



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 6, June 2021

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 7.512**

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)



# Increasing Accuracy of Classification Using CNN and Random Forest Classifier for Classification of X-RAY Image of Pneumonia, Covid, Lung Opacity Patients

Naumanurrahman Shaikh Mujiburrahman, Dr. Manoj E. Patil

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

Associate Professor, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

**ABSTRACT:** In this paper we have successfully implemented the ensemble method using CNN and Random Forest for increasing the accuracy and decreasing the loss when we only use CNN. We have created this model to classify the X ray image of patient diagnosed with Pneumonia vs patient diagnosed with Covid Or patients Diagnosed with Lung Opacity and Normal Lungs we have used the data set of X rays images of patients which are labeled Pneumonia, Covid, Lung Opacity and Normal. In the traditional CNN is used for both feature extraction and classification but in this ensemble method we are using the CNN for feature extraction and Random Forest Classifier for classification. Using these in this order we have successfully increased the accuracy of model by 23.1281% and decreased the loss while training. The model created with high accuracy can be used for further detection and classification of Pneumonia, Covid, Lung Opacity via X ray images of that patient which can be easily identified by the classifier.

**KEYWORDS:** CNN, Random Forest Classifier, Accuracy, Loss, Training, Validation, Pooling, Dense, Alexnet

## I. INTRODUCTION

Pneumonia and Lung Opacity are here with us for several years. Pneumonia is an infection in one or both lungs. Bacteria, viruses, and fungi cause it. The infection causes inflammation in the air sacs in your lungs, which are called alveoli. The alveoli fill with fluid or pus, making it difficult to breathe. In Lung Opacity Pulmonary opacification represents the result of a decrease in the ratio of gas to soft tissue (blood, lung parenchyma and stroma) in the lung. When reviewing an area of increased attenuation (opacification) on a chest radiograph or CT it is vital to determine where the opacification is. The patterns can broadly be divided into airspace opacification, lines and dots.

In last year we struck by covid which come with a bang from out of nowhere and here we are still dealing with it the, symptoms of the covid are similar to some extent to the symptoms Pneumonia. All three diseases can be found and recognized through the chest X ray images or with radiology in year 2020 when covid and pneumonia patients both were getting identified with chest x rays then lots of patient of pneumonia are misunderstood with covid patients then after certain amount of study finally we are able to classify between them now the covid still coming across our way and still Radiology (X ray images) is one of the way to identify patient situation we have built a model which can classify between the images of Covid, Pneumonia, Lung opacity and Normal Healthy lungs by analyzing the chest X ray image. We created this model using the ensemble method of CNN and Random Forest Classifier which increase the accuracy of the Model by measurable amount and also increase the testing and validation accuracy after in this method we have created two models one with CNN and with RFC and CNN.

[11] This novel algorithm was implemented by Dhvani Contractor at UNC Charlotte, USA for Fruit image recognition process.

### A. Convolutional

When programming a CNN, the input is a tensor with shape (number of images) x (image width) x (image height) x (image depth). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with



shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).
- The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

#### B. Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Average pooling uses the average value from each of a cluster of neurons at the prior layer.

#### C. Fully Connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

#### D. Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights. The vector of weights and the bias are called filters and represent particular features of the input (e.g., a particular shape).

#### E. Activation Functions

Activation layers apply a non-linear operation to the output of the other layers such as convolutional layers or dense layers.

- ReLu Activation: ReLu or Rectified Linear Unit computes the function  $f(x)=\max(0,x)$  to threshold the activation at 0.
- SoftMax Activation: SoftMax function is applied to the output layer to convert the scores into probabilities that sum to 1.

#### F. Optimizer

- Adam (Adaptive moment estimation): is an update to RMSProp optimizer in which the running average of both the gradients and their magnitude is used. In practice Adam is currently recommended as the default algorithm to use, and often works slightly better than RMSProp. In my experiments Adam also shows general high accuracy while Adadelta learns too fast. I've used Adam in all the experiments because I felt having similar optimizer would be a better baseline for comparing the experiments.

#### G. Rectified Linear Unit Activation Function(ReLU)

A type of activation function that transforms the value results of a neuron. The transformation imposed by ReLU on values from a neuron is represented by the formula  $y=\max(0,x)$ . The ReLU activation function clamps down any negative values from the neuron to 0, and positive values remain unchanged. The result of this mathematical transformation is utilized as the output of the current layer and used as input to a consecutive layer within a neural network.

#### H. Softmax Activation Function

A type of activation function that is utilized to derive the probability distribution of a set of numbers within an input vector. The output of a softmax activation function is a vector in which its set of values represents the probability of an occurrence of a class or event. The values within the vector all add up to 1.



I. Dropout

Dropout technique works by randomly reducing the number of interconnecting neurons within a neural network. At every training step, each neuron has a chance of being left out, or rather, dropped out of the collated contributions from connected neurons.

II. METHODOLOGY

A. CNN

A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analysing visual imagery. CNNs are regularized versions of multilayer perceptron (fully connected networks). The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The final convolution, in turn, often involves backpropagation in order to more accurately weight the end product.

In this method we are using CNN for both the feature extraction and Classification of images which are the images of chest X ray in our data set. Which is the custom method of performing classification using CNN. CNN is supposed to be the best algorithm in feature extraction but there are other algorithm which are better than the CNN for classification in next method we have used those methods.

Network Architecture

[12,13]We have used the Alexnet for the network layers . AlexNet CNN is probably one of the simplest methods to approach understanding deep learning concepts and techniques. AlexNet is not a complicated architecture when it is compared with some state of the art CNN architectures that have emerged in the more recent years. AlexNet is simple enough for beginners and intermediate deep learning practitioners to pick up some good practices on model implementation techniques.

AlexNet was the first convolutional network which used GPU to boost performance. AlexNet architecture consists of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. Each convolutional layer consists of convolutional filters and a nonlinear activation function ReLU. The pooling layers are used to perform max pooling. Input size is fixed due to the presence of fully connected layers. The input size is mentioned at most of the places as 224x224x3 but due to some padding which happens it works out to be 227x227x3 but in our case we are using 100x100x3. AlexNet overall has 60 million parameters.

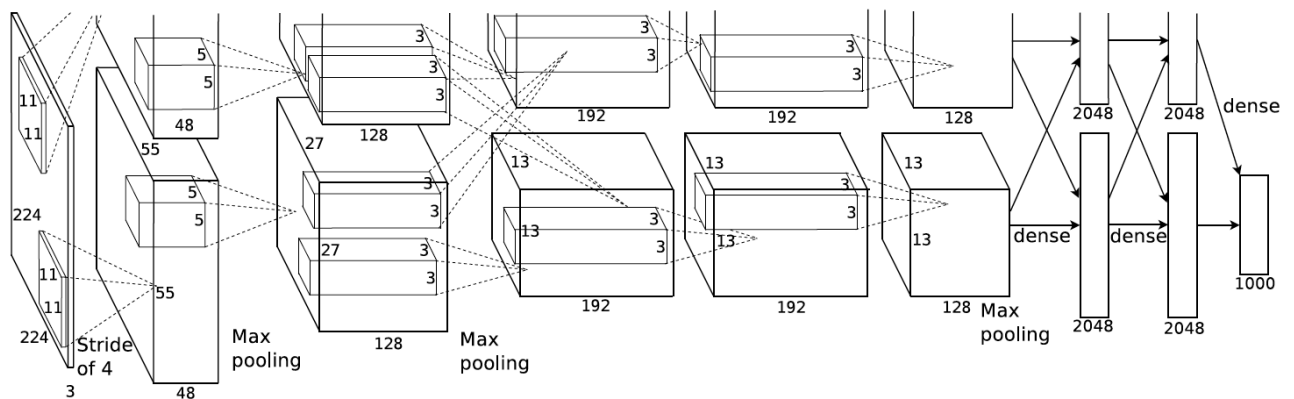


Fig1:-Alexnet Structure



*B. CCN with Random Forest Classifier*

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.

Visualization of a Random Forest Model making a prediction. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

There needs to be some actual signal in our features so that models built using those features do better than random guessing.

The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other. As we have seen in the above methodology of normal CNN. For both extraction of feature and Classification CNN is used here we are going to use the two different algorithms for two different purposes. We are going to use the CNN for the feature extraction and Random Forest Classifier for the classification of the image. This algorithm gives us measurable increase in accuracy.

*Network Architecture*

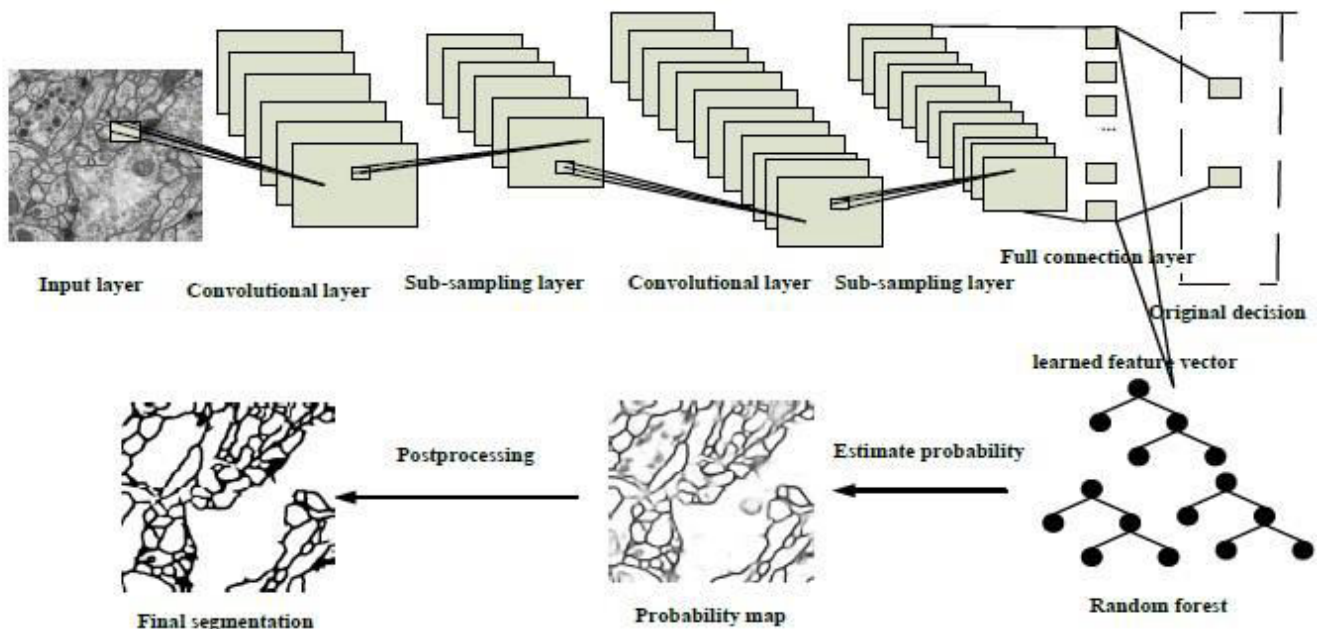


Fig2:-Ensemble Of CNN and Random Forest Classifier

*C. Metric*

Here, we are using the metric for this project is categorical cross entropy

$$logloss = \frac{1}{N} \sum_i^N \sum_j^M y_{ij} \log(p_{ij})$$



Here, each image has been labelled with one true class and for each image a set of predicted probabilities should be submitted.  $N$  is the number of images in the test set,  $M$  is the number of image class labels,  $\log$  is the natural logarithm,  $y_{ij}$  is 1 if observation  $i$  belongs to class  $j$  and 0 otherwise, and  $p_{ij}$  is the predicted probability that observation  $i$  belongs to class  $j$ .

A perfect classifier will have the log-loss of 0.

Multiclass log-loss punishes the classifiers which are confident about an incorrect prediction. In the above equation, if the class label is 1 (the instance is from that class) and the predicted probability is near to 1 (classifier predictions are correct), then the loss is really low as

$\log(x) \rightarrow 0$  as  $x \rightarrow 1$ , so this instance contributes a small amount of loss to the total loss and if this occurs for every single instance (the classifier is accurate) then the total loss will also approach 0.

On the other hand, if the class label is 1 (the instance is from that class) and the predicted probability is close to 0 (the classifier is confident in its mistake), as  $\log(0)$  is undefined it approaches  $-\infty$  so theoretically the loss can approach infinity. In order to avoid the extremes of the log function, predicted probabilities are replaced with  $\max(\min(p, 1 - 10^{-15}), 10^{-15})$ .

Graphically [1], assuming the  $i^{\text{th}}$  instance belongs to class  $j$  and  $y_{ij} = 1$ , it's shown that when the predicted probability approaches 0, loss can be very large.

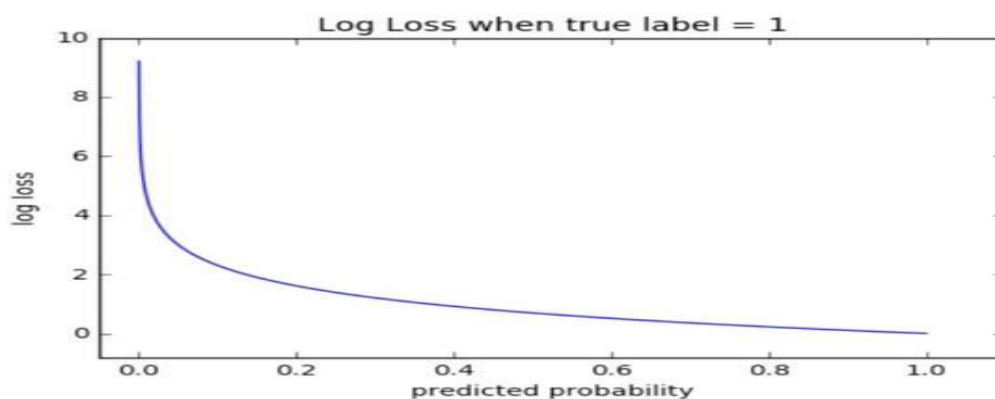


Fig3:-Cross\_Entropy Loss Representation

### III. IMPLEMENTATION

#### A. Dataset

##### COVID-19 CHEST X-RAY DATABASE

[1] A team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors have created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. This COVID-19, normal and other lung infection dataset is released in stages. In the first release they have released 219 COVID-19, 1341 normal and 1345 viral pneumonia chest X-ray (CXR) images. In the first update, they have increased the COVID-19 class to 1200 CXR images. In the 2nd update, they have increased the database to 3616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection) and 1345 Viral Pneumonia images.

\*\*COVID-19 data:

COVID data are collected from different publicly accessible dataset, online sources and published papers.

-2473 CXR images are collected from padchest dataset[2].

-183 CXR images from a Germany medical school[3].

-559 CXR image from SIRM, Github, Kaggle & Twitter[4,5,6,7]

-400 CXR images from another Github source[8].

\*\*\*Normal images:



10192 Normal data are collected from from three different dataset.

-8851 RSNA [8]-1341 Kaggle [10]

\*\*\*Lung opacity images:

6012 Lung opacity CXR images are collected from Radiological Society of North America (RSNA) CXR dataset [9]

\*\*\*Viral Pneumonia images:

1345 Viral Pneumonia data are collected from the Chest X-Ray Images (pneumonia) database [10]

**B. CNN**

While training the custom CNN we use Image Data Generator to ease Data preparation as it lables images based on Folder Name which is ideal for the way Data Set is arranged which are COVID, PNEUMONIA, LUNG OPACITY and NORMAL. In the process of Image Data Generation we provided directory of training and validation data set, in the image data generation class\_mode is set as "binary", batch\_size is equal to 100 and the image shape is defined as 100x100.

Before training the features of images were extracted using CNN and saved as array which then used for the training and testing purpose for building Model. We used alexnet as our network for training purpose.

Found 21165 images belonging to 4 classes.

Found 6533 images belonging to 4 classes.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 23, 23, 96)	34944
batch_normalization (BatchNo)	(None, 23, 23, 96)	384
max_pooling2d (MaxPooling2D)	(None, 11, 11, 96)	0
conv2d_1 (Conv2D)	(None, 11, 11, 256)	614656
batch_normalization_1 (BatchNo)	(None, 11, 11, 256)	1024
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 256)	0
conv2d_2 (Conv2D)	(None, 5, 5, 384)	885120
batch_normalization_2 (BatchNo)	(None, 5, 5, 384)	1536
conv2d_3 (Conv2D)	(None, 5, 5, 384)	1327488
batch_normalization_3 (BatchNo)	(None, 5, 5, 384)	1536
conv2d_4 (Conv2D)	(None, 5, 5, 256)	884992
batch_normalization_4 (BatchNo)	(None, 5, 5, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 256)	0
flatten (Flatten)	(None, 1024)	0
0dense (Dense)	(None, 4096)	4198400
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4)	16388

Total parameters: 24,748,804

Trainable parameter: 24,746,052

Non-trainable parameter: 2,752

**Layers:**

- Convolution: Convolutional layers convolve around the image to detect edges, lines, blobs of colors and other visual elements. Convolutional layers hyper parameters are the number of filters, filter size, stride, padding and activation functions for introducing non-linearity.
- MaxPooling: Pooling layers reduces the dimensionality of the images by removing some of the pixels from the image. MaxPooling replaces a n x n area of an image with the maximum pixel value from that area to down sample the image.
- Flatten: Flattens the output of the convolution layers to feed into the Dense layers.
- Dense: Dense layers are the traditional fully connected networks that maps the scores of the Convolutional.
- Batch Normalisation layer: Batch Normalization is a technique that mitigates the effect of unstable gradients within a neural network through the introduction of an additional layer that performs operations on the inputs from the previous



layer. The operations standardize and normalize the input values, after that the input values are transformed through scaling and shifting operations.

- Dropout: Dropout technique works by randomly reducing the number of interconnecting neurons within a neural network. At every training step, each neuron has a chance of being left out, or rather, dropped out of the collated contributions from connected neurons.

The shape Found while working with CNN is (21165, 100, 100, 3) (21165,) (6533, 100, 100, 3) (6533,) 4.

Which is then normalized for the further training using label binarizer and then in one hot encoding.

The Model is trained over 21165 images belonging to 4classes COVID, PNEUMONIA, LUNG OPACITY and NORMAL respectively. And Validate over 6533 images belonging to 4 classes. Total number of parameters are 24,748,804.Adam Optimizer is used in the training of the model as metric we use categorical\_crossentropy. The model is trained over 2 epoch. We retrieve a list of results on training and test data sets for each training epoch which is used to plot training and validation accuracy per epoch.

After 2 Epoch model is saved as CNNmodel.h5.

### C. CNN with Random Forest Classifier

While implementing this method we are using the CNN for image feature extraction for that we take the first fully connected dense as the features which will be of dimension (1 x 4096).After that we will find 4096 features for every image that means for every image we get 4096 features and for N images we will get Nx4096 features now we will save this array of features as features.npy. Then we have created data frame with features and column name using pandas and numpy. And now will feed the those features to Random Forest Classifier for the training purpose.

The shapes of input and features are as follows:

Training Features Shape: (21165, 4096)

Training Labels Shape: (21165,)

Test Features Shape: (6533, 4096)

Test Labels Shape: (6533,)

Then After training we did testing for the accuracy the process for testing is also same. We used the first fully connected layer of. Alexnet then Extracted features of the image which is 4096 and then created a npy file for those features created a data frame with features and column name using pandas and numpy now we feed those features to the Random Forest Classifier. Then after that we calculated the accuracy of results.





#### IV. RESULTS AND DISCUSSION

##### A. CNN

After completion of 2 epochs Accuracy and loss recorded for Training and Validation are as follows.

Plots

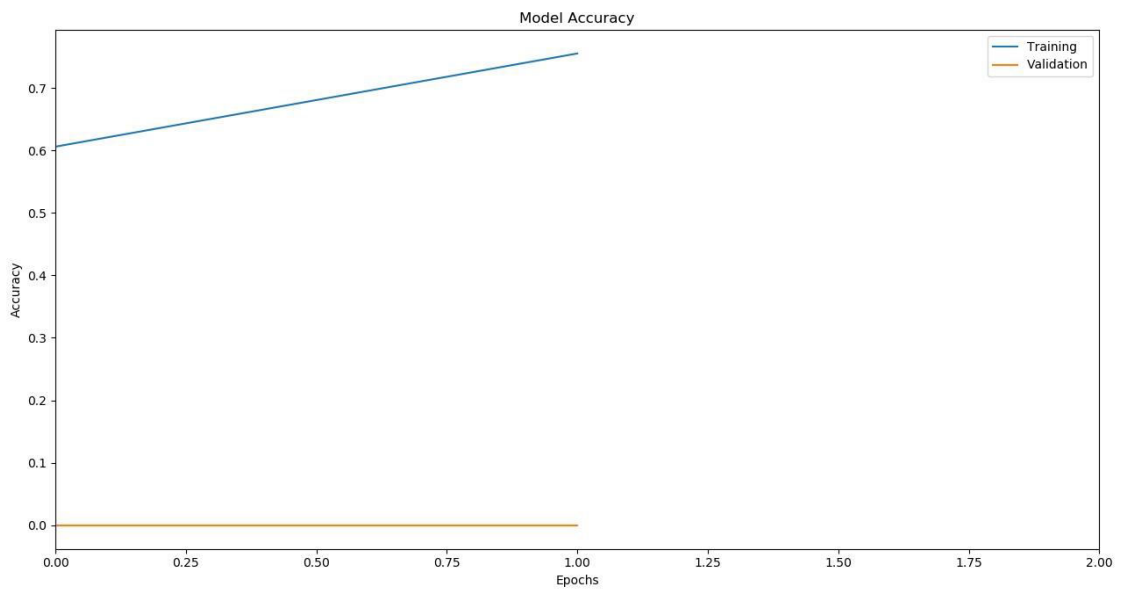


Fig4:- Training And Validation Accuracy

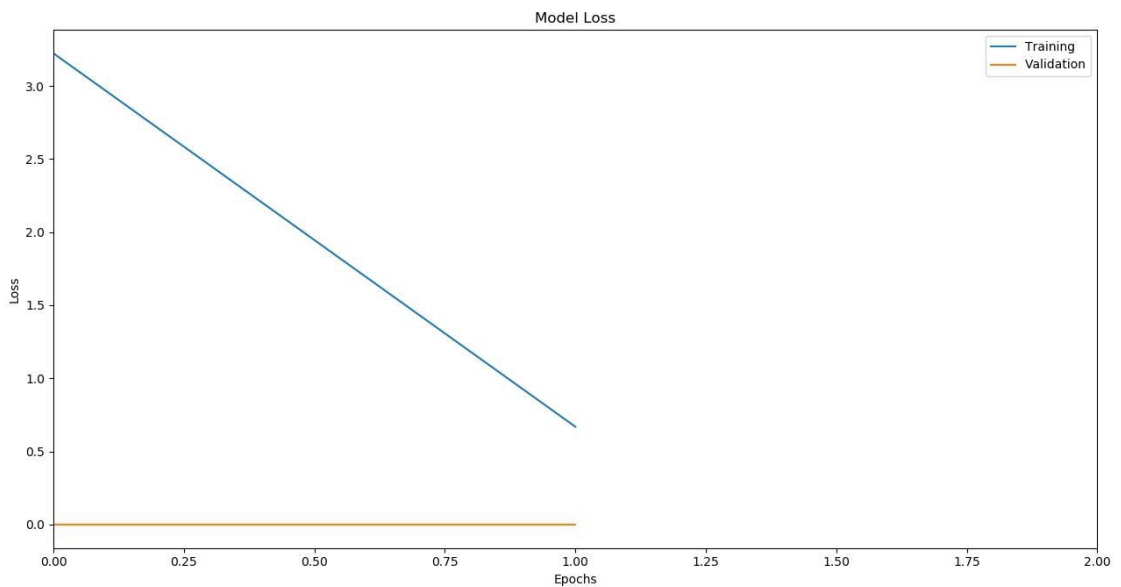


Fig5:- Training And Validation Loss



Training:

Accuracy=0.7678

Loss=0.6529

Validation:

Accuracy=0.0000e+00

Loss=0.0000e+00

Test loss: 8.284736633300781

Test accuracy: 0.2603704333305359

B. CNN with Random Forest Classifier

After Completion of this method we have found the Accuracy of the Random Forest Classifier which is as given below

Accuracy: 99.90815857951937 %.( To get Accuracy in % we have multiplied it with 100)

## V. CONCLUSION AND FUTURE SCOPE

### A. Conclusion

After implementation of both CNN and CNN with Random Forest Classifier on the dataset of COVID-19 CHEST X-RAY DATABASE and recording the accuracy and loss of both training and validation on CNN method and after that seeing the accuracy of the Random Forest Classifier we concluded that we have successfully increased the accuracy of CNN by 23.1281% and we have successfully used the feature extraction power of the CNN and Classification power of the Random forest Classifier .

### B. Future Scope

The model created using Random Forest Classifier and CNN can be used in future development of application for detection and classification patients of Covid, Pneumonia , Lung Opacity by using the image input the chest X ray of the patient which can be powerful tool for the medical use.

## REFERENCES

- [1] -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.
- [2] -Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughair, S.M., Khan, M.S. and Chowdhury, M.E., 2020. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images. arXiv preprint arXiv:2012.02238.
- [3] <https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711>
- [4] <https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png>
- [5] <https://sirm.org/category/senza-categoria/covid-19/>
- [6] <https://eurorad.org>
- [7] <https://github.com/ieee8023/covid-chestxray-dataset>
- [8] [https://figshare.com/articles/COVID-19\\_Chest\\_X-Ray\\_Image\\_Repository/12580328](https://figshare.com/articles/COVID-19_Chest_X-Ray_Image_Repository/12580328)
- [9] <https://github.com/armiro/COVID-CXNet>
- [10] <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>
- [11] <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- [12] <https://github.com/dhwanirc/Image-Classification-with-CNN-RF>
- [13] "The data that transformed AI research—and possibly the world".
- [14] Jump up to: a b c d e Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" (PDF). Communications of the ACM. 60 (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782. S2CID 195908774.
- [15] [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
- [16] Covid-19 detection in chest X-ray through random forest classifier using a hybridization of deep CNN and DWT optimized features Rafid Mostafiz a,b , Mohammad Shorif Uddin c,† , Nur-A- Alam a,b , Md. Mahfuz Reza a , Mohammad Motiur Rahman a
- [17] Automatic COVID-19 Detection from X-Ray images using Ensemble Learning with Convolutional Neural Network AMIT KUMAR DAS1,2 , SAYANTANI GHOSH1 , SAMIRUDDIN THUNDER 2 , ROHIT DUTTA 2 , SACHIN AGARWAL 2 and AMLAN CHAKRABARTI 1 1 A. K. Choudhury School of Information Technology,



University of Calcutta, Kolkata, India 2 Institute of Engineering & Management, Kolkata, India email: amitkrdas.kol@gmail.com; sayantani31g@gmail.com; samird2000@gmail.com; rohitdutta2510@gmail.com; agarwalsachin518@gmail.com; [acakcs@caluniv.ac.in](mailto:acakcs@caluniv.ac.in)

- [17] Insect Classification using Custom CNN Vs Transfer Learning Naumanurrahman Shaikh Mujiburrahman<sup>1</sup>, Manish Sanjay Zalte<sup>2</sup>, Dr. Manoj E. Patil<sup>3</sup> UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India<sup>1</sup> UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India<sup>2</sup> Assoc. Prof., Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India<sup>3</sup>



**INNO**  **SPACE**  
SJIF Scientific Journal Impact Factor

**Impact Factor:  
7.512**

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
**INDIA**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details