Convid-Net: An Enhanced Convolutional Neural Network Framework for COVID-19 Detection from X-Ray Images

Chapter · January 2021

DOI: 10.1007/978-981-33-4673-4_55

3 authors:

Sabbir Ahmed
Jahangirnagar University

Md. Farhad Hossain
Jahangirnagar University

Manan Noor
Jahangirnagar University

Convid-Net: An enhanced Convolutional Neural Network framework for COVID-19 detection from X-ray images

Sabbir Ahmed, Md. Farhad Hossain, and Manan Binth Taj Noor

Institute of Information Technology, Jahangirnagar University, Dhaka, Bangladesh
Emails: {sabbir.iit.ju,farhadiitju}@gmail
Email: manan.noor@juniv.edu

Abstract. This article aims to demonstrate a deep convolutional neural network (CNN) framework namely *Convid-Net* based on a combination of residual network and parallel convolution ($CONV$) to detect COVID-19 from chest X-ray images. The proposed architecture can choose optimum features from different parallel $CONV$ and residual connection increasing overall accuracy with less computational expenses. A custom dataset has been created for this work which consists of total 1440 images of COVID-19, 2470 normal images and 2407 chest X-ray images of viral and bacterial pneumonia; collected from different publicly available sources. Augmentation and preprocessing have been applied as well to increase the number of data for better training purposes. *Convid-Net* has been trained and tested on a prepared augmented dataset which achieved accuracy of 97.99%. The promising result of the proposed system shows that it converges to an overall higher accuracy and can be a very useful method for physicians and radiologists to assist them in rapid detection and diagnosis of COVID-19 from radiography images. These results also indicate that *Convid-Net* architecture can further be used in other image based classification tasks.

Keywords: Convid-Net, Coronavirus, Covid-19, Deep Learning, X-ray images, CNN, LSTM, Residual Network.

1 Introduction

Coronavirus disease 2019 (COVID-19) is characterized as a disease caused by a novel coronavirus now known as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2; formerly known as 2019-nCoV), first identified as a result of an outbreak of respiratory disease in Wuhan City, Hubei, China. Globally, as of 26 August 2020, there have been 23,752,965 confirmed cases of COVID-19, including 815,038 deaths, reported to the World Health Organization (WHO) [5]. Currently, Reverse Transcription Polymerase Chain Reaction (RT-PCR) and antibody testing are prominent methods of testing people for COVID-19 across global healthcare systems. In medical science, different test-based techniques are used for diagnosis, which require expert opinion. From that perspective, exploring data strategies for medical image is a task in terms of obtaining insight, then assessing and diagnosing a particular disease. Following that, in order to track the diagnosis of SARS-CoV-2 infections, there are some medical imaging methods utilizing radiological images; such as chest X-rays or Computed Tomography (CT). Former experiments [21] indicate that COVID-19 in the context of ground-glass opacities causes abnormalities that are noticeable in the chest X-rays and CT images. Classification of medical images is a tedious task for handcrafted features, thus recent trends are focused on deep learning approaches for higher recognition rate. A survey summarizes over 300 contributions in deep learning for medical image classification, object detection, segmentation, registration, and other tasks discussed in [14]. Deep learning-based image classification can also be conducted on X-ray and CT images to detect COVID-19 patterns. However, datasets of X-ray images are more available than CT images and less time-consuming in terms of processing the image [13].

In this article, multiple parallel $CONV$ with different $CONV$ filters and residual layer-based architecture has been proposed to classify X-ray images into COVID-19, Pneumonia and Normal classes. Dataset has been split into a testing set and training set before augmentation for correctly measuring classifier performance. The initial dataset is created by combining several other publicly available datasets which contain different number of COVID-19, normal and pneumonia chest X-ray images. Since, the initial dataset generated for this work is imbalanced, the same number of data have been generated for each class with augmentation for this work. Parallel $CONV$ layer has been used for feature extraction and data complexity reduction. Residual layer with varying parallel $CONV$ has been applied to deal with vanishing gradient problems. Finally the trained networks later can be used for COVID-19 detection from X-ray images in general, thus larger dataset can be collected by crowdsourcing. Flow diagram of the proposed approach is shown in figure 1.

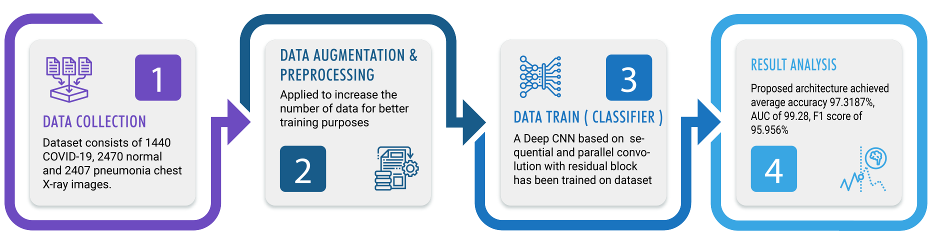

Fig. 1: Flow diagram of the proposed system

2 Related Studies

This section discusses the newly developed COVID-19 detection systems based on deep learning techniques. Islam et al. proposed a CNN for deep feature extraction with Long Short-Term Memory ($LSTM$) for detection model [11]. A patch-based CNN approach has been developed for COVID-19 diagnosis with clinically interpretable saliency maps with limited data by [16]. However, their datasets only contain 180 chest X-ray (CXR) of COVID-19 patients. In another work Wang et al. introduced COVID-Net to identify distinctive anomalies in CXR images [20]. Overall this method has been hampered by training on imbalance data, lack of rigorous preprocessing of data, and inadequate simulation of decisions. Transfer learning-based CNN has also been implied in [15] for the diagnosis of infected, coronavirus and pneumonia patients using chest X-ray radiographs with 5-fold cross validation. Then, Resnet-18 with Decompose, Transfer and Compose has been developed in [6] for detection of COVID-19 from chest X-ray images. But their validation accuracy severely exceeded train accuracy which is a result of overfitting. Bayesian Deep Learning classifier was trained on COVID-19 X-Ray images utilizing transfer learning method for estimating model uncertainty by Ghoshal et al [10]. A Generative Adversarial Networks for a pneumonia chest x-ray dataset creation and AlexNet, GoogLeNet, Squeeznet, ResNet-18 for detection is presented by M. khalifa et al [12]. Ozturk et al. on the other hand used handcrafted feature extraction, SMOTE algorithm-based data processing and stacked auto-encoder for classifying imbalance COVID-19 chest X-ray [17]. Four X-ray based detection methods [19], 48 Diagnostic imaging [21], 69 Models [8] based on X-ray images and CT have been studied from the preprocessing perspective, extraction of the characteristics and classification. In contrast to the studies above, the proposed methodology enforces a customized larger dataset incorporating a newer and effective version of CNN; $Convid\text{-}Net$ which tends to probe optimum solution for covid-19 detection.

3 Methodology

3.1 Dataset

Table 1: Public datasets used to create the customized dataset for this work

Dataset	Covid-19	Normal	Pneumonia
COVID-19 Xray [4]	94	94	0
COVID-19 Patients Lungs X Ray Images 10000 [2]	70	20	0
COVID-19 Radiography Database [3]	219	1341	1345
ieee8023/covid-chestxray-dataset [9]	180	0	0
COVID-19 Detection X-Ray Dataset [1]	60	880	1072
Actualmed COVID-19 Chest X-ray Dataset [7]	58	127	0
BIMCV covid19+ [18]	805	0	0

Table 1 contains the information of those publicly available datasets which have been used to create the customized dataset for the proposed architecture. Collected data set contains total 1440 chest X-ray images of COVID-19, 2470 normal chest X-ray images and 2407 chest X-ray images of pneumonia. Since this amount of data is insufficient for neural network training, augmentation has been applied in this context to generate a larger dataset. Online augmentation with neural networks takes more time than offline augmentation. If the train and test data are augmented from the same source, then this undermines the performance of neural network. For this reason, each of the three classes: COVID-19, Normal, pneumonia has been split into testing set and training set. 30% of the total data of each class is reserved for testing and 70% for training purposes. Furthermore, as the combined dataset contains images from different sources; randomly selecting 30% images can result in erroneous accuracy. Because, testing sets may contain completely different images than training images, or testing sets may be the same as training sets. To overcome this problem, images of each class have been ordered. Then every third image is selected sequentially for the test set; rest are selected for the train set. Thus both test and train set contain images from every collected dataset which in turns creates a balanced dataset.

3.2 Data preprocessing and Augmentation

While classifying using deep learning algorithms, the mismatch ratio of dataset class distribution has an immense effect on model output after splitting data into test and train directories; thus image augmentation has been applied to create a larger dataset. In this case, the number of COVID-19 images are significantly lower than in other classes. To solve this issue, augmentation has been applied to generate same number of images for each class. Again, the dimension of images varies in the collected dataset. Thus, after resizing the images at $256 * 256$ pixels; the following augmentation techniques have been applied:

- **Feature wise center** Normalizing input by subtracting mean value of the input distribution.
- **Rotation** Rotates image in certain direction (clockwise and counterclockwise) at most 20 degrees.
- **Zoom** Scaling up images at most 10%.
- **Width shift** Moving image to a certain direction along the horizontal axis at most 15%.
- **Shear range** Displaces each point in the vertical direction by a sum commensurate with its distance from the picture edge, at most 10%.
- **Horizontal Flip** Making mirror of input image by flipping horizontal axis.
- **Fill mode nearest** Points outside the boundaries of the input are filled according to nearest color.

Table 2 shows the train and test data distribution of augmented and non-augmented data used for the dataset, created for this work.

Table 2: Dataset details

	Non Augmented			Augmented		
	Train	Test	Total	Train	Test	Total
COVID-19	1008	432	1440	7052	3022	10074
Normal	1729	741	2470	7092	3004	10096
Pneumonia	1685	722	2407	6925	2928	9903
Total Augmented data						30,073

3.3 Convid-Net

Proposed deep learning architecture *Convid-Net* consists of the following building blocks:

($CONV$) layer Set of learnable filter (kernel) which calculates the output of neurons attached to local input areas, each calculating a dot product between their weights and a specific area related to them in the input volume.

Sub sampling (pooling) layer Pooling layer takes maximum(Max Pooling (MP)) or average (Average Pooling) value from each $CONV$ feature matrix input.

Batch normalization (BN) layer A process used to render neural networks quickly in a robust way by re-centering and re-scaling of the input layer.

Dropout (DO) layer A regularization method to temporarily remove units, including all its incoming and outgoing connections. This regularization prevents overfitting of the network.

Add layer Add same dimension input tensor by allowing skipping connection (residual) from previous layers and also parallel $CONV$, which in turns creates an alternative route for gradients to pass.

Dense layer Completely Connected layers in a neural network where all inputs from one layer are linked to each activation unit of the previous and following layer. Softmax is generally used as an activation function in the dense layer.

Figure 2 shows the overall *Convid-Net* architecture.

Here, deep CNN architecture has been proposed with various degrees of reduction based valid padding parallel $CONV$ and reduction less same padding parallel $CONV$ with residual block. This network subsequently expands into different $CONV$, pooling, BN layers and then fuses those layers for different feature extraction. Valid padding $CONV$ reduces image dimension (height and width) while extracting features (Block 0-4 in figure 2). Subsequent $CONV$ layers have been used in sub-blocks for complex feature extraction. Since, the valid padding $CONV$ layers reduce input dimension at a constant rate, it constrains the number of convolutions applied to a specific image input. The same padding $CONV$ with a residual connection from the previous layer has also been applied so that input images get fully covered by filters and output preserves the dimension of the input image(Block 5-10 in figure 2). Subsequent $CONV$ layer(Block 0,2 in figure 2) has been used in this architecture because larger kernel size

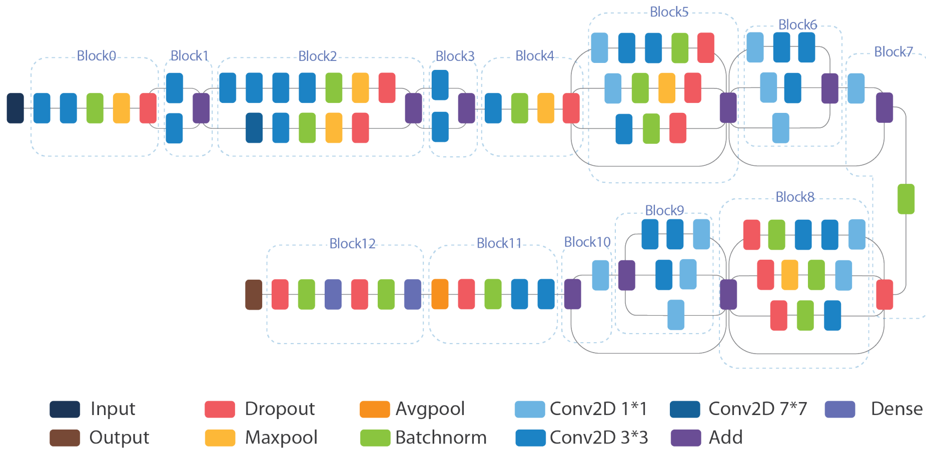

Fig. 2: *Convid-Net* Architecture

CONV is computationally expensive. Instead of using subsequent smaller kernel size *CONV* also results in the same feature extraction and output dimension with less computation. While creating the subsequent layer, learning of the previous layer may disappear. Parallel *CONV* allows a network to choose relevant filter size *CONV*. To reduce overfitting *BN* and *DO* have been added either in each parallel *CONV* or at the end of the concatenation of parallel layers. Parallel *CONV* have been used with a residual block (Block 5,7,8,10 of figure 2) to prevent vanishing gradient. Thus the network can select relevant features from different *CONV* filters or input from the previous block. To minimize the number of trainable parameters and computational complexity; reduction based blocks have been used. Block 0-4 reduces the image dimension from $120 * 120$ to $10 * 10$ pixels. Then more complex parallel *CONV* (block 5-10) were applied for feature extraction. To further reduce dimension, for being able to feed into dense layer, sequential *CONV-MP* blocks were placed after block 10. Following that, since flatten turns a multidimensional vector into a one dimensional vector without any calculation; thus global average pooling has been used in this case for additional special dimensions and overfitting depletion. A Summary of the proposed architecture is shown in figure 3. To compile the given architecture, *Adam* has been used as optimizer and categorical cross entropy as loss function with batch size 64. Total parameters calculated as 4,292,067 for the proposed architecture.

4 Experimental Analysis

The experiments are performed in anaconda environment using python language and TensorFlow 2.1 running on a machine equipped with intel Core i5 processor, 8 GB ram, 2GB Nvidia 920mx GPU. All models trained on a total 21,096 and tested on a total 8,954 chest X-ray images. $120 * 120$ dimensional images have

	Layer(type)	Output Shape	Param	Connected to
	Image (Input Layer)	(120, 120, 3)	0	
Block_0	Conv2D_0	(118, 118, 32)	896	Image
	Conv2D_1	(116, 116, 32)	9248	Conv_0
	BatchNorm_0	(116, 116, 32)	128	Conv_1
	MaxPooling_0	(58, 58, 32)	0	BatchNorm_0
	DropOut_0	(58, 58, 32)	0	MaxPooling_0
Block_1	Conv2D_2	(56, 56, 32)	9248	DropOut_0
	Conv2D_3	(56, 56, 32)	9248	DropOut_0
	Add_0	(56, 56, 32)	0	Conv2D_2
				Conv2D_3
Block_2	Conv2D_6	(54, 54, 64)	18496	Add_0
	Conv2D_7	(52, 52, 64)	36924	Conv2D_6
	Conv2D_4	(50, 50, 64)	100416	Add_0
	Conv2D_8	(50, 50, 64)	36928	Conv2D_7
	Conv2D_5	(48, 48, 64)	36928	Conv2D_4
	Conv2D_9	(48, 48, 64)	36928	Conv2D_8
	BatchNorm_1	(48, 48, 64)	256	Conv2D_5
	BatchNorm_2	(48, 48, 64)	256	Conv2D_9
	MaxPooling_1	(24, 24, 64)	0	BatchNorm_1
	MaxPooling_2	(24, 24, 64)	0	BatchNorm_2
	DropOut_1	(24, 24, 64)	0	MaxPooling_1
	DropOut_2	(24, 24, 64)	0	MaxPooling_2
	Add_1	(24, 24, 64)	0	DropOut_1
				DropOut_2
Block_3	Conv2D_10	(22, 22, 64)	36928	Add_1
	Conv2D_11	(22, 22, 64)	36928	Add_1
	Add_2	(22, 22, 64)	0	Conv2D_10
				Conv2D_11
Block_4	Conv2D_12	(20, 20, 128)	73856	Add_2
	BatchNorm_3	(20, 20, 128)	512	Conv2D_12
	MaxPooling_3	(10, 10, 128)	0	BatchNorm_3
	DropOut_3	(10, 10, 128)	0	MaxPooling_3
Block_5	Conv2D_15	(10, 10, 128)	16512	DropOut_3
	Conv2D_13	(10, 10, 128)	16512	DropOut_3
	Conv2D_16	(10, 10, 128)	147584	Conv2D_15
	BatchNorm_4	(10, 10, 128)	512	Conv2D_13
	Conv2D_14	(10, 10, 128)	147584	DropOut_3
	Conv2D_17	(10, 10, 128)	147584	Conv2D_16
	MaxPooling_4	(10, 10, 128)	0	BatchNorm_4
	BatchNorm_5	(10, 10, 128)	512	Conv2D_14
	BatchNorm_6	(10, 10, 128)	512	Conv2D_17
	DropOut_4	(10, 10, 128)	0	MaxPooling_4
	DropOut_5	(10, 10, 128)	0	BatchNorm_5
	DropOut_6	(10, 10, 128)	0	BatchNorm_6
	Add_3	(10, 10, 128)	0	DropOut_3
				DropOut_4
				DropOut_5
				DropOut_6
Block_6	Conv2D_21	(10, 10, 128)	16512	Add_3
	Conv2D_19	(10, 10, 128)	16512	Add_3
	Conv2D_22	(10, 10, 128)	147584	Conv2D_21
Block_6	Conv2D_18	(10, 10, 128)	16512	Add_3
	Conv2D_20	(10, 10, 128)	147584	Conv2D_19
	Conv2D_23	(10, 10, 128)	147584	Conv2D_20
	Add_4	(10, 10, 128)	0	Conv2D_18
				Conv2D_20
				Conv2D_23
Block_7	Conv2D_24	(10, 10, 128)	16512	Add_4
	Add_5	(10, 10, 128)	0	Conv2D_24
				Add_3
	BatchNorm_7	(10, 10, 128)	512	Add_5
	DropOut_7	(10, 10, 128)	0	BatchNorm_7
Block_8	Conv2D_27	(10, 10, 128)	16512	DropOut_7
	Conv2D_25	(10, 10, 128)	16512	DropOut_7
	Conv2D_28	(10, 10, 128)	147584	Conv2D_27
	BatchNorm_8	(10, 10, 128)	512	Conv2D_25
	Conv2D_26	(10, 10, 128)	147584	DropOut_7
	Conv2D_29	(10, 10, 128)	147584	Conv2D_28
	MaxPooling_5	(10, 10, 128)	0	BatchNorm_8
	BatchNorm_9	(10, 10, 128)	512	Conv2D_26
	BatchNorm_10	(10, 10, 128)	512	Conv2D_29
	DropOut_8	(10, 10, 128)	0	MaxPooling_5
	DropOut_9	(10, 10, 128)	0	BatchNorm_9
	DropOut_10	(10, 10, 128)	0	BatchNorm_10
	Add_6	(10, 10, 128)	0	DropOut_7
				DropOut_8
				DropOut_9
				DropOut_10
Block_9	Conv2D_33	(10, 10, 128)	16512	Add_6
	Conv2D_31	(10, 10, 128)	16512	Add_6
	Conv2D_34	(10, 10, 128)	147584	Conv2D_33
	Conv2D_30	(10, 10, 128)	16512	Add_6
	Conv2D_32	(10, 10, 128)	147584	Conv2D_31
	Conv2D_35	(10, 10, 128)	147584	Conv2D_34
	Add_7	(10, 10, 128)	0	Conv2D_30
				Conv2D_32
				Conv2D_35
Block_10	Conv2D_36	(10, 10, 128)	16512	Add_7
	Add_8	(10, 10, 128)	0	Conv2D_36
				Add_6
Block_11	Conv2D_37	(8, 8, 256)	295168	Add_8
	Conv2D_38	(6, 6, 512)	1180160	Conv2D_37
	BatchNorm_11	(6, 6, 512)	2048	Conv2D_38
	DropOut_11	(6, 6, 512)	0	BatchNorm_11
	GlobalAvgPool	(512)	0	DropOut_11
Block_12	Dense_0	(512)	262656	GlobalAvgPool
	BatchNorm_12	(512)	2048	Dense_0
	DropOut_12	(512)	0	BatchNorm_12
	Dense_1	(256)	131328	DropOut_12
	BatchNorm_13	(256)	1024	Dense_1
	DropOut_13	(256)	0	BatchNorm_13
	Dense_2 (Output layer)	(3)	771	DropOut_13

Fig. 3: Summary of *Convid-Net* Architecture

been used as input in all networks. *DO* of 0.25 and 0.5 have been applied in the *CONV* and dense/ *LSTM* portion respectively. Learning rate of 0.001 has been utilized for each model. For comparison a less deep version of proposed architecture and a *CNN-LSTM* model have also been designed and evaluated on the same dataset. These model details are following:

Model 1 Model 1 consists of 8 blocks with less $CONV$ layer than the proposed architecture. Only two parallel $CONV$ residual blocks have been used for it. Total parameters used in this model are 2,716,38.

Model 2 A sequential $CONV$ layer with MP, BN and DO has been utilized in model 2. The feature extraction CNN is 18 layer deep. Flatten parameters fed into 512 units of $LSTM$ layer with DO. Subsequently, similar $LSTM$ and DO layer have been applied again before the dense layer. Two hidden dense layers with 256,256 neuron units have been applied before the output layer. Total parameters used in this model are 11,140,483.

Model 3 Model 3 comprises the proposed architecture. Model 3 has been created from model 1 by optimizing parameters and layers

Figure 4 and 5 shows the graphical representation of different performance metrics; evaluated for different models of this work. In order to evaluate and quantify the efficiency of the proposed architecture *Convid-Net* on the dataset; several evaluation metrics have been utilized. Accuracy, Recall, Precision, F1-Score, Sensitivity, Specificity, Area Under Curve (AUC) are calculated for each of the models to distinguish the performance measures among the models.

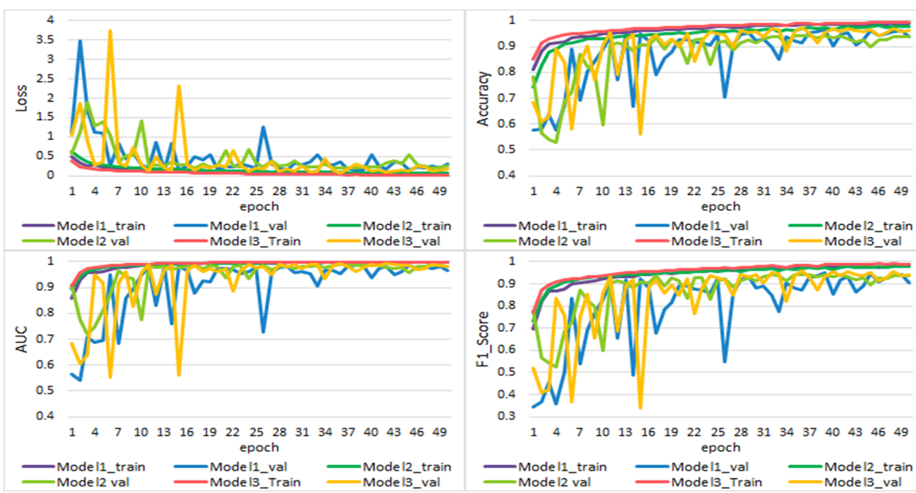

Fig. 4: AUC, Loss, F1-score and accuracy measurements of Model 1, 2 and 3 of train and test dataset

Different evaluation metrics show that Model 3 performs better than Model 1 and Model 2 owing to the presence of residual connections; nonetheless the number of parameters in Model 3 are nearly one-third of Model 2. Since test accuracy is lower than train accuracy and test accuracy closely follow train accuracy it can be stated that proposed model does not suffer from any overfitting or underfitting problem; which further validates dataset creation logic. Proposed architecture achieved average training accuracy of 97.3187% and 89.37% testing

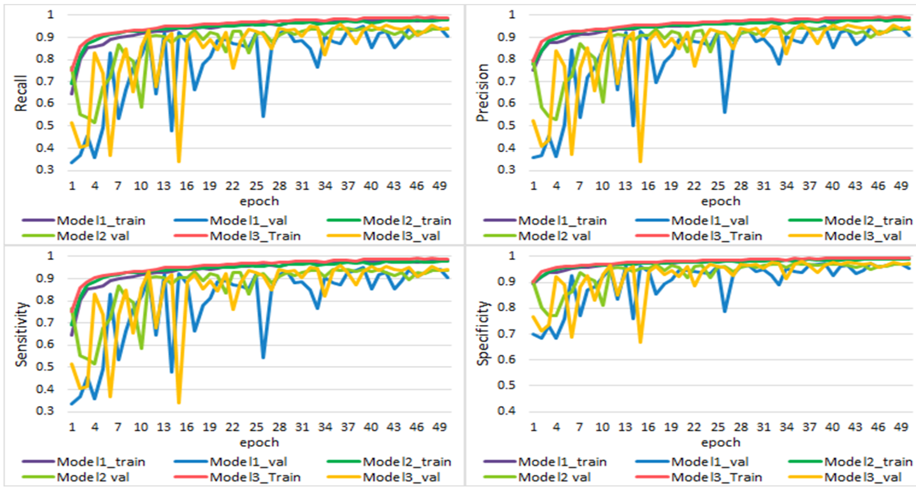

Fig. 5: Precision, Recall, Sensitivity and Specificity measurements of Model 1, 2 and 3 of train and test dataset

accuracy with maximum 99.47% training and 97.4% testing accuracy in only 50 epochs. Average training recall of 95.78%, precision of 96.13% and AUC of 99.28, F1 score of 95.956%, specificity of 98.08%, sensitivity of 95.78% has also been achieved by the proposed architecture.

5 Conclusion

As COVID-19 emerges all over the world; more tests are needed to detect and track the spread of COVID-19. PCR test kit shortage is one of the main obstacles for rapid testing. From that perspective, a deep CNN framework ($Convid$-Net) has been proposed with parallel $CONV$ and residual connection for the detection of COVID-19 from X-ray images for this work. Proposed model achieves great performance in the given augmented dataset. Since data have been collected from various sources and augmentation has been used to generate enough data to train the networks, shortage of COVID-19 chest X-ray data is a major limitation. Though $Convid$-Net has achieved admissible performance in classifying X-ray images, usage of the architecture in other imaging techniques such as CT scan, ultrasonic imaging is yet to be determined. In future, an openly available prediction application can be built for general use and to crowdsource COVID-19 X-ray images.

References

1. COVID-19 Detection X-Ray Dataset, https://kaggle.com/darshan1504/covid19-detection-xray-dataset

2. COVID-19 Patients Lungs X Ray Images 10000, https://kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images
3. COVID-19 Radiography Database, https://kaggle.com/tawsifurrahman/covid19-radiography-database
4. COVID-19 Xray Dataset (Train & Test Sets), https://kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets
5. WHO Coronavirus Disease (COVID-19) Dashboard, https://covid19.who.int
6. Abbas, A., Abdelsamea, M.M., Gaber, M.M.: Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. arXiv preprint arXiv:2003.13815 (2020)
7. agchung: agchung/Actualmed-COVID-chestxray-dataset (Sep 2020), https://github.com/agchung/Actualmed-COVID-chestxray-dataset, original-date: 2020-04-17T15:45:49Z
8. Chen, D., Ji, S., Liu1, F., Li, Z., Zhou, X.: A review of automated diagnosis of covid-19 based on scanning images (2020)
9. Cohen, J.P.: ieee8023/covid-chestxray-dataset (Sep 2020), https://github.com/ieee8023/covid-chestxray-dataset, original-date: 2020-02-14T23:22:23Z
10. Ghoshal, B., Tucker, A.: Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. arXiv preprint arXiv:2003.10769 (2020)
11. Islam, M.Z., Islam, M.M., Asraf, A.: A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. Informatics in Medicine Unlocked p. 100412 (2020)
12. Khalifa, N.E.M., Taha, M.H.N., Hassanien, A.E., Elghamrawy, S.: Detection of coronavirus (covid-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest x-ray dataset. arXiv preprint arXiv:2004.01184 (2020)
13. Khan, A.I., Shah, J.L., Bhat, M.M.: Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images. Computer Methods and Programs in Biomedicine p. 105581 (2020)
14. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis **42**, 60–88 (2017)
15. Narin, A., Kaya, C., Pamuk, Z.: Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. arXiv preprint arXiv:2003.10849 (2020)
16. Oh, Y., Park, S., Ye, J.C.: Deep learning covid-19 features on cxr using limited training data sets. IEEE Transactions on Medical Imaging (2020)
17. Ozturk, S., Ozkaya, U., Barstugan, M.: Classification of coronavirus images using shrunken features. medRxiv (2020)
18. Region, B.M.I.D.o.t.V., Pertusa, A., Vaya, M.d.l.I.: BIMCV-COVID19+ (May 2020). https://doi.org/10.17605/OSF.IO/NH7G8, https://osf.io/nh7g8/, publisher: OSF
19. Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., He, K., Shi, Y., Shen, D.: Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. IEEE reviews in biomedical engineering (2020)
20. Wang, L., Wong, A.: Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. arXiv preprint arXiv:2003.09871 (2020)
21. Wynants, et al.: Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal. bmj **369** (2020)