Experimental Investigation of CT Scan Imaging based COVID-19 Detection with Deep Learning Techniques

An Internship Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Aditya Vijay Shinde (MIS- 112015003)

Under the Supervision of: Dr. Sonam Maurya

Semester: VIII

Department of Computer Science and Engineering

Indian Institute of Information and Technology, Pune

(An Institute of National Importance by an Act of Parliament)

April 2024

BONAFIDE CERTIFICATE

This is to certify that the internship project report entitled **"Experimental Investigation of CT Scan Imaging based COVID-19 Detection with Deep Learning Techniques"** submitted by **Aditya Vijay Shinde** bearing the **MIS No: 112015003**, in completion of his/her project work under the guidance of **Dr. Sonam Maurya** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering,** Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2023-24**.

Dr. Sonam Maurya	**Dr. Sanjeev Sharma**
Project Guide	Head of the Department
Assistant Professor	Assistant Professor
Department of CSE	Department of CSE
IIIT Pune	IIIT Pune

Internship Presentation held on : __/__/2024

Undertaking for Plagiarism

I, Aditya Vijay Shinde solemnly declare that research work presented in the **report** titled **“Experimental Investigation of CT Scan Imaging based COVID-19 Detection with Deep Learning Techniques”** is solely **my** research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report/dissertation has been written by **me**. I understand the zero tolerance policy of **Indian Institute of Information Technology Pune** towards plagiarism. Therefore, **I** declare that no portion of my **report** has been plagiarized and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the rights to withdraw/revoke my **B.Tech** degree.

Aditya Shinde

Date-__/__/2024

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our honorable Director, **Prof. O.G. Kakde**, for providing his kind support in various aspects. I would like to express my gratitude to my internal guide **Dr. Sonam Maurya, Department of CSE and external guide Dr. Ketan Desale,** for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this Six Months Internship. I would like to express my gratitude to the **Head of Department (**Dr. Sanjeev Sharma**)**, **Department of CSE**, for providing his kind support in various aspects. I would also like to thank all the faculty members in the **Department of CSE** and my classmates for their steadfast and strong support and engagement with this project.

Abstract

This internship report covers my experience at Machine Intelligence Research (MIR) Labs, where I learned and worked on machine learning for CT scan images of Covid using python. The report starts by explaining the introduction to the topic and detailed literature review.
Throughout the internship, I gained hands-on experience in creating machine learning and deep learning models. The proposed methodology focuses on developing a COVID-19-compatible deep learning-based medical imaging system using widely accessible CT scan images. The proposed system accepts CT scan images and classifies them into predefined categories of COVID and non-COVID. Experimental investigation with different hyper-parameters is done to evaluate the effectiveness of deep learning for COVID-19 diagnosis. This system will also perform augmentation of CT scan images and uses a dropout strategy to improve the model's generalization ability. Experiments are performed on two benchmark datasets, resulting in a maximum accuracy of 95.96% on training data and 91.93% on test data. Thus by using the readily available CT scan images and executing the deep neural network design customized for COVID-19 detection, the suggested system will be an effective AI-based medical imaging solution for COVID-19. Overall, the report reflects on what I learned and how these skills can contribute to the research at MIR lab, USA.

Keywords: - *CT Scan Images, COVID-19, Convolution Neural network, Augmentation, Dropouts*

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1
Introduction

1. Introduction

Since 2019, all countries have been struggling to come out of the negative impact caused by this virus. COVID-19 incidents have been reported in even more than 190 countries ever since then. This virus spreads rapidly, and still, it is impacting badly in the year 2022; many countries, including China, are experiencing a major outbreak resulting in many deaths [1]. The number of incidents increased because the coronavirus infection occurs within the body in just two weeks. Meanwhile, individuals unaware they would have an infected virus may pass it on to someone else upon interacting with them. Although many vaccines and screening methods have been discovered, many countries face a shortfall in healthcare resources due to its larger spread [2].

The on-going screening method for COVID-19 is the Reverse Transcription Polymerase Chain Reaction (RT-PCR) test. While RT-PCR is thought to become the standardized method for COVID-19, it could not provide a hundred percent reliable results since it looks only for the existence of specific pathogens to the virus. The RT-PCR has the limitations of being labour-intensive and highly variable test outcomes. Also, this test only confirms whether a patient is corona positive or negative. It does not state the level of corona infection and the patient's severity level. Corona infection spreads into the chest and clogs the respiratory system, thus affecting the patients' oxygen levels.

Screening methods like X-ray and CT scan screenings are useful in determining the severity level of the corona infection. Also, these methods give a fast and accurate diagnosis of COVID-19. Zhu et al. [1] have investigated the comparative performance of both kinds of imaging. They concluded that CT scan imaging performs better as they offer better quality and contrast that helps extract the most relevant information from the images. Also, several diverse studies are being suggested to identify corona signs, mostly in the lungs utilizing different deep learning image recognition systems [2-4].

The proposed method uses the CT scan images for COVID-19 diagnosis. The main contributions of the proposed report are as follows-

- Detecting COVID-19 using CT Scan imaging techniques
- CNN Model for mass screening, early and accurate diagnosis of COVID-19
- Detailed performance analysis by using train and validation data
- Executing the proposed model on two datasets of CT scan images to reassure its performance.

The organization of the report is as follows. Chapter 1 is about introduction to the topic and Chapter 2 describes the literature review. The Chapter 3 put forth the proposed methodology, Chapter 4 explains the detailed experimental setup and Chapter 5 describes the results and discussions. Finally Chapter 6 presents the conclusion and future work.

Chapter 2
Literature Review

This Chapter describes the past developments in deep learning based COVID-19 identification from chest CT scan images. Zhu et al. [1] used P-V-Net algorithm to segment the infected lung regions followed by the feature extraction to classify CT scan images. In [2], the performance of different pre-trained models is evaluated and authors concluded that models trained on out-of field datasets, boost the performance of COVID-19 diagnosis. The underlying distributions of the biomarkers and differences in model performances were studied by [3].

In [4], authors have trained numerous deep convolutional networks for unbalanced datasets to categorise X-ray images into three categories: normal, pneumonia, and COVID-19. The COVID MTNet architecture and the NABLA-N (∇^N-*Net*) technique to segment image for both CXR and CT scan images, and COVID-19 detection based on multi-task deep learning employing transfer learning were all suggested in [5]. For CXR images and CT scan images, the detection model's Inception Residual Recurrent Convolutional Neural Network provided testing accuracy of 84.67% and 98.78%, respectively. The authors also introduced a new method for quantitatively analysing the diseased area in X-ray and CT scan images.

The deep convolution network known as "COVID-Net" is suggested in [6] and made publicly available online as one of the open source networks for COVID-19 detection. It is tailored exclusively for COVID-19 diagnosis based on CXR pictures. The dataset of 13,975 CXR images with the most COVID-19 images has also been contributed by researchers. This model delved extensively into crucial aspects of COVID situations using a novel explainable technique.

The approach "DeepCOVIDExplainer" suggested for automatically diagnosing COVID-19 symptoms based on CXR pictures and classifying them into pneumonia, COVID-19, and normal cases. The ensemble of DenseNet's, ResNet's, and VGGNet's pre-processed, enhanced, and classified 16,995 CXR images [7]. The Deep Convolutional Neural Network (DCNN) used for binary classification of CXR images of pneumonia is compared to finely tailored pre-trained models of Resnet50, Inception V3, Inception ResNet V2, Xception, VGG16, VGG19, DenseNet201, and MobileNet V2. These refined models are tested using CXR and CT scan images, totaling 5856 images, of which 4273 are of pneumonia and 1583 are normal. Inception Resnet V2 and MobileNet V2 have provided the greatest accuracy of more than 96% among these Resnet50 models [8].

ResNet50 was used to extract deep features from CXR images, which were then fed to support vector machines (SVM) for COVID-19 detection. Deep networks were not used for classification because of less data availability for SVM. The COVID-19, pneumonia, and normal classes were taken into account. The outcomes were contrasted with those obtained using SVM in combination with conventional image processing techniques, such as local binary patterns plus SVM, the Gray Level Co-occurrence Matrix (GLCM) plus SVM, and the histogram of directed gradients plus SVM [8]. The ResNet50 model was also used to pre-train, fine-tune, further train, and test CT scan images. The COVID-19 cases associated with pneumonia and other medical disorders were categorised using this approach. Early detection and a low fatality rate were attained by employing this model. The study [10] described a method for segmenting lung images from CT (Computerized Tomography) scans using the U-Net architecture.

In order to battle the COVID-19, a detailed evaluation of current tactics, including diagnosis, prediction, treatment, vaccine development, and preventive measures, along with the technologies employed and their limitations is given in [11]. Additionally, this evaluation gave a comparative analysis of the various data types, cutting-edge technologies, methods used for COVID-19 diagnosis, prediction, data on contact tracing apps, and platforms for COVID-19 vaccine manufacture and identified certain difficulties and problems found in the systematic review that may help the researchers create more effective methods for regulating and preventing the spread of COVID-19.

Chapter 3

Proposed Methodology

The proposed methodology is as given in Figure 1. It takes the input of CT scan images and applies pre-processing, image augmentation and given to CNN for classification. These steps are described in detail in the following subsections.

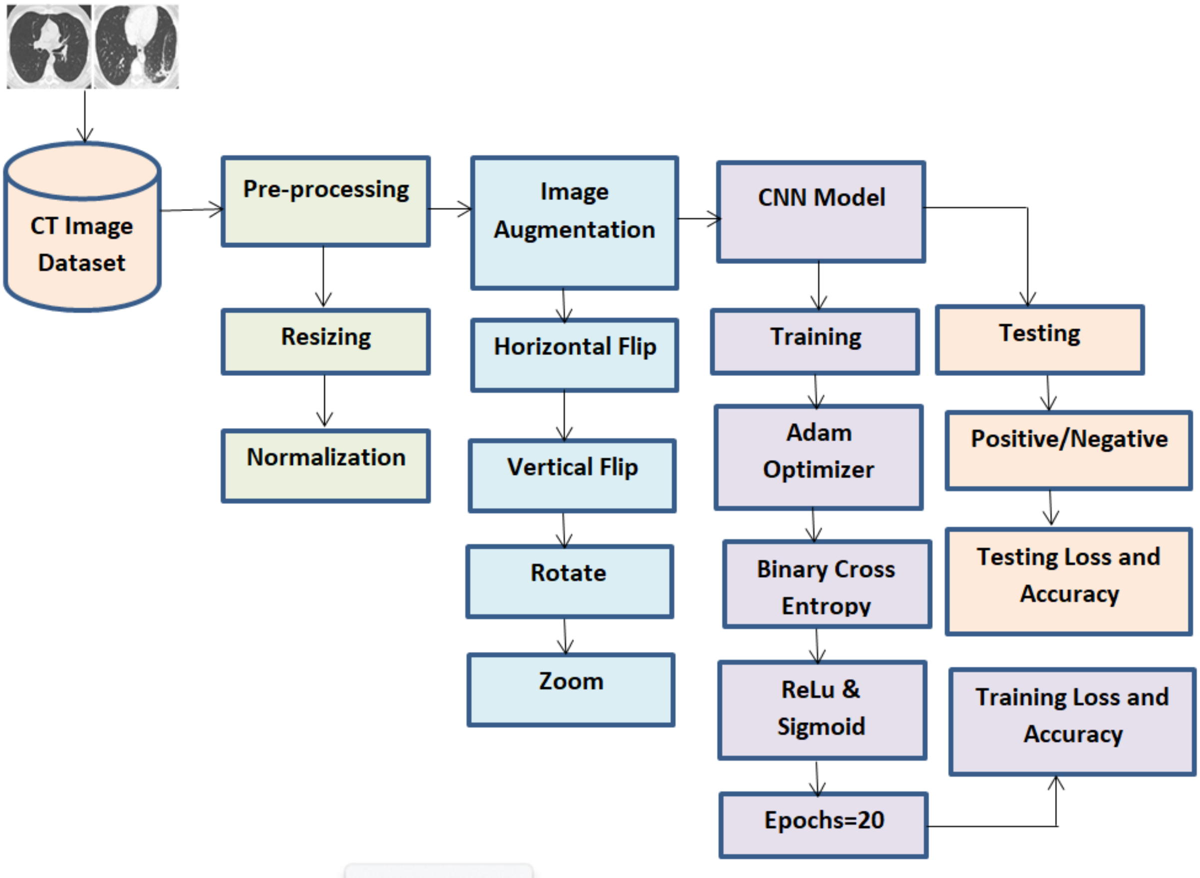


Figure 1. Proposed Methodology

3.1 Image Pre-processing

Each image is converted into the uniform size of 150*150 and normalized wherein each pixel intensity value is converted in the range of 0 to 1. The pre-processing in terms of rescaling the images and resizing them into required dimensions is done with the Batch size of 32 and input image size of 224*224.

3.2 Augmentation

Data augmentation is a powerful technique that applies number of operations on images and generates more variation in the training data so that overfitting can be avoided and more robust model can be created. This technique is mainly useful when data size is less. Because of data augmentation, generalization ability of the model gets improved. The proposed model uses the following operations as a part of augmentation-rotation, width shift, height shift, zooming and horizontal flip as stated in the Table 2. The proposed method has explored two kinds of variations in augmentation as given in Table. Augmentation-1 uses the rotation of 45 degrees and zooming of 0.50. It has changed the images considerably so there is a drop in performance of the model. So Augmentation-2 is done with rotation of 30 degrees and zooming of 0.20 and it has improved the model performance with good generalization.

Table 2. Image Augmentation Details

	Augmentation-1	**Augmentation -2**
Parameter of Image Augmentation	Values/ Range	Values/ Range
Rotation	45 degree	30 degree
Width Shift	0.15	0.15
Height Shift	0.15	0.15
Zooming	0.50	0.20
Horizontal Flip	True	True

3.3 Convolution neural network model

Convolution neural networks (CNNs) are mainly used for image classification. These are proved to be efficient and widely preferred because of both the tasks of feature extraction as well as classification by using the single architecture. CNNs have the standard structure with two main parts: first part consists of convolution and pooling layers and it is for feature extraction while second part consists of fully connected layers and it is for image classification. The number of layers of convolution, pooling and fully connected depends on the application and available computing power. Each of these layers has number of hyper parameters like number of layers, number of filters, filter size, activation functions, etc. and also training has the hyper parameters like optimizer, loss function, epochs, batch size, performance metric, etc. The proposed model uses the following values of these hyper-parameters as given in Table 3.

Table 3. Hyper-parameters used in the CNN Model

Convolution Layers	Conv1: 16 filters of 3*3 size Conv2: 32 filters of 3*3 size Conv3: 64 filters of 3*3 size Conv4: 64 filters 3*3 size Activation Function: ReLu
Input Size	150*150
Dense1 Layer	512 Nodes Activation Function: Sigmoid
Dense2 Layer	1 Node Activation Function: Sigmoid
Optimizer	Adam
Loss Function	Binary Cross entropy
Performance Metric	Accuracy
Epochs	20
Batch Size	128
Image size	150*150

After customizing the CNN with the defined hyper parameters in Table 3, it will result into the layer-wise output size and number of trainable parameters as given in Table 4.

Table 4 Layer wise listing, output shape and number of parameters

Layer	Output Shape	#Parameters
Conv2D	150*150*16	448
MaxPooling	75*75*16	0
Conv2D	75*75*32	4640
MaxPooling	37*37*32	0
Conv2D	37*37*64	18496

MaxPooling	18*18*64	0
Conv2D	18*18*64	36928
MaxPooling	9*9*64	0
Flatten	5184	0
Dense	512	2654720
Dense	1	513
Total Parameters	2,715,745	

It uses a convolution network model with four blocks of convolution layers and layers with maximum pooling. These convolution layers use 16, 32, 64 and 64 filters (each with a size of 3*3), respectively. Two dense layers with 512 and 1 nodes each are placed after these blocks. ReLu is used in the feature extraction part consisting of convolution & pooling layers while Sigmoid activation functions are used in classification part consisting of dense layers respectively.

3.4 Dropout

It is the simplest method for avoiding overfitting in networks. Some neurons are randomly selected and eliminated during network training. In other words, their activation levels are ignored in the forward pass and their weights are not updated in the backward pass of error propagation. This prevents the development of delicate models that are highly specialised to the training set of data. When a layer's dropout rate is set in terms of fractional integers like 0.1, 0.2, or 0.3, it signifies that 10%, 20%, or 30% of the nodes in the specified layer will be dropped at random. The first and last max pooling layers in this case are subject to a dropout of 30%, while a fully connected layer with 512 nodes is subject to a dropout of 10%. Combining the augmentation and dropout procedures yields the improved performance and successfully removes the overfitting problem. Its details are given in Table 5.

Table 5 Layer wise listing, output shape and number of parameters of the CNN with dropouts and augmentation

Layer	Output Shape	#Parameters
Conv2D	150*150*16	448
MaxPooling	75*75*16	0
Dropout	75*75*16	0
Conv2D	75*75*32	4640
MaxPooling	37*37*32	0
Conv2D	37*37*64	18496

MaxPooling	18*18*64	0
Dropout1	18*18*64	0
Conv2D	18*18*64	36928
MaxPooling	9*9*64	0
Dropout2	9*9*64	0
Flatten	5184	0
Dense	512	2654720
Dropout	512	0
Dense	1	513
Total Parameters	2,715,745	

3.5 Model compilation and training

Accuracy is used as a performance metric while compiling the defined model using the ADAM optimizer and binary cross entropy loss function. The assembled model is trained using training data for 20 epochs, and its performance is assessed using training and validation.

Chapter 4

Experimental setup

The suggested model framework is trained and tested using the cloud-based service Google Colaboratory, which is online and accessible. Python 3 is used in Google Colaboratory, along with supported libraries including Tensorflow, Keras, and OpenCV and hardware GPU accelerator.

Chapter 5

Results and Discussion

5.1 Experiments are performed on two different benchmark datasets of CT scan images. This is done because many time models are very much accustomed to the trained datasets and their performance degrades as soon as data from different sources changes. Dataset-I have total 2482 images out of which 1252 are COVID positive patients and 1230 are of COVID negative patients [12-13]. These images have been collected from real patients in hospitals from Sao Paulo, Brazil. Dataset-II has total 746 images out of which 349 contain the clinical findings of COVID-19 from 216 patients while 397 images are of non-COVID scans [14]. These images are in .png and .jpg format. The details of these datasets are given in Table 1.

Table 1. Dataset Information

Sr. No.	Dataset	URL	Class	Number of images	Total Images
Dataset-I	**SARS-COV-2 Ct-Scan**	www.kaggle.com/plamenedua rdo/sarscov2-ctscan-dataset	Positive	1252	2482
			Negative	1230	
Dataset-II	**COVID-CT**	https://gas.graviti.com/dataset/ graviti/COVID_CT	Positive	349	746
			Negative	397	

Both the datasets are described in Table 1. Subsection 5.1 describes the results on dataset-I and Subsection 5.2 on dataset-II. Sample input images of dataset-I and dataset-II are as given in Figure 2 and 3 respectively. For each of these datasets, two variations of experiments are performed- without augmentation and with augmentation. In the experiments with Augmentation-I (rotation=45, zoom=0.50) and Augmentation-II (rotation=30, zoom=0.20) are performed.

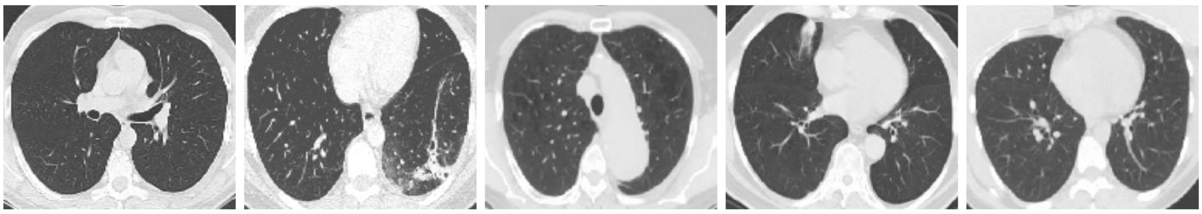


Figure 2. Sample Images of Dataset-I

Figure 3. Sample Images of Dataset-II

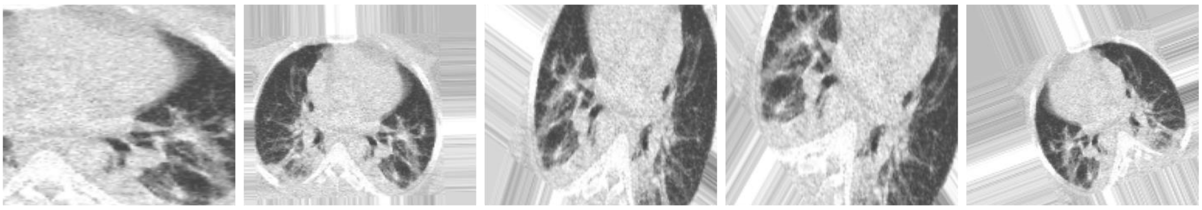


Figure 4. Sample Images Dataset-I after Augmentation-I (rotation=45, zoom=0.50)

Figure 5. Sample Images Dataset-I after Augmentation-II (rotation=30, zoom=0.20)

5.1 Experimental results on Dataset-I

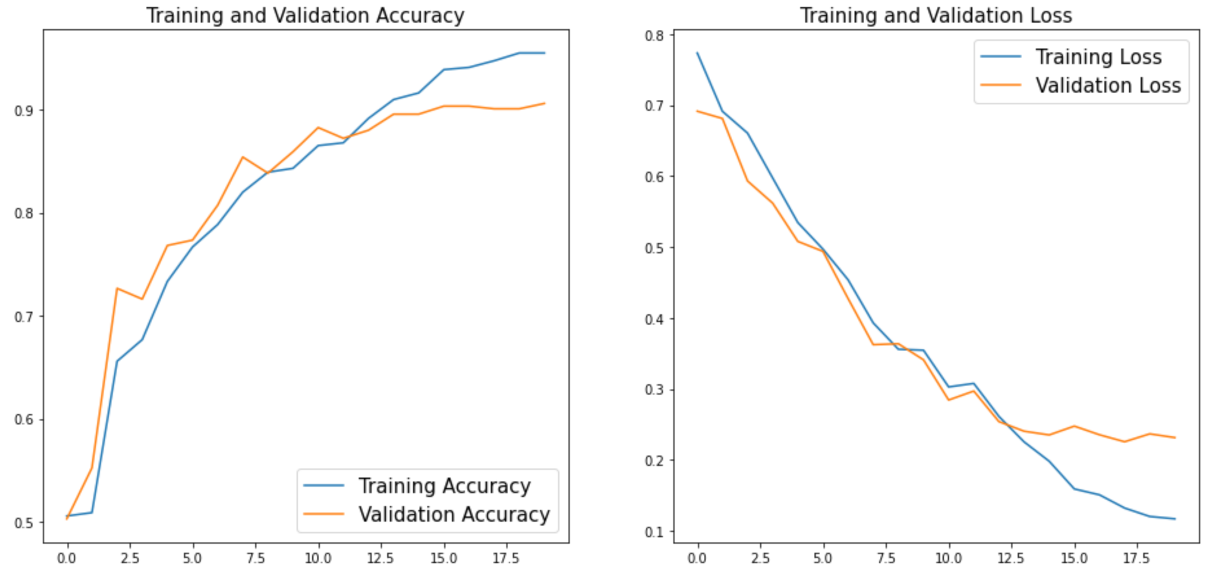
Table 6 shows the training and validation accuracies & losses for all the three variations of experimentations as listed in the columns. These performances are recorded for 20 epochs. From the results, it is clear that accuracy without augmentation is higher. Augmentation-I (rotation=45, zoom=0.50) has resulted into more challenging images as shown in Figure 4 than that of less challenging images as in Figure 5 with Augmentation-II (rotation=30, zoom=0.20). Although the accuracy with augmentation is slightly less, it has produced the robust model that can handle images with variations in appearances. Also, it has reduced the overfitting wherein the difference between training and validation accuracy is less than in the first case without augmentation.

Table 6. Training and Validation Performance for Dataset-I

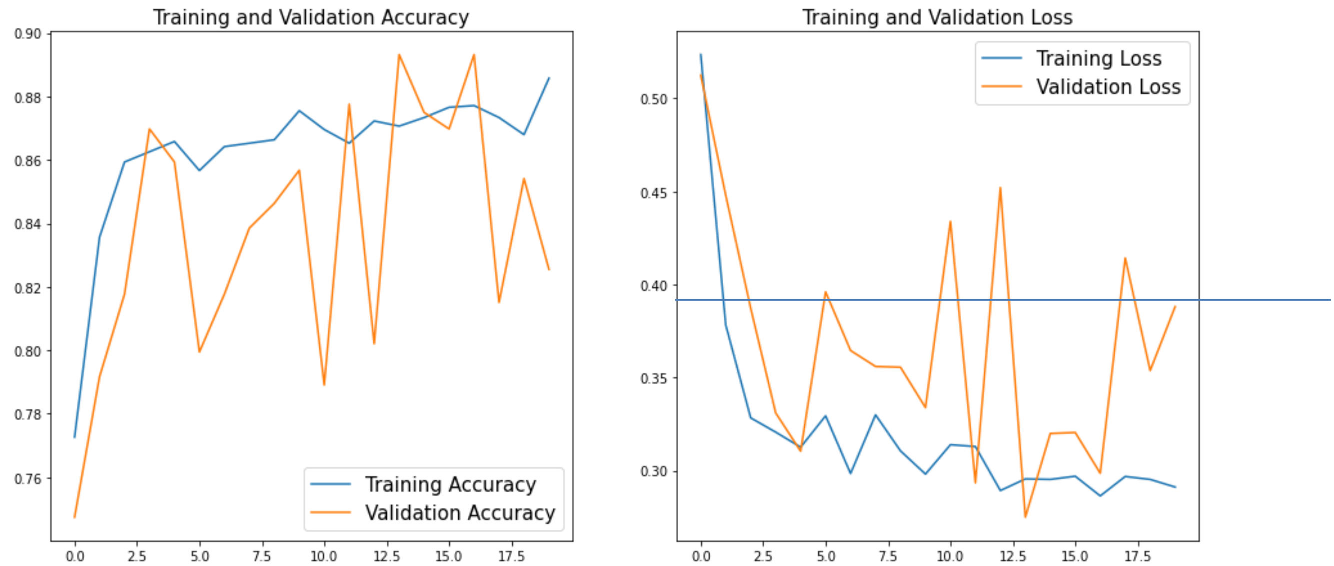
Without Augmentation and Dropouts						
Epoch#	Training Accuracy	Testing Performance				
		Accuracy	Loss	Precision	Recall	F1-Score
1	50.54	50.26	69.15	0	0	0
5	73.33	76.82	50.79	0.86	0.63	0.72
10	84.32	85.94	34.09	0.87	0.86	0.86
20	95.53	90.62	23.12	0.90	0.91	0.90

With Augmentation and Dropouts [Rotation=30, Zoom=0.20]						
Epoch#	Training Accuracy	Testing Performance				
		Accuracy	Loss	Precision	Recall	F1-Score
1	77.26	74.74	51.25	0.68	0.94	0.79
5	86.58	85.94	31.04	0.84	0.65	0.77
10	87.55	85.68	33.39	0.90	0.79	0.84
15	87.34	89.50	27.50	0.91	0.85	0.86
20	88.58	82.55	39.37	0.91	0.70	0.80

These results are clearly visualized in the following plots given in Figure 6.



(a)



(b)

Figure 6. Dataset-I: Plots of Accuracy and Loss- a) Without Augmentation b) With Augmentation-I c) With Augmentation-II

5.1 Experimental results on Dataset-II

Table 7 shows the training and validation accuracies & losses for all the three variations of experimentations on Dataset-II. The performances for this dataset too are recorded for 20 epochs so that results can be compared with Dataset-I. From the results, it is again clear that the accuracy without augmentation is higher as images are simpler and not tilted. Performance of Augmentation-I is lesser as it feeds more tougher and tilted images as given in Figure 7 than that of Augmentation-II as given in Figure 8.

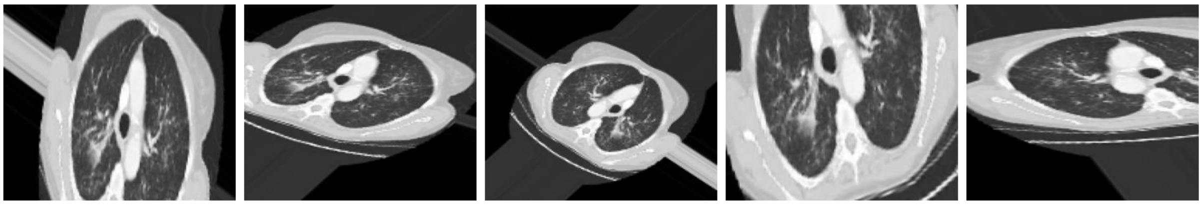


Figure 7. Sample Images Dataset-II after Augmentation-I (rotation=45, zoom=0.50)

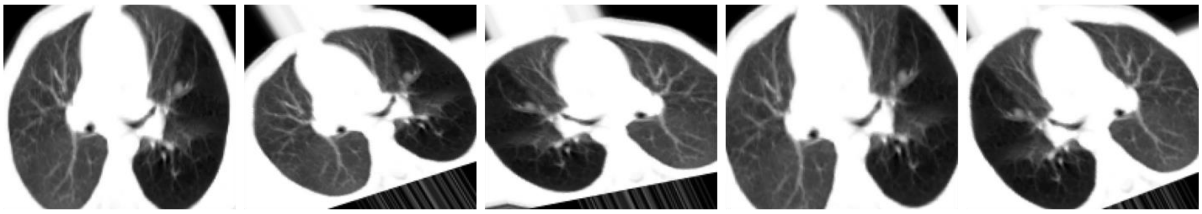


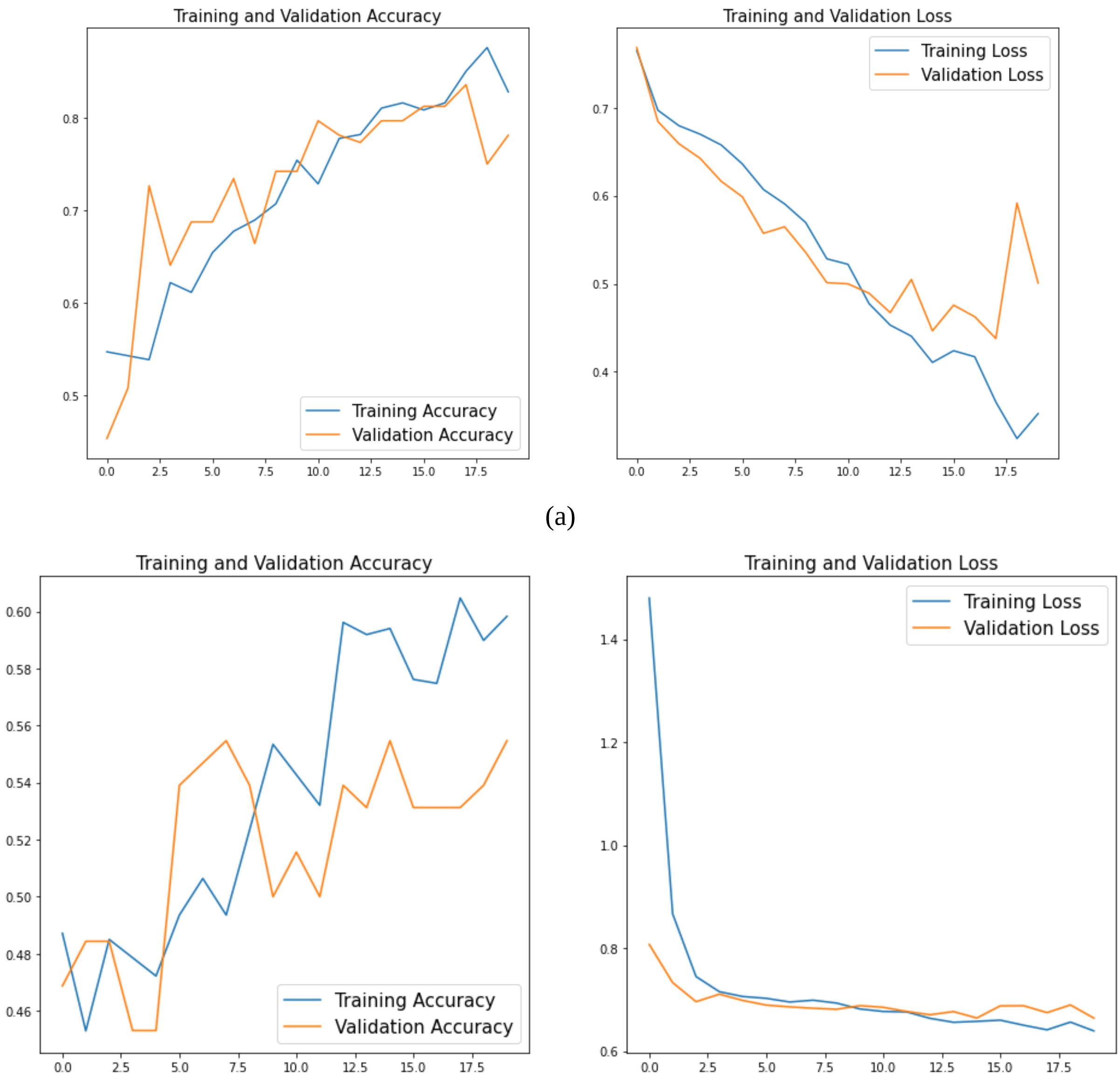
Figure 8. Sample Images Dataset-II after Augmentation-II (rotation=30, zoom=0.20)

Table 7. Training and Validation Performance for Dataset-II

Without Augmentation and Dropouts						
Epoch#	**Training Accuracy**	**Testing Performance**				
		Accuracy	**Loss**	**Precision**	**Recall**	**F1-Score**
1	54.70	45.31	76.987	0	0	0
5	61.13	68.75	61.66	0.64	0.88	0.74
10	75.43	74.28	50.13	0.72	0.87	0.79
15	81.62	79.69	44.65	0.76	0.89	0.82
20	82.81	78.12	50.08	0.74	0.91	0.82
18	85.04	83.59	43.78	0.79	0.92	0.85

With Augmentation and Dropouts [Rotation=30, Zoom=0.20]						
Epoch#	**Training Accuracy**	**Testing Performance**				
		Accuracy	**Loss**	**Precision**	**Recall**	**F1-Score**
1	48.72	45.88	80.72	0	0	0
5	47.22	45.31	69.84	0	0	0
10	55.34	50	68.84	0.5	1	0.66
15	59.40	55.47	66.43	55.47	1	0.71
20	59.83	55.47	66.42	0.55	1	0.71

The plots of these experimentation are given in Figure 9.



(b)

Figure 9. Dataset-II: Plots of Accuracy and Loss- a) Without Augmentation b) With Augmentation-I c) With Augmentation-II

After comparing the performance of the same model on Dataset-I and Dataset-II, it is observed that the accuracy of Dataset-I is higher than that of Dataset-II. This is because Dataset-I has 2482 images and Dataset-II has 746 images. So the model has learned well for Dataset-I with good generalization ability.

Comparison with GLCM

Classifier/Model	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor	62.7	0.625	0.63	0.63
Decision Tree	72.2	0.725	0.72	0.72
Naive Bayes	60.9	0.615	0.61	0.61
Support Vector Machine	61.5	0.63	0.615	0.605
Logistic Regression	63.5	0.635	0.635	0.635
Random Forest	79.6	0.8	0.8	0.795
Gradient Boosting	75.8	0.755	0.76	0.76
Ada Boosting	73.8	0.74	0.735	0.735
Artificial Neural Network	50.9	0.56	0.505	0.37
Proposed Model	**90.62**	**23.12**	**0.90**	**0.91**

Classifier/Model	Accuracy	Precall	Recall	F1-Score
K-Nearest Neighbor	59.3	0.6	0.59	0.58
Decision Tree	64.6	0.65	0.645	0.645
Naive Bayes	61.3	0.655	0.605	0.575
Support Vector Machine	56.0	0.585	0.555	0.51
Logistic Regression	63.3	0.645	0.63	0.62
Random Forest	68.6	0.69	0.685	0.68
Gradient Boosting	63.3	0.64	0.63	0.625
Ada Boosting	64.6	0.65	0.645	0.645
Artificial Neural Network	54.6	0.695	0.535	0.42
Proposed Model	**83.59**	**0.79**	**0.92**	**0.85**

Chapter 6

Conclusion

S This study has demonstrated the efficacy of deep learning for diagnosing COVID-19. The convolution network, a kind of deep neural network, is trained on patient CT scan images. By changing its hyper characteristics, this report explains the experimentation details with CNN. The use of augmentation and dropouts has made the network robust. The highests accuracy of 95.96% accuracy is attained and it can further be improvised by increasing the dataset size and executing the model for higher number of epochs on GPU powered machines. Also segmenting the region of interest will contribute in making the system more adaptable in real world scenarios.

References

1. Zhu F, Zhu Z, Zhang Y, Zhu H, Gao Z, Liu X, Zhou G, Xu Y, Shan F. Severity detection of COVID-19 infection with machine learning of clinical records and CT images. Technol Health Care. 2022; 30(6):1299-1314. doi: 10.3233/THC-220321. PMID: 36314176.
2. Zhao, W., Jiang, W. & Qiu, X. Deep learning for COVID-19 detection based on CT images. *Sci Rep* **11**, 14353 (2021). https://doi.org/10.1038/s41598-021-93832-2
3. Goncalves, J., Yan, L., Zhang, HT. *et al.* Li Yan et al. reply. *Nat Mach Intell* **3**, 28–32 (2021). https://doi.org/10.1038/s42256-020-00251-5
4. Mohammad Rahimzadeh, Abolfazl Attar,A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2, Informatics in Medicine Unlocked,Volume 19, 2020,100360,ISSN 2352-9148,https://doi.org/10.1016/j.imu.2020.100360.
5. Alom, Md. Zahangir&Rahman, M MShaifur&Nasrin, Mst&Taha, Tarek&Asari, Vijayan. (2020). COVID_MTNet: COVID-19 Detection with Multi-Task Deep Learning Approaches.
6. Gao, Terry & Wang, Grace. (2020). Chest X-ray image analysis and classification for COVID-19 pneumonia detection using Deep CNN. 10.21203/rs.3.rs-64537/v1.
7. M. R. Karim, T. Döhmen, M. Cochez, O. Beyan, D. Rebholz-Schuhmann and S. Decker, "DeepCOVIDExplainer: Explainable COVID-19 Diagnosis from Chest X-ray Images," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 1034-1037, doi: 10.1109/BIBM49941.2020.9313304. [6] Rodolfo M. Pereira, Diego Bertolini, Lucas O. Teixeira, Carlos N. Silla, Yandre M.G.
8. Idri A. Asnaoui KE, Chawki Y. Automated methods for detection and classification pneumonia based on x-ray images using deep learning. arXiv preprint arXiv:2003.14363, 2020
9. Walvekar, Sanika and Shinde, Dr. Swati, Detection of COVID-19 from CT Images Using resnet50 (May 30, 2020). 2nd International Conference on Communication & Information Processing (ICCIP) 2020, Available at SSRN: https://ssrn.com/abstract=3648863 or http://dx.doi.org/10.2139/ssrn.3648863
10. S. Walvekar and S. Shinde, "Efficient Medical Image Segmentation Of COVID-19 Chest CT Images Based on Deep Learning Techniques," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 203-206, doi: 10.1109/ESCI50559.2021.9397043.
11. Weiping Ding, Janmenjoy Nayak, H. Swapnarekha, Ajith Abraham, Bighnaraj Naik, Danilo Pelusi, Fusion of intelligent learning for COVID-19: A state-of-the-art review and analysis on real medical data, Neurocomputing, Volume 457, 2021, Pages 40-66, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2021.06.024.
12. Soares, Eduardo, Angelov, Plamen, Biaso, Sarah, Higa Froes, Michele, and Kanda Abe, Daniel. "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification." medRxiv (2020). doi: https://doi.org/10.1101/2020.04.24.20078584.
13. Angelov, P., & Soares, E. (2020). Towards explainable deep neural networks (xDNN). Neural Networks, 130, 185-194.
14. Zhao J, Zhang Y, He X, Xie P. COVID-CT-Dataset: A CT Scan Dataset about COVID-19. PPR:PPR346150.