PAPER • OPEN ACCESS

Covid-19 Detection from Chest X-Ray using Convolution Neural Networks

To cite this article: Pillalamarry Mahesh *et al* 2021 *J. Phys.: Conf. Ser.* **1804** 012197

Covid-19 Detection from Chest X-Ray using Convolution Neural Networks

Pillalamarry Mahesh[1], Yakkala Gnana Prathyusha[1*], Botlagunta Sahithi[1], S Nagendram[2]**

[1]B. Tech Students, Department of Electronics and Communication Engineering, K L E F, Guntur, A.P, India
[2]Associate Professor, Department of Electronics and Communication Engineering, K L E F, Guntur, A.P, India

reena1286@gmail.com

Abstract. A corona virus has infected more than 36,087,836 people and 1,055,387 Deaths since December 2019. As it rapidly spreads across the planet, scientists and public-health experts are racing to slow down the spreading and trying to find methodologies to detect it. To do that, they need to understand the new virus. It's called severe acute respiratory syndrome coronavirus 2, or SARS-CoV-2. There are different ways to diagnose the COVID-19, but they are cost-effective and increasing the time taken to produce, buy using chest x-ray we can reduce cost and result in time. But to diagnose x-ray's we need expert radiotherapists. Thus, we developed a model that automatically detect COVID and non-COVID X-rays. These days Deep Learning algorithms showing the foremost results in Disease classification. Also, features learned by pre-trained Convolution Neural Networks (CNN) models on large-scale datasets are much useful in image classification tasks. we train and test our model to analyze the images as COVID or normal. we analytically determine the optimal CNN model for the purpose. The accuracy metrics are used to validate the classification of the model.

1. Introduction

The x-rays contain very small details, to examine that we need an expert radiotherapist. Even though some cases are a failure because of eyesight.

Fig. A

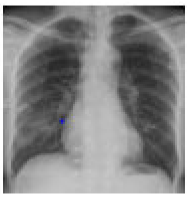

Fig. B

Figure 1. Fig A was a Normal chest image and Fig B was COVID affected chest image.

From the above two Figure 1. We can't find which image is COVID and normal. But the computer vision can classify them by training with some labeled data. Thus, we use deep learning [1]. computer vision is one of the areas that have been advancing. Deep Learning computer vision is now helping in image recognizing and also in many image classifications tasks. deep Learning requires a labeled dataset. For example, in our project, we need to find whether the x-ray image is COVID or normal, for that we need to train the model by giving two image sets COVID and normal as labeled. Also, deep learning required high GPU or CPU because of its high computations. Nowadays all systems are with the best CPU and GPU or also we can use cloud IDEs to perform tests and train of the model.

Convolution neural network is one of the deep learning techniqueswith wide and deep structure Figure 2. Where pixels extracted from images and transfer those pixels to neural networks. we process the pixels of the image for less complexity of the model.

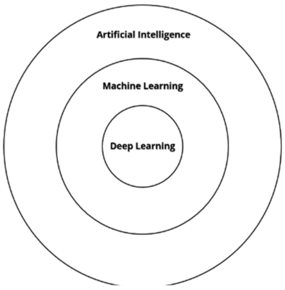

Figure 2. Relation between Artificial intelligence, Machine Learning and Deep Learning

1.1 Why CNN

Convolution Neural Networks (CNN) extract the features of an image and reduce the size without loss of its characteristics. For example, if we have an image with a size of 35x35x1(where 1 represents only 1 channel in RGB). The total number of Neurons for the fully connected layer will be 35*35= 1125. if the image size is 100x100 we need 10000 neurons as input for the fully connected layer, where it was very difficult to compute that many layers [2]. But by using Convolution neural networks (CNN) we can reduce the number of neurons with the help of the conv and pooling layers, where we can reduce the number of computations in a fully connected network.

1.2 Image Representation:

The images are represented as pixels in image processing, the pixels are packed in matrices as height x width x number of channels (for colored image 3 channels RGB as in Figure 3). Based on the red, green, and Green channels intensity the pixel values may vary. for the grayscale images, there will be only one channel.

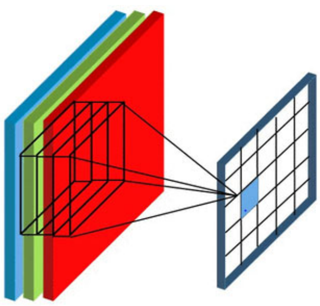

Figure 3. RGB channels representation

1.3 Edge Detection

The edge detection is the main building block for neural networks because by the edges the object will be identified. Mainly we have horizontal, vertical, and diagonal edge detection. based on the given kernel/filter the edges are identified [3]. this edge detection will be seen in the convolution layer.

1.4 Layers

In Convolution Neural Networks (CNN) we have 5 Layers. they are the input layer, convolution layer, pooling layer, Fully connected layer, and output layer.

1.4.1 Input layer: The input layer contains image data. the images are given in a three-dimensional matrix, we need to reshape the images as a single dimension. suppose if we have an image size of 30x30x1=900 then we need to reshape as 900x1. If you have "n" training examples then the dimension of input will be (900, n).

1.4.2 Convo Layer: The convo layer is called a feature extractor because the image features have extracted within the layer. The edge detection has to be done in the convo layer [4]. The convolution operation has been taken between image and kernel. the kernel is either a horizontal edge filter, vertical edge filter, or diagonal edge filter. Based on the filter the edges will detect. first, A part of the image is convoluted with the kernel, based on kernel size and same as sliding window or dot product total image will convolve as shown in Figure 4.

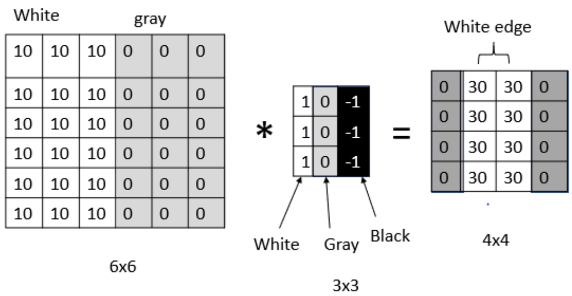

Figure 4: Convo operation with horizontal edge detection

$$(N \times N) * (F \times F) = (N-F+1) \times (N-F+1)$$

Here the *N x N* is an Image size and *F x F* is the Filter size a kernel size. By convoluting both results an image size (N-F+1) x (N-F+1). Like the above example, the input size is *6 * 6*, the kernel size is *3* 3* the output size is *4 * 4*.

There are also different types of filters like the Sobel filter, Scharr filter, etc., based on the requirement we use them.

1.4.3 Pooling layer: The pooling layer is in between two convo layers, used to reduce the size of the image which is the output of the convo layer. Without the pooling layer, the input for a fully connected network is very computationally expensive, we need to reduce it with the pooling layer [5]. the max-pooling is the only way to reduce the size of images. In max-pooling, we use stride (Stride denoted how many steps we need to move for each iteration, by default the stride is set to 1) as in Figure 5.

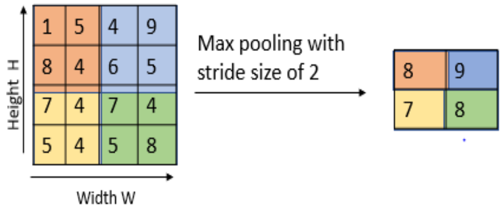

Figure 5. Max pooled with a filter size of 2 x 2 and with a stride of 2

$$W2=(W1-F)/S+1$$

$$H2=(H1-F)/S+1$$

$$D2=D1$$

Where *W1, H1, D1* is width, Height, the dimension of the image, and *W2, H2, D2* are the width, Height, dimension of the max pooled image.

1.4.4 Fully Connected Layer: The fully connected layer is used to classify the images between several images by training.it contains neurons, weights, and bias. in a fully connected layer, we have few layers, all the neurons are interconnected.

The Last layer Contains the Logistic/SoftMax Layer, basically logistic is used for binary classification and SoftMax is for multi-classification.

1.4.5 Output Layer: The Output layer presents the obtained results in binary form.

2. Dataset preparation

The Dataset used in this model contains two folders COVID and Normal. The COVID images obtained from a GitHub respiratory contains a total of 1,140 images which are a collection of Bacterial, viral, Chlamydophila, E. coli, Fungal, COVID, Influenza, Klebsiella, Legionella, Lipoid, MERS, Mycoplasma, No Finding, Pneumocystis, pneumonia, SARS, Strecoccus, Varicella, and Viral. For COVID images there are two views PA and AP views, we need only PA view, we have extracted 183 PA (PA view is like a front view) view COVID images.

The Normal images were obtained from the Pneumonia dataset found on the Kaggle website. There are more than 500 images but we need the images to count equal to COVID images. So, we took 183 normal images randomly from the pneumonia normal dataset as in Figure 6.

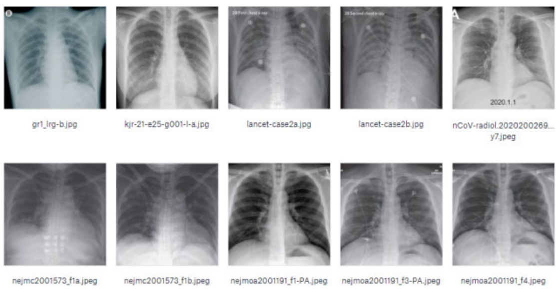

Figure 6. Example of Dataset Images

After the extraction of images from the original datasets, we need to divide the datasets as test and train. Train set with 80% of images and test as 20% images. we have prepared the dataset with 152 COVID, 152 normal for training, and 33 COVID, 33 Normal for testing [6].

3. Related Work

3.1 image classification and object detection by Deep Learning

Several studies have used Convolutional Neural Networks (CNNs) for the problem of image classification in the literature, most of which create different architectures for the neural networks. Convolutional Neural Networks is one of the best deep learning algorithms. In Recent times we use Convolutional Neural Networks frequently in self-driving cars, face recognition, image classifications for detection, etc.

The evolution of Convolutional Neural Networks is gradually increasing in the past few years. Were in every category like medical, automobiles, mobile technology, etc., we use Convolutional Neural Networks for image processing, image classification, and object detection

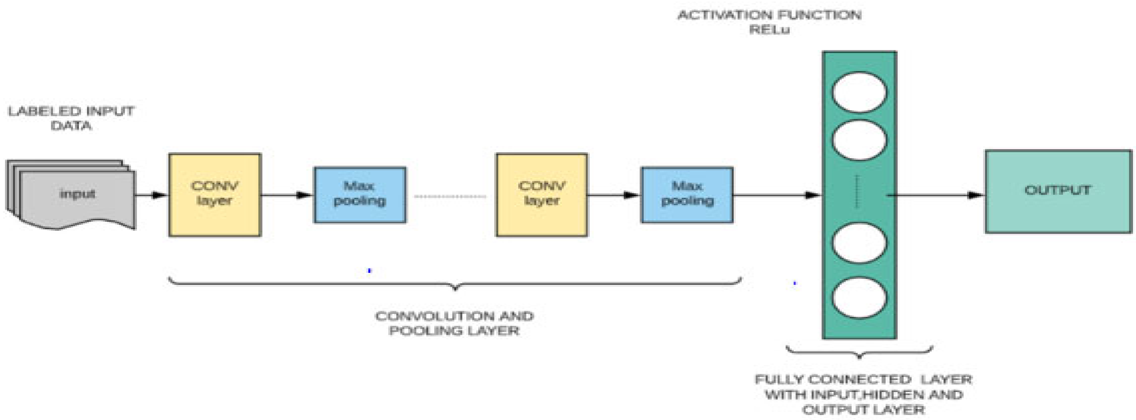

Figure 7 : Block diagram

[7]. there are different types of architectures in Convolutional Neural Networks they are VGG Net (Visual Geometry Group), ResNet (Residual Network), Dense Net, Inception Net. based on the different customs we make use of those architectures.

3.2 CNN approaches in the medical field

Deep learning is top-notch at image work that many are used to create medical images, not only for creating and also for image classification and detection.

The Convolution Neural Networks performs the best accuracy for the Pneumonia detection with chest Xray images. The results show with above 95% accuracy [8]. Not only Pneumonia the Convolution Neural Networks shows good results in other chest problems like bacterial, viruses, viral, etc.,

Not Only Chest related, but also for many diseases or diagnose by the convolution Neural Networks achieves up to the mark, by showing the best results.

4. Proposed Method

The main grail of the model is 1. To explore the deep learning algorithm to distinguish the Xray's of COVID.2. to promote the discovery of patterns in such X-rays via CNN interpretability algorithms 3. To achieve the best accuracy.

Generally, the Relu (Rectified linear unit) activation function was used in the Convolution Layer. The activation function was used after the Convolution layer/ Convo layer to rise the non-linearity in the output figure 8.

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

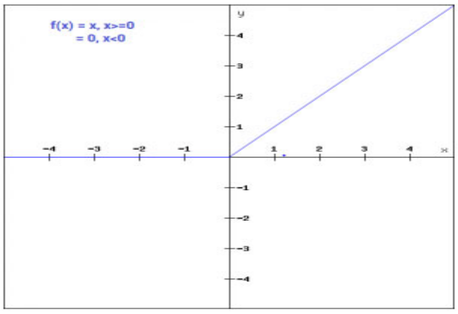

Figure 8. Relu activation graph

The sigmoid and Relu activation functions are used in the Fully Connected layer. Where sigmoid function was used for 0 or 1 classification as shown in Figure 8. In our model we need only 0 or 1 classification then the sigmoid function was the right choice [9].

$$y = \frac{1}{1 + e^{-x}}.$$

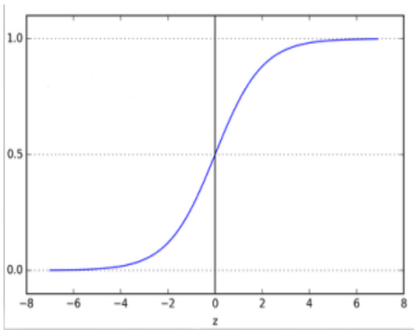

Figure 9. Sigmoid function

Initially, we need to prepare the data set as stated in *Dataset Preparation*. In this model the basic Convolution Neural Networks structure had used as Input layer -> Convolution Layer -> Max pooling layer -> Convolution layer -> …………….. -> Fully Connected Layer -> Output Layer as shown in Figure 7. In this model, the libraries used are NumPy, matplotlib, Keras, Keras. layers, Keras. Models, Keras. Preprocessing [10]. which are machine learning python libraries used to build the model.

To train the model First, in the input layer *train generator*, we reshaping any size of an image to *224,224,3*with the batch size of 32(The batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated, it makes the gradient descent accurate), the class mode was binary, shear range of 0.2, zoom range 0.2 and horizontal flip is true [11]. Then after four convolution layers with the first 2D convolution layer as 32 filters, filter size of *3 x 3*, and activation function was Relu (Rectified linear unit) [12]. shadowed by the next 2D convolution layer of 64 filters as the filter /kernel size of *3 x 3* with 2D max-pooling and dropout (Dropout used to prevent the network from overfitting) of *0.25.* The 3rd 2D convolution layer contains the same as the second 2D Convolution layer [13]. the same parameters are at the next two 2D Convolution layers contains the same parameters but there was a change in the numbers of filters. In the 4th 2D convolution layer, the number of filters was 128, all the kernel size and activation function are identical [14].

The next Layer was the Dense or Fully connected layer, where we used Adam Optimizer (Adam optimizer is used to update network weights based on training data) [15]. In the dense layer, we have 64 neurons with 0.5 dropouts. The detailed layers of information are given below in figure 10.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 220, 220, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 110, 110, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 108, 108, 64)      36928
_________________________________________________________________
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 86528)             0
_________________________________________________________________
dense (Dense)                (None, 64)                5537856
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 5,668,097
Trainable params: 5,668,097
Non-trainable params: 0
```

Figure 10. Model layers information

The training results two indices signifying {'COVID':0,'NORMAL':1}. after the training, we need to test the model for accuracy metrics. for testing we created a function validation *generator* which contains input image size and batch size same as *train generator.* for testing we used stepsperepoch=8 (For every iteration the number of epochs is considered as steps per epoch), epoch=10, validation steps=2(Similar to stepsperepoch but validation steps are for validation/test data) [16].

5. Results & Discussion

In this section, the Accuracy metrics of Convolutional Neural Networks (CNNs) are discussed. As in the above section we successfully trained and tested the model.

```
- loss: 1.2806 - accuracy: 0.5875 - val_loss: 0.6839 - val_accuracy: 0.5000

- loss: 0.5961 - accuracy: 0.7042 - val_loss: 0.5820 - val_accuracy: 0.9464

- loss: 0.4557 - accuracy: 0.8083 - val_loss: 0.4465 - val_accuracy: 0.9107

- loss: 0.3821 - accuracy: 0.8333 - val_loss: 0.3632 - val_accuracy: 0.9643

- loss: 0.2826 - accuracy: 0.8792 - val_loss: 0.2253 - val_accuracy: 0.9286
```

```
- loss: 0.2346 - accuracy: 0.9125 - val_loss: 0.1427 - val_accuracy: 0.9643

- loss: 0.1935 - accuracy: 0.9453 - val_loss: 0.1584 - val_accuracy: 0.9464

- loss: 0.1823 - accuracy: 0.9500 - val_loss: 0.1253 - val_accuracy: 0.9821

- loss: 0.1440 - accuracy: 0.9542 - val_loss: 0.1312 - val_accuracy: 0.9821

- loss: 0.1408 - accuracy: 0.9531 - val_loss: 0.0772 - val_accuracy: 0.9821
```

Figure 11. Test loss, test accuracy, train loss, train accuracy obtained after testing

After validation/test we got the validation accuracy of 98%from the starting to ending of training the validation accuracy was improved. The loss percentage decreased gradually.

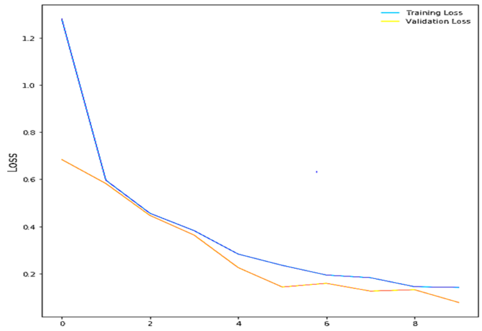

Figure 12. Graph of training loss vs validation loss

The above Figure 12 indicates a gradual decrease in loss per step in validation. The blue line indicates the training Loss and the Orange line indicates validation loss.

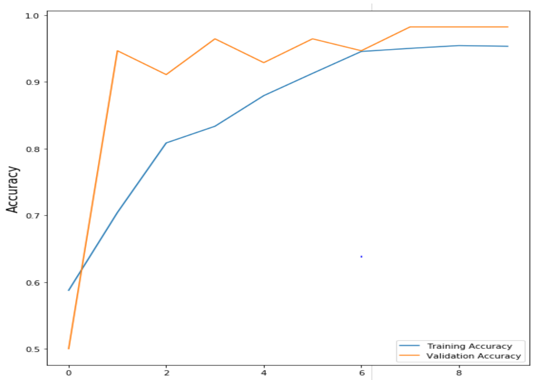

Figure 13. Training Accuracy vs Validation Accuracy

The above Figure 13 shows the Accuracy of our mode. Which shows the best results. The blue lines indicate the training accuracy and the orange line indicates validation accuracy. were the best validation accuracy being 98% and the best training accuracy was 95%. This happens after you use Dropout, since the behavior once training and testing is completely different. We have also applied the confusion matrix to our model. The confusion matrix is one of the best accuracy matrices. the confusion matrix is an easier way to measure the performance of the classification problem. A confusion matrix is a two-dimensional table with actual and predicted values containing "True Positives (TP)", "True Negatives (TN)", "False Positives (FP)", "False Negatives (FN)" as shows in figure 14.

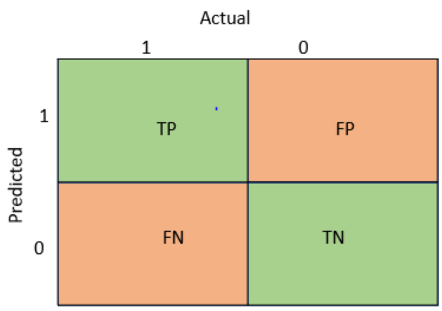

Figure 14. Confusion Matrix

The TP (True Positives) and TN (True Negatives) need to be a higher number because those only indicates correct classification. The remaining FP and FN need to be very less number because Those are false predictions.

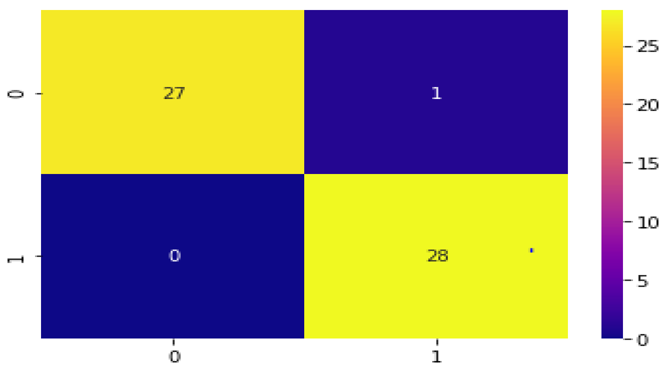

Figure 15. Confusion matrix of this Model

We have given 28 COVID images and 28 Normal images for the validation dataset, the model correctly predicted 27 images as COVID. Were 1 image as False positive. Out of 28 normal images, the model predicted 28 images as normal as shown Figure 15.

The model got trained and tested, which results in 95% training accuracy. which shows the best results. We have also tested the model by giving individual images, and it shows top-notch results.

6. Conclusion and Future Work

In this model, a study was conducted and presented how Convolutional Neural Networks (CNN's) shows top-notch results in the detection of COVID images. Research findings indicated that Convolutional Neural Networks have the latent to detect respiratory diseases with the best accuracy, although a large amount images are needed, achieves an accuracy of 95% train accuracy and 98% of validation accuracy. By improving the networks, we can achieve 100% accuracy. The future work for this model is to develop a web page or to build a mobile app, where all the public can use. And also, we can add GradCam techniques to visually see the COVID infected areas for better results. This Model proves that Convolutional Neural Networks can able to work superlatively in the medical field also.

References

[1] Kishore, P. V. V.; Kumar, K. V. V.; Kumar, E. Kiran; Sastry, A. S. C. S.; Kiran, M. Teja; Kumar, D. Anil; Prasad, M. V. D, "Indian Classical Dance Action Identification and Classification with Convolutional Neural Networks", ADVANCESS IN MULTIMEDIA, Hindawi, January 2018, India.

[2] Basha, C. M. A. K. Zeelan; Padmaja, Maruthi; Balaji, G. N, "Computer-Aided Fracture Detection System", JOURNAL OF MEDICAL IMAGING AND HEALTH INFORMATICS, March 2018, India.

[3] AyishaShamna.KK, Jamsheera. K, Shameena.P, "CNN Based Landmark Detection and Alzheimer's Diagnosis Using Landmark Feature", Cochin College of Engineering and Technology, Velachery, Kerala, February 2018, India.

[4] Yadav, S.S., Jadhav, S.M., "Deep convolutional neural network based medical image classification for disease diagnosis", Dr. Babasaheb Ambedkar Technological University, Raigad, Lorene, December 2019, India.

[5] Richa Bhatia is a seasoned journalist, "Advances in object recognition, image captioning, semantic segmentation and human pose estimation", Times of India, September 2017, India.

[6] Okeke Stephen, MangalSain, Uchenna Joseph Maduh, Do-Un Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare", Department of Civil and Computer Engineering, Dongseo University, Busan, March 2019, Korea.

[7] Rahul Chauhan; Kamal Kumar Ghanshala; R.C Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition" ,Graphic Era University, Dehradun, December 2018, India.

[8] Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi, "Understanding of a convolutional neural network", Department of Computer Engineering, Istanbul Kemerburgaz University, Istanbul, Turkey, August 2017.

[9] DimpyVarshni; Lucky Agarwal; Kartik Thakral; Rahul Nijhawan; Ankush Mittal, "Pneumonia Detection Using CNN based Feature Extraction", College of Engineering Roorkee, Graphic Era University Dehradun, August 2020, India.

[10] NiZhanga; Yi-XinCaia; Yong-YongWanga; Yi-TaoTiana; Xiao-LiWangb; BenjaminBadamic, "Skin cancer diagnosis based on optimized convolutional neural network", Department of Throracic Surgery, Hubei 430030, November 2019, China.

[11] HebaMohsena; El-Sayed A.El-Dahshanbc; El-Sayed M.El-Horbatyd; Abdel-BadeehM.Salemd , "Classification using deep learning neural networks for brain tumors" , Faculty of Computers and Information Technology, Future University, Cairo, Egyptian E-Learning University, Giza, December 2017, Egypt.

[12] MaayanFrid-Adara; IditDiamanta; EyalKlangb; MichalAmitaib; JacobGoldbergerc; HayitGreenspana, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification", Department of Biomedical Engineering, Tel Aviv University, September 2018, Israel.

[13] DhwaaniParikh; Vineet Menon, "Machine Learning Applied to Cervical Cancer Data", Association of Modern Education and Computer Science, January 2019, India.

[14] Hongwei Lin ; Kai Sun; Zheng-Hua Tan; Chengxi Liu; Josep M. Guerrero; Juan C. Vasquez, "Adaptive Protection Combined with Machine Learning for Microgrids", January 2019, China.

[15] Luca Abel, "Classification and Prediction of Acute Stress-Induced Response Patterns", Bachelor's Thesis in Medical Engineering, January 2019, Malsch.

[16] Rohan Bhardwaj ; Ankita R. Nambiar ; Debojyoti Dutta, "A Study of Machine Learning in Healthcare", Cisco Systems Inc, San Jose, CA, July 2017, USA.