.

26. Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2015, December). Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In International Symposium on Visual Computing (pp. 638-645). Springer, Cham.

27. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.

28. Prasanna Mohanty, S., Hughes, D., & Salathe, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. arXiv, arXiv-1604.. [CrossRef]

29. Nachtigall, L. G., Araujo, R. M., & Nachtigall, G. R. (2016, November). Classification of apple tree disorders using convolutional neural networks. In 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 472-476). IEEE.

30. Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016, December). Basic investigation on a robust and practical plant diagnostic system. In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 989-992). IEEE.

31. Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. Applied Artificial Intelligence, 31(4), 299-315.

32. DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., ... & Lipson, H. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology, 107(11), 1426-1432.

33. Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. Neurocomputing, 267, 378-384.

34. Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. Computational intelligence and neuroscience, 2017.

35. Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases:

detection and saliency map visualisation. In Human and machine learning (pp. 93-117). Springer, Cham.

36. Wang, J., Chen, L., Zhang, J., Yuan, Y., Li, M., & Zeng, W. (2018, April). Cnn transfer learning for automatic image-based classification of crop disease. In Chinese Conference on Image and Graphics Technologies (pp. 319-329). Springer, Singapore.

37. Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.

38. Khandelwal, I., & Raman, S. (2019). Analysis of transfer and residual learning for detecting plant diseases using images of leaves. In Computational Intelligence: Theories, Applications and Future Directions-Volume II (pp. 295-306). Springer, Singapore.

39. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

40. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., ... & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677.

41. Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489.

42. Akiba, T., Suzuki, S., & Fukuda, K. (2017). Extremely large minibatch sgd: Training resnet-50 on imagenet in 15 minutes. arXiv preprint arXiv:1711.04325..

43. Jia, X., Song, S., He, W., Wang, Y., Rong, H., Zhou, F., ... & Chen, T. (2018). Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. arXiv preprint arXiv:1807.11205.

44. Ying, C., Kumar, S., Chen, D., Wang, T., & Cheng, Y. (2018). Image classification at supercomputer scale. arXiv preprint arXiv:1811.06992.

45. Mikami, H., Suganuma, H., Tanaka, Y., & Kageyama, Y. (2018). Massively distributed SGD: ImageNet/ResNet-50 training in a flash. arXiv preprint arXiv:1811.05233.

46. Narin, A., Kaya, C., & Pamuk, Z. (2020). Automatic detection of coronavirus disease(covid-19) using x-ray images and deep convolutional neural networks. arXiv preprint arXiv:2003.10849.

47. Abbas, A., Abdelsamea, M. M., & Gaber, M. M. (2020). Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. arXiv preprint arXiv:2003.13815.

48. Zhang, J., Xie, Y., Li, Y., Shen, C., & Xia, Y. (2020). Covid-19 screening on chest x-ray images using deep learning based anomaly detection. arXiv preprint arXiv:2003.12338.

49. Douarre, C., Schielein, R., Frindel, C., Gerth, S., & Rousseau, D. (2018). Transfer learning from synthetic data applied to soil–root segmentation in x-ray tomography images. Journal of Imaging, 4(5), 65.

50. Zhang, Y., Wang, G., Li, M., & Han, S. (2018). Automated classification analysis of geological structures based on images data and deep learning model. Applied Sciences, 8(12), 2493.

51. Wang, Y., Wang, C., & Zhang, H. (2018). Ship classification in high-resolution SAR images using deep learning of small datasets. Sensors, 18(9), 2929.

52. Sun, C., Yang, Y., Wen, C., Xie, K., & Wen, F. (2018). Voiceprint identification for limited dataset using the deep migration hybrid model based on transfer learning. Sensors, 18(7), 2399.

53. Chen, Z., Zhang, Y., Ouyang, C., Zhang, F., & Ma, J. (2018). Automated landslides detection for mountain cities using multi-temporal remote sensing imagery. Sensors, 18(3), 821.

54. Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. In Classification in BioApps (pp. 323-350). Springer, Cham.

55. Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. Annual review of biomedical engineering, 19, 221-248.

56. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. nature, 542(7639), 115-118.

57. Milletari, F., Navab, N., & Ahmadi, S. A. (2016, October). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 fourth international conference on 3D vision (3DV) (pp. 565-571). IEEE.

58. Grewal, M., Srivastava, M. M., Kumar, P., & Varadarajan, S. (2018, April). Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018) (pp. 281-284). IEEE.

59. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... & Kim, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama, 316(22), 2402-2410.

60. Bar, Y., Diamant, I., Wolf, L., Lieberman, S., Konen, E., & Greenspan, H. (2015, April). Chest pathology detection using deep learning with non-medical training. In 2015 IEEE 12th international symposium on biomedical imaging (ISBI) (pp. 294-297). IEEE.

61. Avni, U., Greenspan, H., Konen, E., Sharon, M., & Goldberger, J. (2010). X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words. IEEE Transactions on Medical Imaging, 30(3), 733-746.

62. Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R. H., Reither, K., Breuninger, M., ... & Sánchez, C. I. (2014). A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest x-rays. IEEE transactions on medical imaging, 34(1), 179-192.

63. Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., ... & Thoma, G. (2013). Automatic tuberculosis screening using chest radiographs. IEEE transactions on medical imaging, 33(2), 233-245.

64. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507

65. Ranzato, M. A., Poultney, C., Chopra, S., & Cun, Y. L. (2007). Efficient learning of sparse representations with an energy-based model. In Advances in neural information processing systems (pp. 1137-1144).

66. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).

67. Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In Icml.

68. Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011, June). Stacked convolutional auto-encoders for hierarchical feature extraction. In International conference on artificial neural networks (pp. 52-59). Springer, Berlin, Heidelberg.

69. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

70. Salakhutdinov, R., & Hinton, G. (2009, April). Deep boltzmann machines. In Artificial intelligence and statistics (pp. 448-455).

71. Younes, L. (1999). On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. Stochastics: An International Journal of Probability and Stochastic Processes, 65(3-4), 177-228.

72. Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2), 270-280.

73. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

74. Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1), 106.

75. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).

76. Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In Icml.

77. Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011, June). Stacked convolutional auto-encoders for hierarchical feature extraction. In International conference on artificial neural networks (pp. 52-59). Springer, Berlin, Heidelberg.

78. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

79. Ghazi, M. M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing, 235, 228-235.].