Norwegian University
of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty of Science and Technology (Realtek)

Identification of Coronavirus Through CT-scan Images Using Supervised and Semi-supervised Learning

Rinju Manandhar
Master of Science in Data Science

This page is intentionally left blank.

Acknowledgements

I am extremely grateful to our supervisor **Habib Ullah** (associate professor) and co-supervisor **Fadi Al Machot** (associate professor) for their continuous guidance and support. Without their feedback and supervision, this work would not have been accomplished on time. I am thankful towards the faculty of science and technology of Norwegian University of Life Sciences (NMBU) for providing this opportunity to explore the theoretical knowledge into practical understandings. I would also like to present my gratitude to the employees of writing centre at NMBU for guiding with the necessary templates and writing techniques for this thesis report. I am thankful to my friends especially Aditya Dey, Nissan Karki, Sanam Maharjan and Krishna Mohan Shah for sharing their criticism and useful knowledge on this work that helped in the completion of this thesis. Last but not the least, I would like to thank my family for their continuous encouragement as without their support it would have been a very tough journey.

Norwegian University of Life Sciences

Ås, Spring 2023

Rinju Manandhar

Abstract

COVID-19 is a transmissible disease originated from SARS-CoV-2. Affecting the respiratory system, this virus infected millions of people around the world in a short span of time. Primarily, laboratory test was the only medium for detecting covid infection. However, with the need of early diagnosis, deep learning models are introduced for COVID detection. In this paper, we focus on CT scan images of lung to detect the existence of coronavirus in human body.

We explored the four different deep learning supervised models namely DenseNet201, ResNet50, CNN_model_1 (five blocks) and CNN_model_2 (seven blocks) for the identification of COVID-19. We used CT-scans dataset having 2481 images from kaggle (1252 COVID and 1229 non-COVID). We experimented transfer learning with DenseNet and ResNet models by using pre-trained weights of ImageNet and fine-tuned them. We determined the COVID and non-COVID labels as 0 and 1 consecutively. Analyzing the performance of all supervised models with and without image augmentation, DenseNet-201 provided the best accuracy of 99.2% with 0.9922 recall without augmentation.

Additionally, we explored semi-supervised learning with EfficientNet-B4 model by utilizing the noisy-student weights. We analyzed this model along with DenseNet201 (supervised model) using only 600 samples from the original dataset having 60 no. of labeled samples. In this case, we attained highest accuracy of 76.67% and recall of 0.88 for semi-supervised approach without augmentation. Thus, semi-supervised techniques can perform better when there are only limited labeled samples available.

Keywords – COVID-19, CNN, DenseNet, ResNet, EfficientNet, transfer learning, image augmentation, supervised learning, semi-supervised learning

Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Meaning
CNN	Convolutional Neural Network
SSL	Semi-supervised Learning
DenseNet	Densely Connected Convolutional Networks
Resnet	Residual Neural Network
ReLU	Rectified non linear unit
Adam	Active Directory Application Mode
AdaMax	Adaptive moment estimation with Maximum
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
TPR	True Positive Rate
no.	Number
Val	Validation data
vs	Versus

Chapter 1

Introduction

1.1 Background

Coronavirus is a global pandemic which was first reported in December 2019 and named as COVID-19. This virus developed from SARS-CoV-2 is contagious in nature and its mutated versions over time are proven to be highly transmissible and effective than the previous ones. A report stated that more than 3.3 million people were affected by May 2, 2020 with death toll surpassing approximately 238,000 [1].

RT-PCR (reverse-transcription polymerase chain reaction) is the laboratory screening technique for the medical diagnosis of COVID-19 approved by world health organization (WHO) [2]. To conduct this laboratory test, first the health professionals collect mucus or swab samples from the throat/nose of the patient. These samples are then sent to the laboratory where lab technicians tests for chemical reactions to confirm the presence of coronavirus in the attained samples.

Many scientists have worked together to discover the proper medicine/vaccine to fight against this virus but unfortunately, no medicine and/or vaccine is found to be 100% effective to cure or prevent this disease. Since coronavirus is highly contagious disease, early detection of this virus in human body served as main factor to avoid its transmission by taking necessary measures after diagnosis. Time is an important factor for the identification of COVID-19 to implement proper isolation approaches and medical treatments for saving the patients and avoiding further outspread of COVID-19. But,

the manual method of RT-PCR test takes more than 24-hours to provide the result and it also requires multiple sample collection of same patient at multiple times to carry out the test more than one-time for confirming the positive/negative case. Therefore, to handle these errors and cover the necessity of early detection of COVID-19, scientists and researchers introduced deep learning approaches by processing medical images of chest X-rays and CT-scans.

1.2 Problem Statement

Fig. 1.1 illustrates the available mediums for coronavirus identification. At the primary stage of COVID-19 outbreak, reverse transcription polymerase chain reaction (RT-PCR) method is used which is totally based on laboratory testing of mucus/swab samples collected from nose or throat of human. However, this standard practice is time consuming as it needs to be done manually one-by-one for each sample and it also has high records of false results (detecting covid positive patients as negative/non-covid cases) caused by sample errors [3].

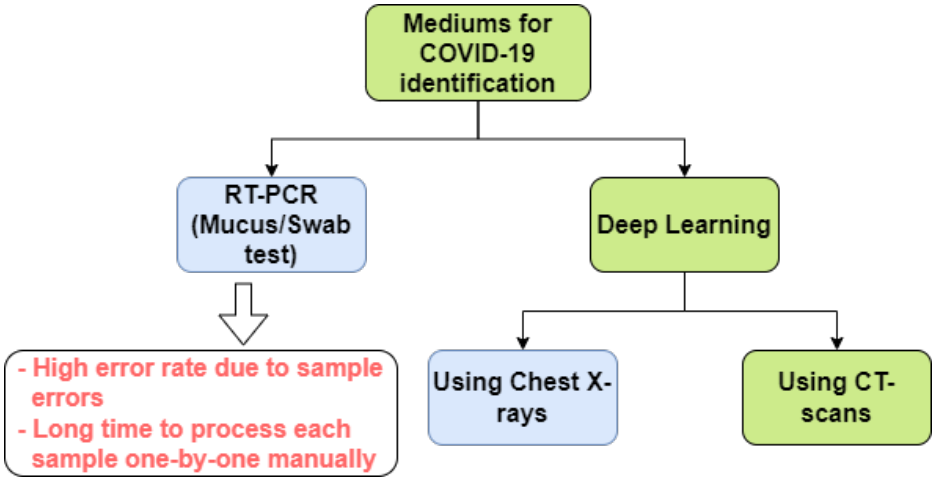


Figure 1.1: Illustration of various mediums used for COVID-19 identification.

Fang et al. [4] have compared the chest CT-scan sensitivity with viral nucleic acid (viral pneumonia) as viral pneumonia is also a major symptom for COVID-19. This experiment took throat swab samples of 51 patients and performed RT-PCR on the samples. Among them, 36 samples had positive outcomes for coronavirus in the

first test while during second test of RT-PCR, only 12 had COVID positive results. However, the CT-scans of those 51 patients showed that 98% of people had COVID-19 i.e. 50 out of 51 patients had COVID-19 in that scenario.

Moreover, the paper published by Xie et al. [5] includes the task of performing CT-scan of five people who were detected as COVID negative cases when tested with RT-PCR. But eventually, the findings of CT-scan contradicted the RT-PCR results. Only after conducting multiple RT-PCR tests, the result showed the patients had COVID-19. Thus, it indicated that RT-PCR test had only 63% of positive rate with high value of false-positives.

Thus, to overcome these obstacles, researchers proposed various deep learning models such as CNN (Convolutional Neural Network) [6], DenseNet [7], ResNet [8], EfficientNet [9] etc. which can be fitted to the training images gathered from reported cases of COVID-19 patients. The images can either be chest X-rays or CT-scans (Computed Tomography scans) [10]. Both of them are advanced medical imaging techniques for detecting any morphological patterns in the scanned image of lung lesions to identify the presence of coronavirus [2]. With the training of such deep learning models, we attain more accurate and timely results for the detection of COVID-19 [11] leading towards the adaption of necessary measures such as proper isolation to avoid further spread of the virus to other individuals.

Computed Tomography (CT scan) is frequently used technology in medical imaging which can provide more detailed view of the imaged organs using multiple X-ray series. With the development in medical science, the availability of CT scanning technology has increased. Therefore, the data of lung lesions recorded from CT-scans can be used as the appropriate medium for COVID-19 detection. Convolutional Neural Network (CNN) is a deep learning algorithm to exhibit the performance of deep neural network in human brain [12]. CNN is widely implemented for image processing thus, chest x-rays or CT-scans are often processed through CNN to identify any abnormalities deriving towards the presence of coronavirus.

In this work, we would like to use CT-scan images with deep learning models. For this purpose, we have different deep learning models supporting supervised and semi-supervised learning approaches. There are many pre-trained CNN models available for transfer learning as they can be fine-tuned by cutting out their last layer and adding other layers as per the relative problem. Thus, we can utilize the concept of transfer learning to use the knowledge from already learnt model as per the relevancy of the

task [13]. This concept is utilized to avoid the necessity of learning different patterns in the input data by making use of already known knowledge from pre-trained models. ImageNet is a popular dataset used for training various CNN models such as VGG [14], DenseNet[7], ResNet [8] etc. The weights of ImageNet can be used publicly which has the classification for thousands of images.

Besides ImageNet, we also have Noisy-Student weights that is used by EfficientNet (B4 to B7 versions) for semi-supervised learning [15][16]. Semi-supervised learning uses limited number of labelled data for training along with large number of unlabelled training data to predict the unlabelled test data [17]. Since the prediction is only based on small quantity of labelled training data, the model has to learn on its own for other unlabelled training which enhanced the concept of semi-supervised learning. Thus, this technique is utilized for COVID detection which provided the improved reliability in diagnosis of coronavirus [18].

Moreover, we have also considered the impact of data augmentation on the performance of different deep learning models. To conduct this study, we have used zoom, rotation etc. for creating the augmented images. Data augmentation can enhance the accuracy of the model based on newly generated data by performing random flips, zoom, rotation, crop etc. on the existing data [19]. So, increasing the number of images/dataset through data augmentation also plays vital role in the research of performance of deep learning models for COVID-19 detection.

1.3 Thesis Objectives

Our main targets proposed for this thesis are:

- To explore various supervised deep learning models such as CNN, DenseNet-201 and ResNet-50 for the detection of COVID-19 with CT-scan images.
- To analyze the effect of data augmentation in the performance of all utilized deep learning models in this work for the diagnosis of coronavirus.
- To explore the impact of semi-supervised learning in COVID-19 identification.

Chapter 2

Related Works

We carried out the extensive study of literature on the related methods of COVID-19 detection. On the basis of our research, we classified the literature into three main categories: deep learning, transfer learning and semi-supervised learning.

2.1 Deep Learning

Deep learning approach is the mostly used technique for the proper analysis of COVID-19 to attain the results on time and with more accuracy. Although there are numerous deep learning models available till date, all those models are based on the concept of convolutional neural network (CNN). There are two mediums used for the identification of coronavirus which are: Chest X-rays and CT-scans.

Benmalek et al. [20] have conducted the comparison of chest X-ray and CT-scan imaging in the context of detection of coronavirus. Here, the authors used three deep learning models (ResNet18, Inception-V3, MobileNet-V2) with chest X-rays as well as CT-scan images to test the performance for COVID-19 detection. This paper concludes that ResNet18 showed higher precision of 98.5% followed by Inception-V3 with 97.4% specificity when implemented with CT-scan images.

Since CT-scan images are more advanced imaging technology than chest X-ray providing the combination of series of X-ray images, CT-scans are better to be used for

enhanced view of the internal organ. Therefore, we work with CT-scan images in this paper with multiple deep learning models.

Convolutional Neural Network (CNN) is a deep learning model which has the ability to process data with images for extracting the general characteristics/patterns in the given dataset. Fig. 2.1 below represents the generic architecture of CNN.

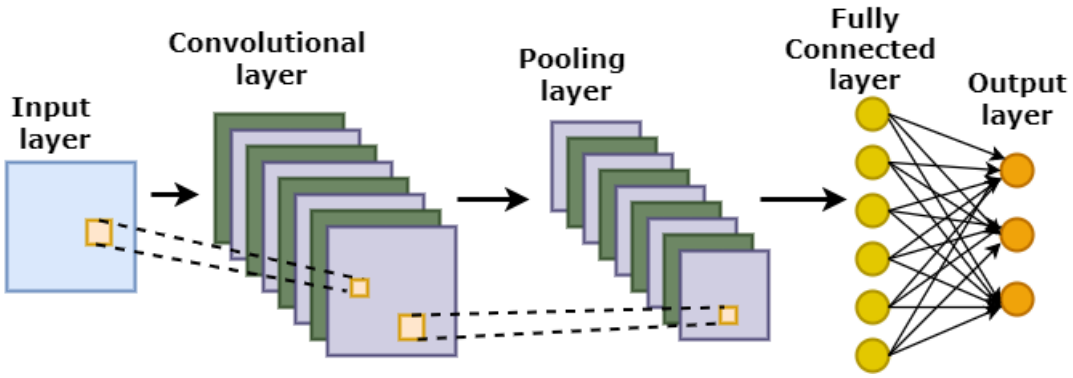


Figure 2.1: General architecture of CNN.

As shown in fig. 2.1, CNN has five major layers which are: input, convolutional, pooling, fully-connected (FC) and output layer. The first layer of CNN is used for up-sampling and usually known as input layer. It contains the size of filter with numerous filters in different convolution layers that aid to convolve the input image for creating feature maps. These filters provide the ability to learn features from the provided inputs. Every layer of convolution incorporates a function for activation that helps to classify the objects to certain labels. Pooling layer works to decrease the size of received output from earlier convolution layer. Thus, it down samples the attained outcomes of convolution layer [21]. Then fully-connected layer combines the outputs of previous convolution layers and flattens the output nodes. Then it connects every node with each distant node which further decides on classification via activation functions (sigmoid and softmax for binary and categorical classification respectively). Then at last the output layer determines number of classification categories applicable for the particular dataset.

With the advancement in deep learning models for image recognition, CNN is mostly used for determining COVID-19 results through CT-scans/chest X-rays. Tumuluru et al. [22] have explained the CNN model used for the classification of COVID-19 patients (positive or negative) with the use of CT-scan images. They used three CNN

models with increasing number of filters among which the third model attained highest accuracy of 88.15%. Thus, it concluded that CNN model can be used for attaining reliable results within short period of time to identify the presence of COVID-19 through CT-scans.

Rajawat et al. [23] have introduced a CNN model for COVID identification based on image processing (C-COVIDNet). For this model, chest x-rays were used for training as a three-class classification problem (COVID-19, normal and pneumonia). Here, the CNN model was able to attain 97.5% accuracy with 91.91% of f1-score.

With the overall literature study regarding CNN and its usage in the context of COVID-19 detection, we acknowledged that lightweight CNN is powerful tool to work with image dataset. It is capable of discerning the patterns in the trained images efficiently with less effort and time. Thus, CNN can be used for the identification of coronavirus through CT-scan images.

2.2 Transfer Learning

Transfer learning implements the information that is already known or learnt by other model relevant to the existing problem scenario. The usage of pre-trained model boosts the memory utilization as the system can use the weights those pre-trained models for solving the current problem by adding some fine-tuned layers at the end of transferred model.

Fig. 2.2 represents the general outline for transfer learning which shows how knowledge is transferred from a pre-trained model to a new fine-tuned model to perform a certain task. To perform transfer learning, we take an existing model which is already trained for a certain problem scenario with large amount of data and use its weights in the new model based on our current problem. This allows to omit the need of high time consumption for model training by utilizing the already learnt weights from a pre-trained model. Then we cut the final layers from the old model and fine-tune it with new layers as per the requirement of current problem domain.

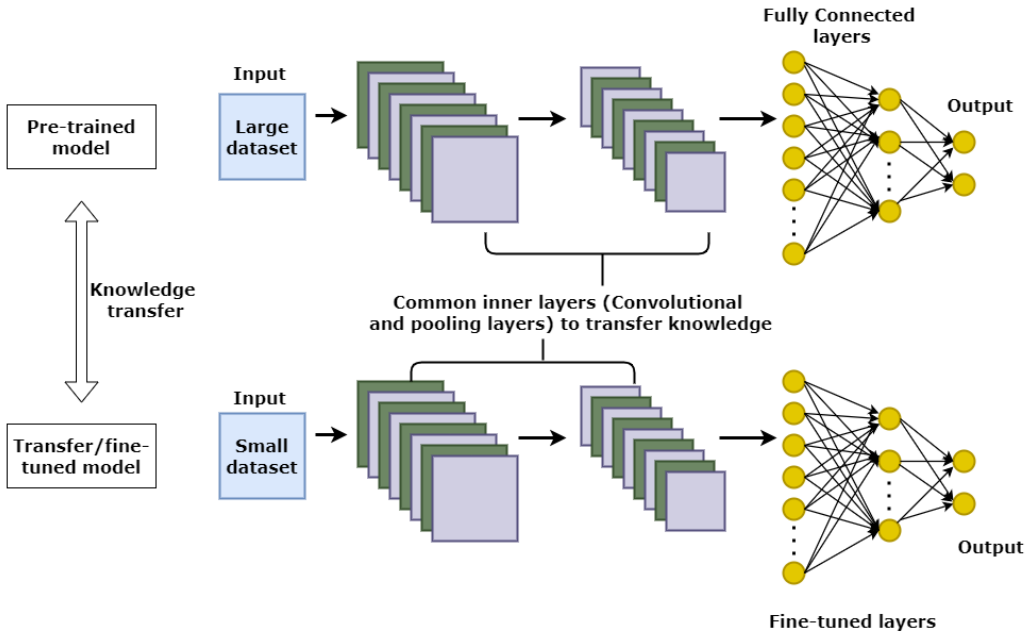


Figure 2.2: Basic outline of transfer learning.

For transfer learning, we have publicly available dataset of ImageNet containing approximately 20,000 different objects for categorizing the millions of images [24]. This dataset is used by many researchers to train distinct convolution neural networks. In case of image recognition, we have various pre-trained models based on CNN-architecture that uses ImageNet to perform image classification for instance: VGG [14], DenseNet [7], ResNet [8] etc.

Shaha and Pawar [25] have described the use of transfer learning by fine-tuning the pretrained network of VGG-19 to perform classification of images. This study showed the comparison of image classification with CNN and transfer learning based on the two databases (i.e. CalTech256 and GHIM10K). This paper summarized that overall performance of transfer learning network (pre-trained VGG19 with fine-tuning) was better than the normal CNN architecture in terms of image classification.

Hussain, Bird and Faria [26] have performed the research on CNN model having architecture of Inception-V3 for evaluating its performance through transfer learning on image classification. This study concluded that transfer learning provides sufficient accuracy even when training dataset is small since the pre-trained weights of imagenet is used for learning the necessary characteristics of images.

Among largely used models for transfer learning, DenseNet and ResNet are two high performing models which are pre-trained on image-classification data of ImageNet. Generally, DenseNet is known as densely connected neural network developed on the basis of CNN (Convolutional Neural Network). Fig 2.3 illustrates the basic architecture of a densenet model. The layers are densely connected with each-other making a deep fully-connected layers. CNN having more depth can have better accuracy if they incorporate shorter paths connecting the layers around input and output. In a densenet model, each layer is connected to every other layer in feed forward manner which allows the flow of feature map from the prior layers to every succeeding layer [27]. This makes a dense block of densely connected layers which further connects to a transition layer. Transition layer incorporates various convolutional and pooling layers for downsampling the size of feature map attained from a dense block.

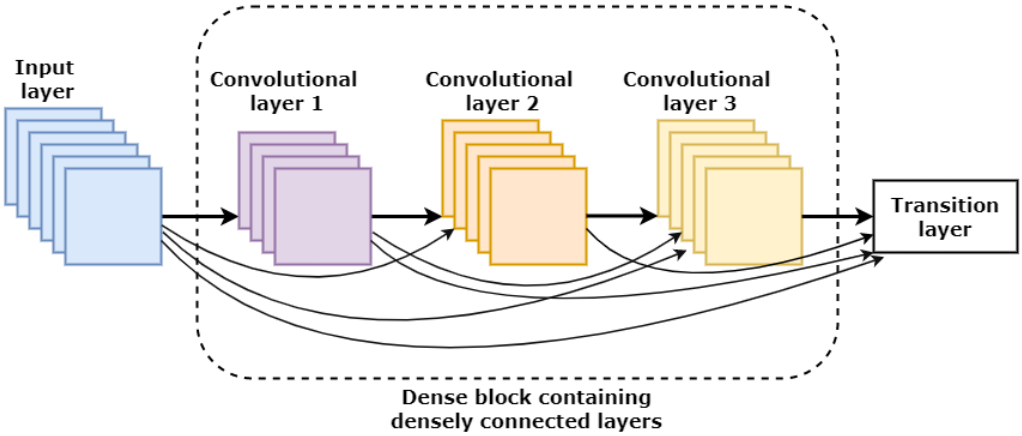


Figure 2.3: Basic architecture of DenseNet showing densely connected layers.

Huang et al. [7] have performed research on densely connected CNN (DenseNet) containing feed forward connection between layers that connects all layers with each-other. The matching feature maps of each layer was used to determine the input nodes. This paper used ImageNet along with other three dataset as benchmark which showed remarkable improvement in performance when DenseNet is implemented.

Singh et al. [28] have explained the experiment to classify COVID-19, pneumonia, tuberculosis and healthy labels through CT scans using VGG16, DenseNet and ResNet152V2. This study revealed that ensemble densely connected networks outperformed the accuracy as compared to other models achieving 96.28% accuracy. Like-

wise, Zhong et al. [29] have proposed the DenseNet model for classification of cancer images. It used the kaggle dataset of cancer cell images (PCam) for binary classification and compared the performance of dense network with VGG19 and ResNet34. The result showed that DenseNet provided higher accuracy for the image classification.

Residual Neural Network (ResNet) is also based on CNN-architecture which contains connection of much deeper layers. It simply utilizes the blocks of CNN numerous times. The convergence of deeper networks results in saturation of accuracy and increase in error during the training which leads to the degradation problem [8]. This situation can be avoided with the implementation of residual network architecture. Fig. 2.4 represents a general architecture of residual network. It shows that in ResNet, we connect the activation of previous layer to the next layer's pre-activation. This aids to flow the required information from previous layer to later layer without taking all weights.

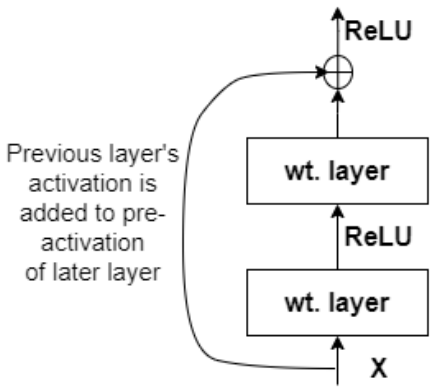


Figure 2.4: General architecture of ResNet.

Farooq and Hafeez [30] emphasized on the usage of residual network for COVID-19 screening using medical radiograph images (Chest X-rays). This paper presents fine-tuned ResNet-50 model to process the chest x-ray images for the diagnosis of COVID-19. The proposed model was able to achieve 96.23% accuracy with 41 epoch demonstrating the efficiency of ResNet for image classification.

From the comprehensive study on transfer learning, we understood that implementation of pre-learned networks and their weights save time of training a new model. Moreover, it can also enhance the evaluation accuracy with proper fine-tuning. Since, densely connected network and residual network provide improved accuracy when

more deeper layers are used, we can implement these network architectures for the image classification of COVID-19 through CT-scans.

2.3 Semi-supervised Learning

Supervised learning is mostly popular while working with data at a learning stage however, in practical world not all data are perfectly labeled. Hence, we have to deal with more no. of unlabeled data in real-life problems. Fig. 2.5 illustrates the general concept of supervised, unsupervised and semi-supervised learning. As per this figure, supervised learning contains all labeled data for the training while unsupervised learning only has unlabeled data. In between these two, we have semi-supervised learning approach. Semi-supervised learning is the combination of supervised and unsupervised learning. It takes a limited number of labeled data that are in small quantity along with larger quantity of unlabeled data to train a model. The concept of semi-supervised learning is to simply train the model with small labeled and large unlabeled data to diminish the need of arranging large labeled dataset [31].

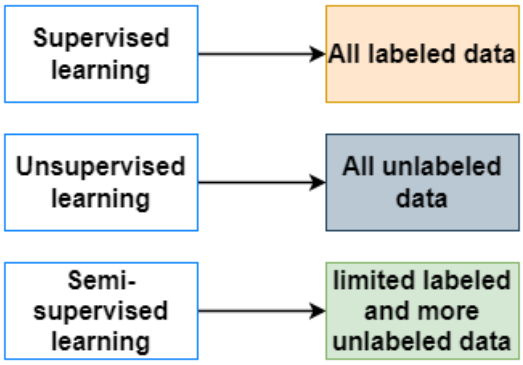


Figure 2.5: General concept of supervised, unsupervised and semi-supervised learning.

In the paper published by Keh [16], EfficientNet architecture was used along with the weights of noisy student to improve the image classification accuracy for plant-pathology. The study revolved around the comparison of pre-trained VGG16, DenseNet-161 and ResNet-101 for plant image classification. These model achieved the standard benchmark of 0.945 for those pre-trained models. However, the further exploration to semi-supervised learning allowed the implementation of EfficientNet with noisy-

student weights and this resulted in the improved accuracy upto 0.962 outperforming other pre-trained models.

Han et al. [18] have proposed an experimental study on semi-supervised method for the classification of COVID-19 and pneumonia via CT scan images. It utilized labeled as well as unlabeled images of CT-scans and attained accuracy of 97.32% with the implementation of semi-supervised method.

Nwosu et al. [32], have deployed semi-supervised learning to omit the need of largely labeled chest x-ray images for the prediction of unlabeled data. The authors of this paper proposed ssResNet model which was developed on the basis of ResNet (Residual Neural Network) architecture incorporated one path for supervised and another path for unsupervised. The results showed significant performance of the model in case of small no. of annotated images for training. EfficientNet is also a pre-trained model employing concept of transfer learning having total of eight versions (B0-B7) [9]. Training the weights of noisy-student dataset is a self training approach which attained 88.4% accuracy when trained with ImageNet [15].

Noisy-student training is the concept of using labeled image data to train the teacher model and perform prediction based on that teacher model to determine pseudo-labels of unlabeled images. Afterwards, it takes large size of student model which is then trained by combining labeled data with noise and the same process is repeated by interchanging student as teacher in the next step [33]. We have pre-trained model of efficient-net on kaggle which is trained on the weights of noisy-student that represents a semi-supervised learning method [34].

COVID-19 is a global pandemic virus whose exact medication has not been invented yet. Thus, this virus is still a threat for humans as it can mutate and re-surface again with more impact on upcoming days. Since the main method of decreasing the spread of coronavirus is timely diagnosis of this disease, professionals need to have access to the properly labeled medical imaging dataset (i.e. Chest x-rays or CT scans). However, this is not possible all the time since in most cases, we may have only small number of COVID-19 labeled data. Thus, based on semi-supervised learning we can combine those available small COVID-19 labeled data with unlabeled ones to classify the presence or absence of coronavirus in human body through chest x-rays/CT-scans. EfficientNet is a pre-trained model and with the implementation of noisy-student weights that supports semi-supervised learning approach, we can employ it for determining the covid positive or negative cases through CT-scan dataset.

Chapter 3

Methodologies

In this chapter the overall system flow is explained in section 3.1 followed by the details of proposed models in section 3.2. For the experimental purpose, we decided to implement pre-trained deep learning models such as DenseNet-201 [7] [35] and ResNet-50 [8] [36]. Moreover, we also proposed two CNN models with distinct number of blocks to evaluate the performance of these models for COVID identification with CT-scans. We further used only a limited no. of labeled data to explore on semi-supervised learning with the use of pre-trained EfficientNet-B4 model using weights of noisy-student and also inspected the same data with one supervised model i.e. DenseNet201.

3.1 System Flow

This section elaborates the overall system flow for the diagnosis of COVID-19 using CT-scan images. Fig. 3.1 illustrates the proposed system flow. We began with the pre-processing of data by performing necessary normalization and train-test split of provided data. We managed the labels as 0 and 1 for COVID and non-COVID patients/cases respectively. Then we prepared five models: DenseNet-201, ResNet-50, two models of CNN and EfficientNet-B4. Afterwards, we trained the respective models with and without augmentation. For image augmentation, we used rotate, shift, flip and zoom features. We performed image augmentation to evaluate the performance of models with increased number of data. We then used the trained

models for predicting labels of the test data and at last, we analyzed the performances of all models to determine the better classifying model.

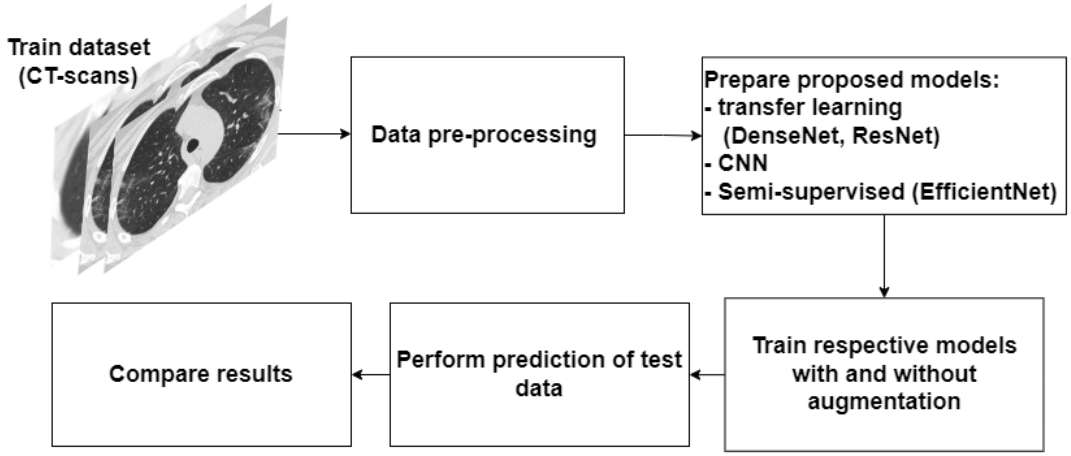


Figure 3.1: Proposed system flow.

3.2 Proposed Models

3.2.1 DenseNet201

DenseNet-201 [7] [35] is a pre-trained deep learning supervised model which has depth of 201 layers. It has continuous dense blocks containing convolution and pooling layer. After each dense block there is a transition layer. Then the model ends to a output layer providing classification of the relative problem. Due to its dense connection of neurons and being trained on ImageNet data [24] for image classification, many researchers use this model for COVID-19 detection through chest x-ray or CT-scan images.

In Fig. 3.2 a detailed layers of densenet model along with the corresponding shapes is shown. We utilized the pre-trained DenseNet-201 model [7] [35] with the weights of ImageNet. We cut-off the existing final layer of this pre-trained model and fine-tuned it with new layers of global average pooling, batch normalization and additional dense layers. The model takes input image of the shape $224 \times 224 \times 3$ while the final output layer provides result with 2 neurons for COVID and non-COVID cases.

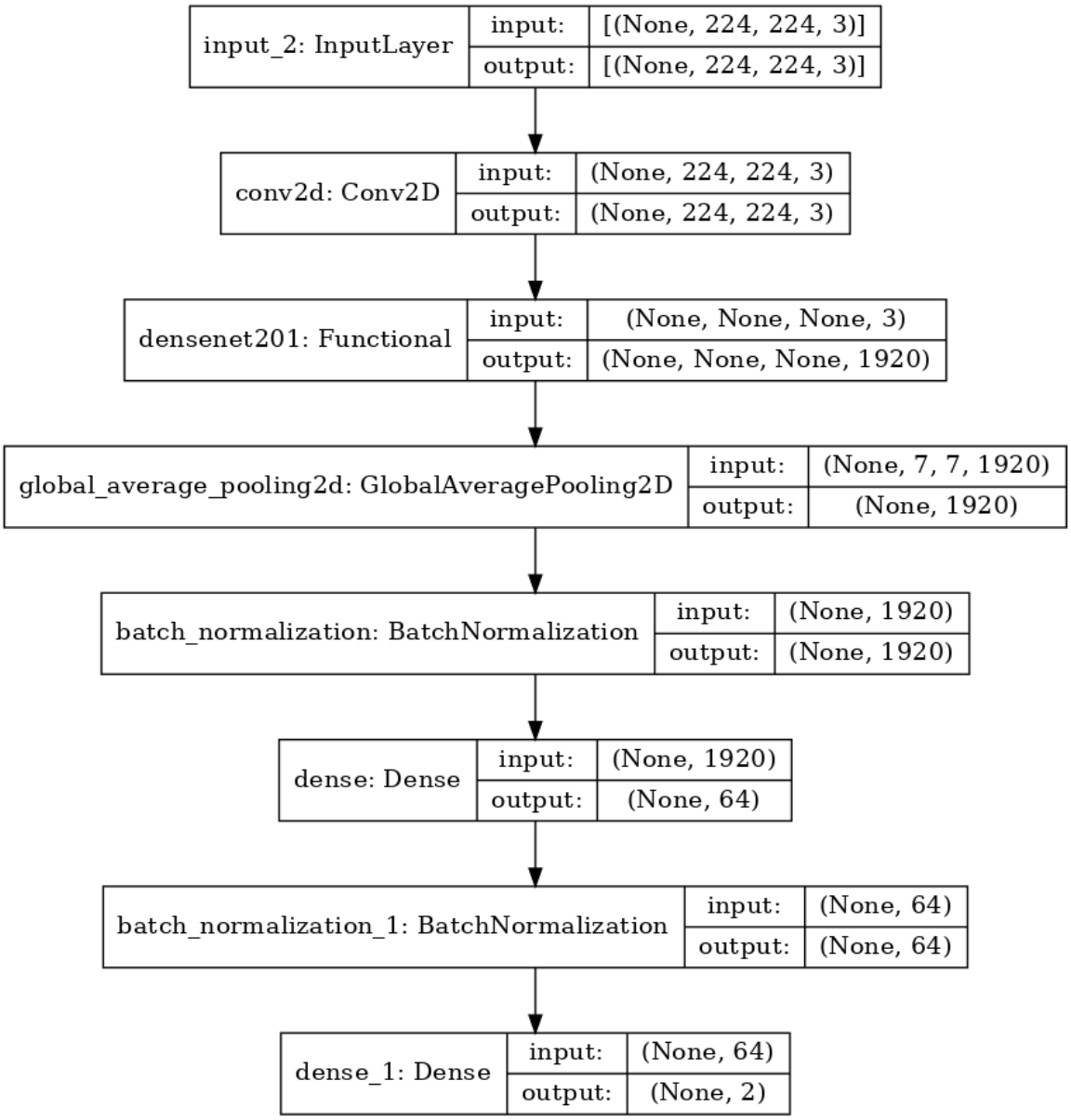


Figure 3.2: Proposed DenseNet model using pre-trained DenseNet-201 with fine-tuned layers.

In this model, we used three number of convolutional filters of the size 3×3 by managing the same padding. We employed this filter in the pre-trained weights attained from DenseNet-201. Then the model was further fine-tuned with global_average_pooling and batchnormalization followed by the dense layer with 64 neurons and 'ReLU' [37]

as activation function. The last layer of this densenet model contained two neurons activated with 'softmax' activation function [38]. Table 3.1 below shows the number of parameters in individual layer. The total no. of trainable parameters in this model are 18,220,054.

Table 3.1: Pre-trained DenseNet-201 model with fine-tuned additional dense layers for the classification of COVID and non-COVID through CT-scans.

Model: DenseNet201

Layer (type)	**Output Shape**	**Param**
Input (InputLayer)	[(None, 224, 224, 3)]	0
conv2d (Conv2D)	(None, 224, 224, 3)]	84
densenet201 (Functional)	(None, None, None, 1920)	18321984
global_average_pooling2d	(None, 1920)	0
batch_normalization	(None, 1920)	7680
dense (Dense)	(None, 64)	122944
batch_normalization_1	(None, 64)	256
dense_1 (Dense)	(None, 2)	130
Total params: 18,453,078		
Trainable params: 18,220,054		
Non-trainable params: 233,024		

3.2.2 ResNet50

ResNet-50 [8] [36] is a CNN model incorporating depth of 50 layers among which 48 are convolutional whereas other two are max-pooling and average-pooling layers respectively. It utilizes the residues from previous layer due to the residual block stacks. Since this model is also pre-trained with ImageNet [24] to extract features from image data to classify the images, it has been used by researchers to extract the features from medical imaging such as CT-scans for diagnosing COVID-19.

Fig. 3.3 depicts the functional plot of the proposed ResNet50 model for this study. It shows the functional layers with their corresponding output shape to process the

input image. We employed already available pre-trained ResNet-50 model [8] [36] incorporating the ImageNet weights. As a part of transfer learning, we excluded the top layer from the existing model of ResNet-50 and added new dense layers after the final layer.

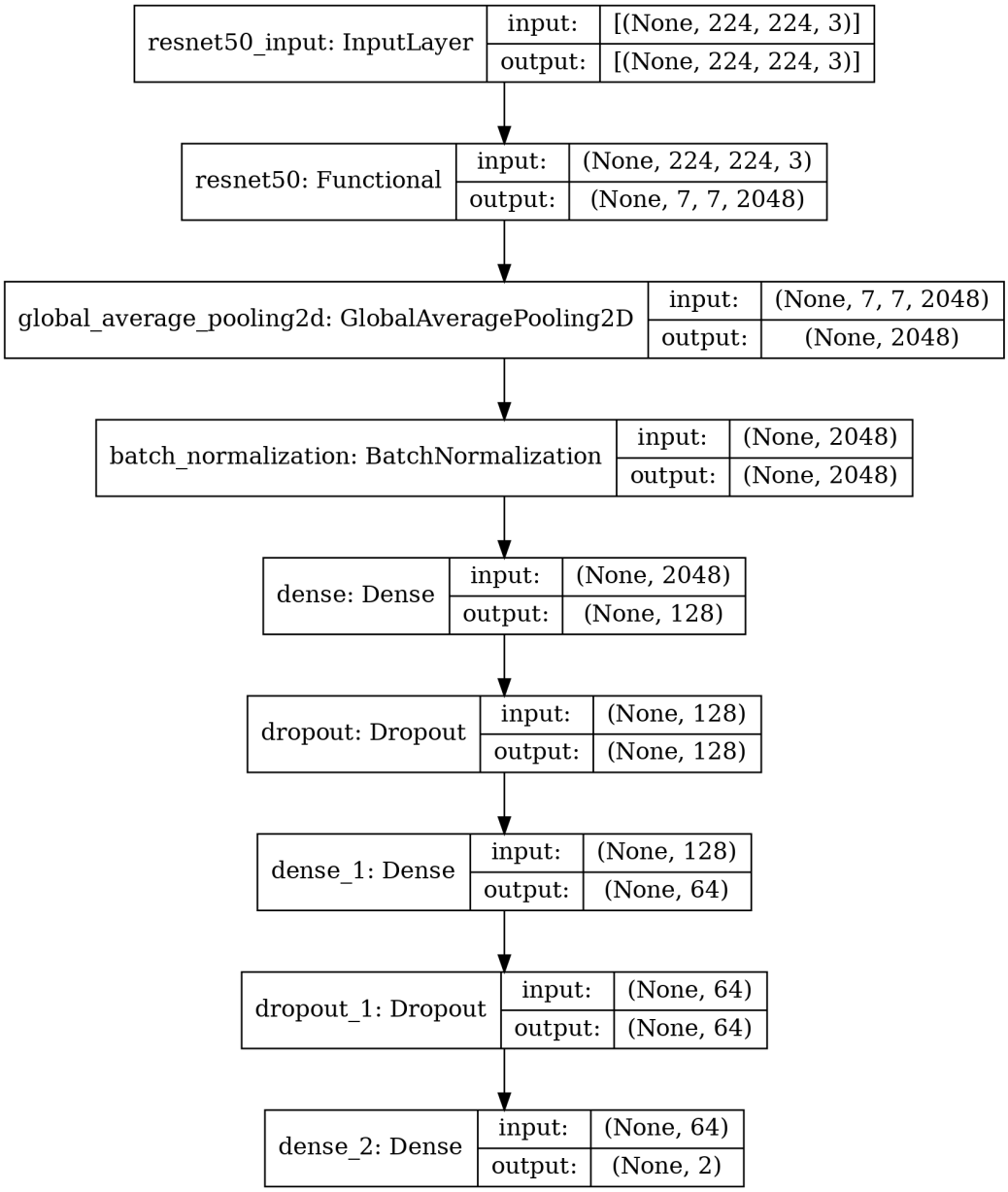


Figure 3.3: Proposed ResNet model using pre-trained ResNet-50 with fine-tuned layers.

The model summary for the proposed ResNet model is shown in table 3.2. In this model we freeze the pre-trained layers and implement three additional dense layers with 128, 64 and 2 no. of neurons simultaneously. After the dense layers with 128 and 64 neurons, we used 0.2 dropout to deal with overfitting issues. This model contained 274,754 trainable and 23,591,808 non-trainable parameters.

Table 3.2: Pre-trained ResNet-50 with fine-tuned additional dense layers for the classification of COVID and non-COVID through CT-scans.

Model: ResNet50

Layer (type)	**Output Shape**	**Param**
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
global_average_pooling2d	(None, 2048)	0
batch_normalization	(None, 2048)	8192
dense (Dense)	(None, 128)	262272
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130
Total params: 23,866,562		
Trainable params: 274,754		
Non-trainable params: 23,591,808		

3.2.3 CNN-model_1

Convolutional neural network is widely popular for image classification and thus, it has been mostly used and explored technique for COVID diagnosis by utilizing the image dataset of patients for instance chest x-rays and CT-scans. With the image feature extraction technology by going through different no. of blocks and layers containing convolutional and pooling layers, CNN model is capable of identifying certain pattern in the input images. CNN model_1 is proposed to perform the performance investigation of normal deep learning model with some of the available pre-trained models such as DenseNet-201 and ResNet-50 for the detection of COVID-19 via CT-

scan images. Table 3.3 depicts the summary of proposed CNN1 (CNN_model_1) by showing each layer with its corresponding output shape and parameters.

Table 3.3: Model summary of CNN_model_1 showing the layers and parameters.

Model: CNN_model_1

Layer (type)	**Output Shape**	**Param**
input (InputLayer)	[(None, 224, 224, 3)]	0
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
maxpool2d_1 (MaxPooling2D)	(None, 112, 112, 64)	0
dropout_1 (Dropout)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 96)	55392
maxpool2d_2 (MaxPooling2D)	(None, 56, 56, 96)	0
dropout_2 (Dropout)	(None, 56, 56, 96)	0
conv2d_3 (Conv2D)	(None, 56, 56, 128)	110720
maxpool2d_3 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_3 (Dropout)	(None, 28, 28, 128)	0
conv2d_4 (Conv2D)	(None, 28, 28, 160)	184480
maxpool2d_4 (MaxPooling2D)	(None, 14, 14, 160)	0
dropout_4 (Dropout)	(None, 14, 14, 160)	0
conv2d_5 (Conv2D)	(None, 14, 14, 192)	276672
maxpool2d_5 (MaxPooling2D)	(None, 7, 7, 192)	0
dropout_5 (Dropout)	(None, 7, 7, 192)	0
global_average_pooling2d	(None, 192)	0
output (Dense)	(None, 2)	386
Total params: 629,442		
Trainable params: 629,442		
Non-trainable params: 0		

The first proposed model of CNN for this work incorporates five no. of blocks. Each block is designed with convolution layer having distinct no. of filters of kernel size

3 and padding as 'same'. The convolution layer is followed by max-pooling function of size 2 to provide down-sampled output and a dropout layer 0.1 dropout size. The five blocks contained 64, 96, 128, 160 and 192 filters continuously from first to the the fifth block. We modeled the final block with global-average pooling and a dense layer of 2 units. This model has total 629,442 trainable parameters.

3.2.4 CNN-model_2

As CNN is a basic model for all other deep learning models, it is mostly applicable to classify simple image based problem. We can improvise the CNN model by introducing more or less layers and/or kernels to study its impact on identification of COVID-19 using CT-scans.

The second model of CNN i.e. CNN_model_2 is deployed with similar block structure as in CNN_model_1 but with seven no. of blocks. CNN_model_2 starts with input layer accepting $224 \times 224 \times 3$ images with seven blocks containing convolution, max-pooling and dropout layers. The filter used in each block is of the kernel size 3. However, the no. of filter varied from 1^{st} to 7^{th} blocks as 64, 96, 128, 160, 192, 224 and 256 respectively. The total parameters available in this model are 1,533,218 and none of them are non-trainable.

Table 3.4: Model summary of CNN_model_2 showing the layers and parameters.

Model: CNN_model_2

Layer (type)	**Output Shape**	**Param**
input (InputLayer)	[(None, 224, 224, 3)]	0
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
maxpool2d_1 (MaxPooling2D)	(None, 112, 112, 64)	0
dropout_1 (Dropout)	(None, 112, 112, 64)	0
conv2d_2 (Conv2D)	(None, 112, 112, 96)	55392
maxpool2d_2 (MaxPooling2D)	(None, 56, 56, 96)	0
dropout_2 (Dropout)	(None, 56, 56, 96)	0
conv2d_3 (Conv2D)	(None, 56, 56, 128)	110720
maxpool2d_3 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_3 (Dropout)	(None, 28, 28, 128)	0
conv2d_4 (Conv2D)	(None, 28, 28, 160)	184480
maxpool2d_4 (MaxPooling2D)	(None, 14, 14, 160)	0
dropout_4 (Dropout)	(None, 14, 14, 160)	0
conv2d_5 (Conv2D)	(None, 14, 14, 192)	276672
maxpool2d_5 (MaxPooling2D)	(None, 7, 7, 192)	0
dropout_5 (Dropout)	(None, 7, 7, 192)	0
conv2d_6 (Conv2D)	(None, 7, 7, 224)	387296
maxpool2d_6 (MaxPooling2D)	(None, 3, 3, 224)	0
dropout_6 (Dropout)	(None, 3, 3, 224)	0
conv2d_7 (Conv2D)	(None, 3, 3, 256)	516352
maxpool2d_7 (MaxPooling2D)	None, 1, 1, 256)	0
dropout_7 (Dropout)	(None, 1, 1, 256)	0
global_average_pooling2d	(None, 256)	0
output (Dense)	(None, 2)	514
Total params: 1,533,218		
Trainable params: 1,533,218		
Non-trainable params: 0		

3.3 Semi-supervised Learning

In real world, it is difficult to attain perfectly annotated data. Moreover, it would be more difficult to annotate the data in medical field due to the variation in detected symptoms of a particular disease in different patient. Thus, there is a high chance that the real-world data of CT-scan images are un-labelled provided that only a limited data are annotated. In this section, we study the effects of semi-supervised learning for covid identification through CT-scans. When we have less no. of labeled images for covid cases, the supervised learning concept is not suitable to determine the labels of large no. of unlabeled images. Thus, to investigate on this scenario, we extended our study towards semi-supervised learning for which we only took 600 no. of data in total from the original dataset. We performed the train-test-validation split in the same ratio as for the other models but we separated the labeled and unlabeled data in the no. of 60 and 420 respectively. The remaining data were used for validation and testing. Since this is an extended topic of this thesis, we explored the semi-supervised model explained in the section 3.3.1 along with DenseNet201 model designed in the section 3.2.1 for the total of 600 samples having only 60 annotated samples. This allowed us to determine the performance of supervised and semi-supervised learning techniques when only few samples are annotated.

3.3.1 EfficientNet-B4 for Semi-supervised Learning

EfficientNet-B4 is based on CNN model architecture which can be found as pre-trained model for image classification on Kaggle. The model of EfficientNet-B4 is fine-tuned for the two class classification of COVID and non-COVID scenario for this study using the weights of 'noisy-student'. Noisy-student incorporates the concept of semi-supervised learning by introducing noise to the student class. This model supports the training of teacher model using limited no. of labeled images and generates pseudo-labels for unlabeled images. Upon attaining the pseudo-labels, they are then combined with labeled ones and trained on student model. The algorithm followed to implement semi-supervised learning is shown in following steps:

Step 1: Load and pre-process labeled and unlabeled data
Step 2: Train supervised model on labeled data
Step 3: Use supervised model to predict labels of un-labeled data

Step 4: Combine labeled and pseudo-labeled data
Step 5: Train semi-supervised model on combined data
Step 6: Perform prediction with semi-supervised model
Step 7: Evaluate the model on test data

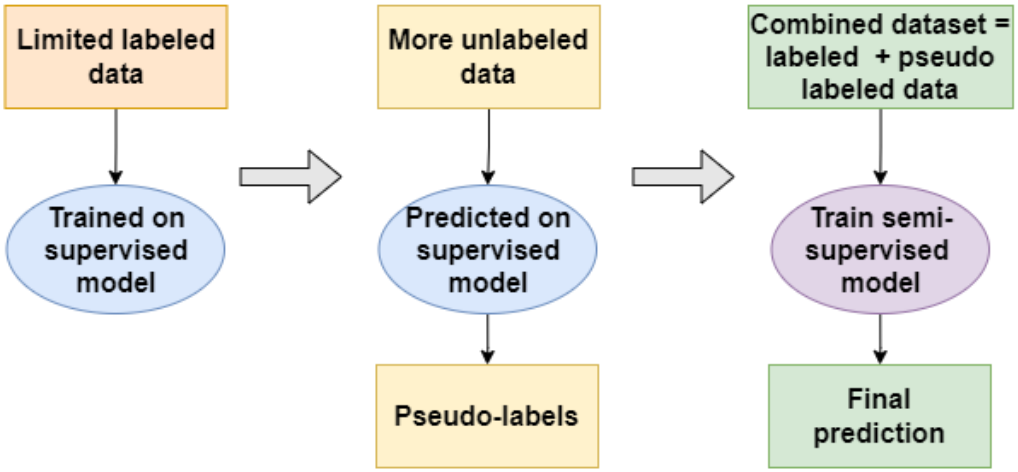


Figure 3.4: Basic flow of semi-supervised learning approach.

Fig. 3.4 depicts the pictorial representation of above explained algorithm for semi-supervised learning. For determining the pseudo-labels of unlabeled data, We used DenseNet model as explained in section 3.2.1. The pre-trained DenseNet201 model is first utilized for the training of labeled images and it determined the labels for unlabeled images. Then we combined the pseudo-labels with labeled data to form a new dataset which is then trained on semi-supervised model (EfficientNet-B4 using noisy-student weights) as shown in fig. 3.5. The details of functional layers and their corresponding parameters for semi-supervised model are shown in the table 3.5. This model has 17,552,202 trainable parameters along with 125,207 non-trainable parameters.

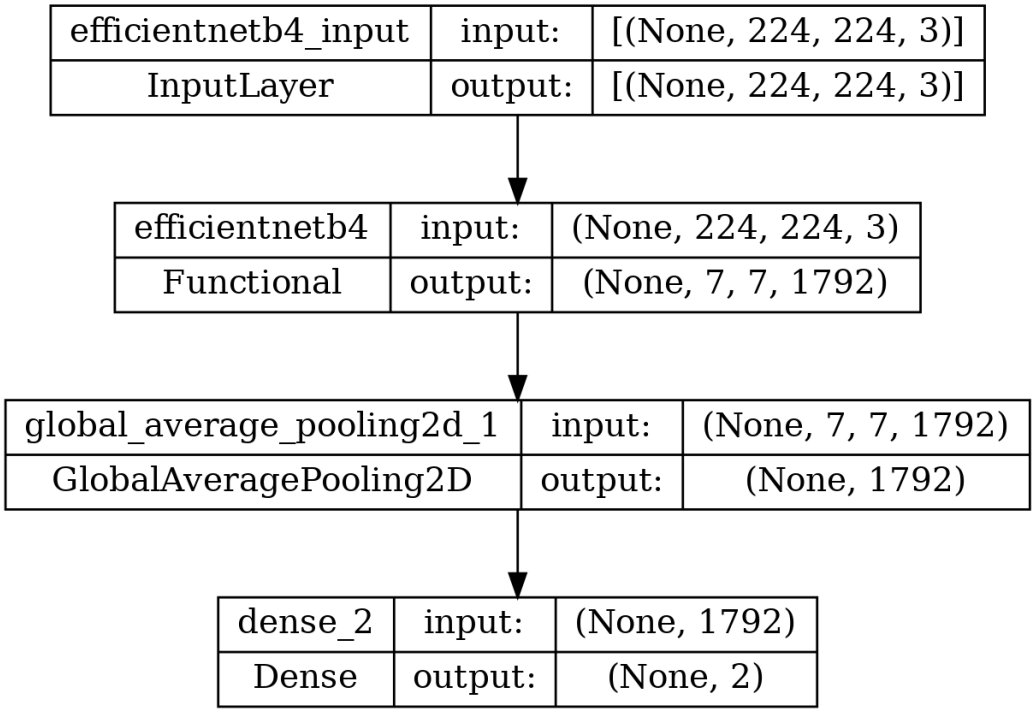


Figure 3.5: Proposed EfficientNet model using pre-trained EfficientNet-B4 with noisy-student weights.

Table 3.5: Model summary of EfficientNetB4 with semi-supervised learning showing the layers and parameters.

Model: EfficientNet-B4

Layer (type)	**Output Shape**	**Param**
efficientnetb4 (Functional)	(None, 7, 7, 1792)	17673823
global_average_pooling2d_1	(None, 1792)	0
dense_2 (Dense)	(None, 2)	3586
Total params: 17,677,409		
Trainable params: 17,552,202		
Non-trainable params: 125,207		

Chapter 4

Results

The description of the dataset used for this study is provided in the section 4.1. The dataset detail is then followed by the description of evaluation metrics in section 4.2 that were used for the result analysis of deep learning models employed in this work.

The results are explained in multiple sections such as results for the supervised models; DenseNet201, ResNet50, CNN_model_1 and CNN_model_2 are incorporated in the section 4.4 and section 4.3 for with and without image augmentation respectively. We used the GPU in Kaggle for the training and prediction of proposed models. The supervised models were trained on the full dataset containing 2481 samples of CT-scans. For the experiments described in the sections 4.3 and 4.4, we determined the common parameters of optimization and compilation. We applied Adam learning rate [39] as an optimizer ranging from 0.001 to 0.0001. The varying learning rate aided in the convergence towards maximum accuracy by reducing the model loss over 65 no. of epochs with batch size of 32. We compiled these models by determining the loss as 'categorical-crossentropy' as our problem is defined as two-class classification. For image augmentation, we generated images with 360°rotation, zoom-in, shifts (width and height) by the range of 0.3 and flips (horizontal and vertical). We utilized these data generator characteristics to fit the training data.

Moreover, the section 4.5 determines the results for the exploration of supervised and semi-supervised learning conducted on 600 samples from the original dataset. In this scenario, we implemented 0.001 learning rate of AdaMax [40] and ran the model till 100 epochs. Besides this, we kept all other parameters same as in other

models explained earlier. The section 4.5 covers the outcomes attained for densenet and efficientnet models incorporating the concept of supervised and semi-supervised learning simultaneously.

4.1 Dataset

The data used for this thesis is SARS-COV-2 Ct-Scan dataset provided by a hospital located at Sao Paulo, Brazil [41]. This dataset is used from kaggle [42]. The table 4.1 exhibits the total number of CT-scan images for each of the COVID and non-COVID label available in the dataset. The dataset has 1252 COVID and 1229 non-COVID CT-scan images incorporating in total 2481 samples for the experiment.

Table 4.1: CT-scans dataset from Kaggle for COVID-19 detection.

CT-scans	No. of Images
COVID	1252
non-COVID	1229

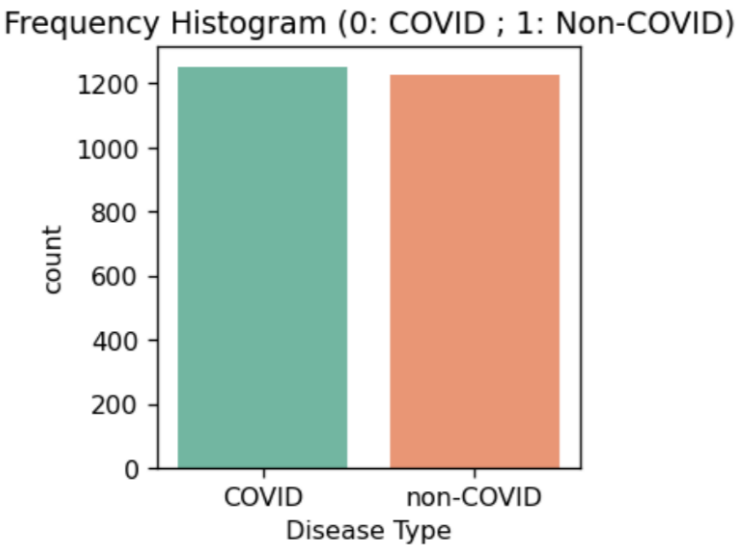


Figure 4.1: Frequency graph representing data counts for each label (Green and orange bar represents COVID and non-COVID respectively).

Fig. 4.1 represents the bar chart showing the frequency bar for COVID and non-COVID cases in the dataset. From this graph, it is clear that the dataset is approximately balanced thus, we proceeded with this dataset without pre-processing for data imbalance. This dataset contained two labels namely, COVID and non-COVID making it two-class classification problem. We determined the labels as 0 for COVID and 1 for non-COVID.

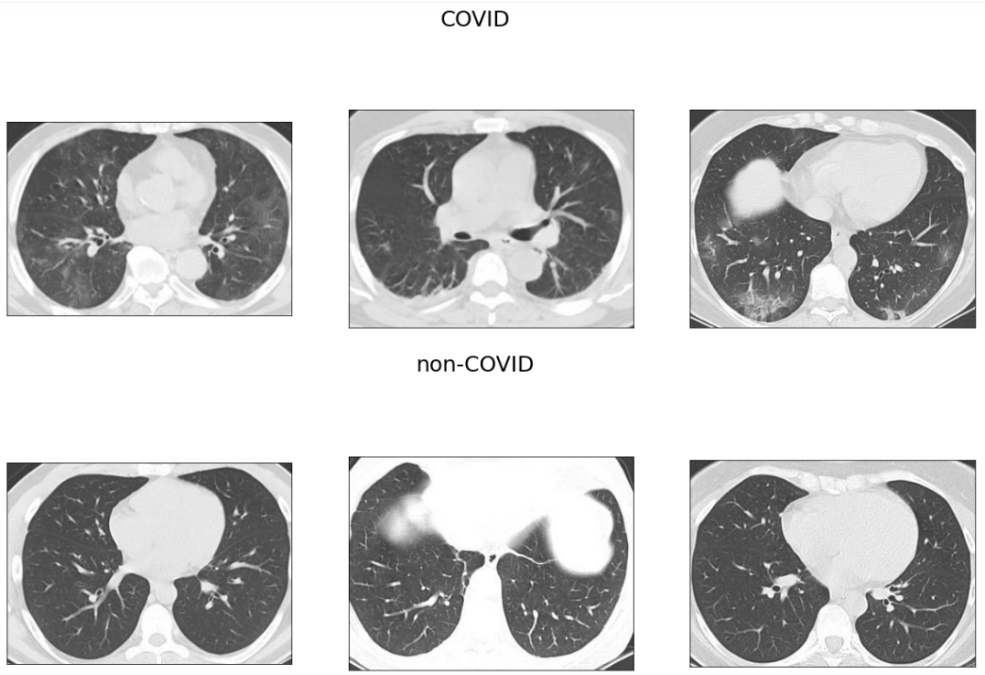


Figure 4.2: Samples of CT-scan images of lungs from COVID and non-COVID patients available in the dataset.

Fig. 4.2 shows three CT-scan samples of lungs from each categories i.e. COVID and non-COVID patients of the above mentioned dataset. For the pre-processing of data, we started by extracting the CT-scan images of total 2481 patients into one dataframe. Then we determined the labels for COVID and non-COVID cases as 0 and 1 respectively. We performed normalization of image pixels by dividing the image input size with 255. Then we split the dataset to train and test data with 80:20 ratio. This means, we split 80% of data from the original dataset for training and remaining 20% for validation. Among the 20% validation data, we separated 10% for validation and 10% to be used for testing as shown in fig. 4.3.

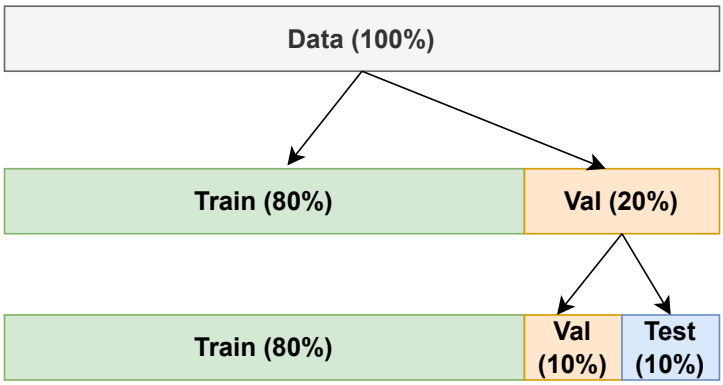


Figure 4.3: Image showing ration of train, test and validation data split during the data pre-processing stage.

4.2 Evaluation Metrics

This section explains the evaluation criteria of the designed models to perform the valuable analysis of model performances for covid detection. The models were analyzed based on the accuracy metric along with recall parameter attained from confusion matrix as explained in section 4.2.1 and section 4.2.2 respectively.

4.2.1 Accuracy

'Accuracy' is the commonly used metric for the evaluation of model's performance. It is a metric to calculate the equality of actual and predicted labels of the test data to determine the accuracy of model's prediction. This evaluation method depends on the prediction and ground truths whose difference determines how accurate the predictions are based on the actual labels.

4.2.2 Confusion matrix

Confusion matrix is a metric to measure the performance of a machine learning model for classification based on the actual and predicted labels. The parameters we attain

from a confusion matrix are: TP, FN, FP and TN. The definitions of confusion matrix parameters based on our study criteria are as follows:

TP: Actual Covid patient predicted as Covid.
FN: Actual Covid patient predicted as non-covid.
FP: Actual non-covid person predicted as Covid.
TN: Actual non-covid person predicted as non-covid.

Table 4.2 represents these parameters by considering the scenario of our problem i.e. detecting COVID and non-COVID labels from the given CT-scan images. Likewise, fig. 4.4 illustrates the confusion matrix based on our problem scenario. The patient having COVID is represented as label 0 and the person with non-COVID is labeled as 1.

Table 4.2: Parameters of confusion matrix.

Actual label	Predicted label	Parameter
0	0	True Positive (TP)
0	1	False Negative (FN)
1	0	False Positive (FP)
1	1	True Negative (TN)

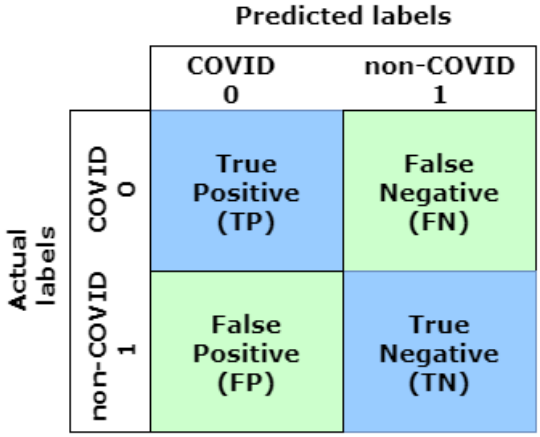


Figure 4.4: General skeleton of confusion matrix for the model evaluation.

Recall is the evaluation criteria which is generally used in case of medical image

classification as it evaluates on the basis of how many actual positives are predicted as positive. Thus, recall is also known as TPR (True Positive Rate) which is determined by the equation 4.1 below:

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

This focuses on increasing True Positives (TP) while reducing False Negatives (FN). False-Negative determines the actually unhealthy person as healthy which can lead to severe health problems for that patient when not diagnosed on time and it also enhances the threats to other people around him if that disease is transmissible like COVID-19. Thus, we take recall into consideration for the evaluation of all of the models performances.

4.3 Results- Supervised Models Without Image Augmentation

This section describes the results attained from DenseNet-201, ResNet-50, CNN_model_1 and CNN_model_2 in the sub-sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4 consecutively. Each model result explained in this section are without the implementation of data augmentation. We had 10% of the dataset separated for the prediction through these supervised models containing 128 actual COVID cases and 121 samples for non-COVID.

4.3.1 DenseNet201

We implemented DenseNet201 [7] [35] as pre-trained deep learning model along with additional dense layer of 64 units followed by the final layer with 2 neurons. From the experiment on DenseNet201 without augmentation, we comprehended that minimum validation loss and maximum validation accuracy was attained at 34^{th} epoch as shown in fig. 4.5. Upon training on 80% data with 10% test and 10% validation data, we attained the accuracy of 99.2% along with 0.0788 loss.

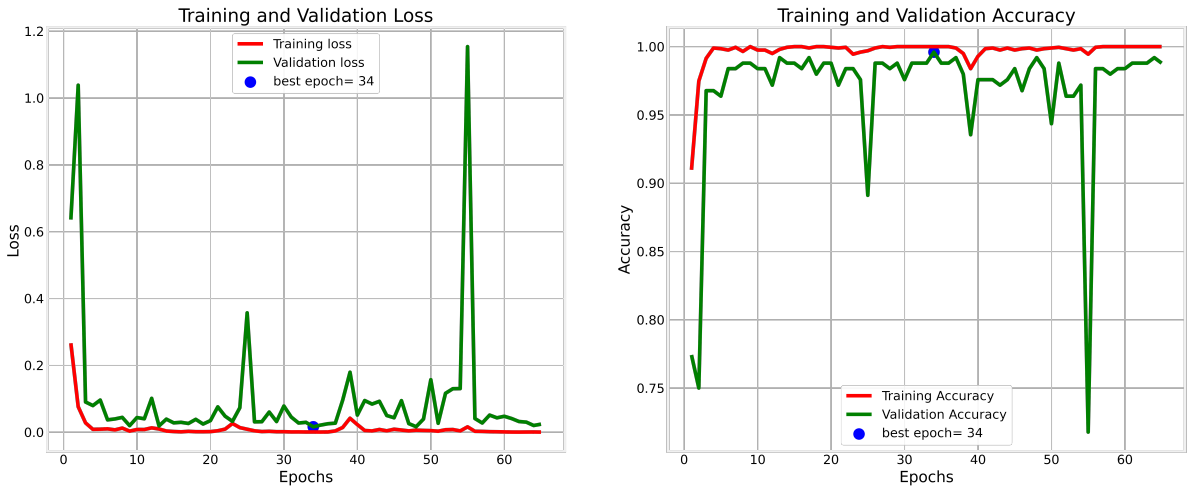


Figure 4.5: Accuracy and loss graph for DenseNet201 without augmentation showing model loss on left side and model accuracy on right side.

Additionally interpreting the performance of this model with the recall metric, we found the true positive rate i.e. recall value as 0.9922. The confusion matrix shown in fig. 4.6 clearly determines that only 1 number of COVID data is wrongly predicted as non-COVID out of 128 samples which increased the recall rate and accuracy of this densenet model.

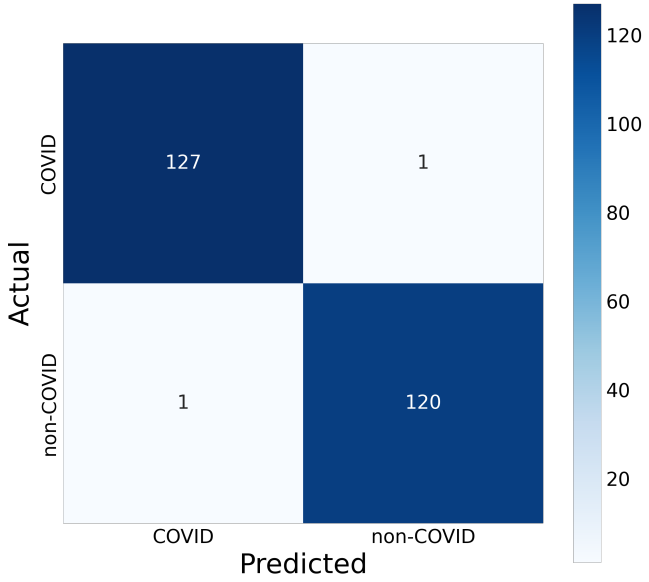


Figure 4.6: Confusion matrix for DenseNet201 without augmentation.

4.3.2 ResNet50

We trained the pre-trained ResNet50 model [8] [36] fine-tuned with two more dense layers having 128 and 64 no. of neurons simultaneously with the training data of CT-scans. We determined the output layer as dense layer with 2 units providing the prediction for COVID and non-COVID. The outcome of this model is shown in fig. 4.7.

From the loss and accuracy graph attained from the training of resnet model in fig. 4.7, we can depict that the best accuracy with lowest validation loss during training is attained at 42nd epoch. The dropout layers added to this model caused some spikes in the loss and accuracy while training. However, observing the left graph of training and validation loss of fig. 4.7, we can see that the validation loss is seemingly increasing after 42nd epoch.

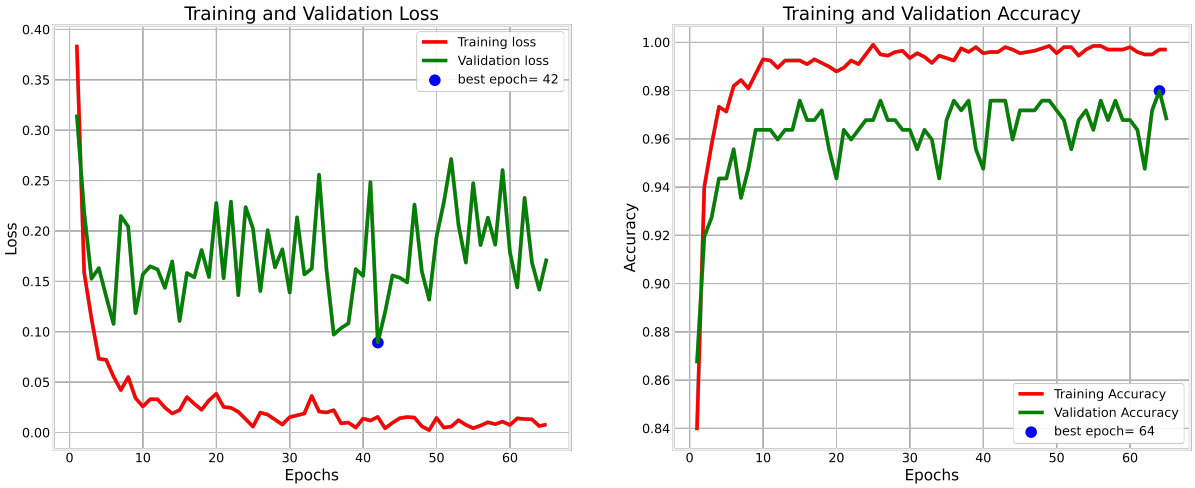


Figure 4.7: Accuracy and loss graph for ResNet50 without augmentation showing model loss on left side and model accuracy on right side.

After predicting the test data using this model, we obtained the model accuracy of 95.58% with 0.3081 model loss. Interpreting the TPR i.e. recall from the confusion matrix presented in fig. 4.8, we found that 6 COVID patients out of 128 were mis-diagnosed as non-COVID which increased the risk of COVID-19 as compared to DenseNet model. Thus, we attained 0.9531 as recall for the ResNet model without augmentation.

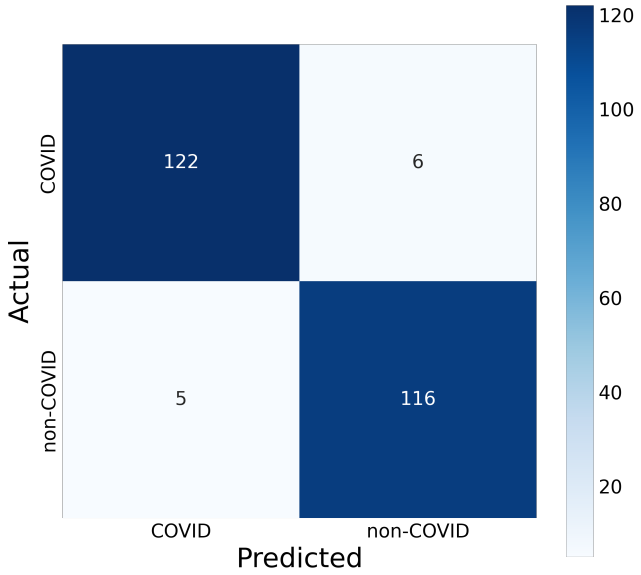


Figure 4.8: Confusion matrix for ResNet50 without augmentation.

4.3.3 CNN_model_1

We first implemented the CNN model having five blocks. Training the data into this model provided the following loss and accuracy graph shown in fig. 4.9 (left side showing model loss and right side representing model accuracy with CNN_model_1). We witnessed the upsurge in validation accuracy with descending pattern in validation loss. From this figure, we observe that best accuracy is attained at 63^{rd} epoch with minimum loss for validation during model training.

Fig. 4.10 illustrates the confusion matrix after predicting the labels for test data. From fig. 4.10 we can interpret that 16 no. of COVID cases from 128 actual COVID samples are mis-classified as non-COVID while using the CNN_model_1 providing 0.875 recall. This model achieved 91.97% accuracy which is far less than DenseNet and ResNet models.

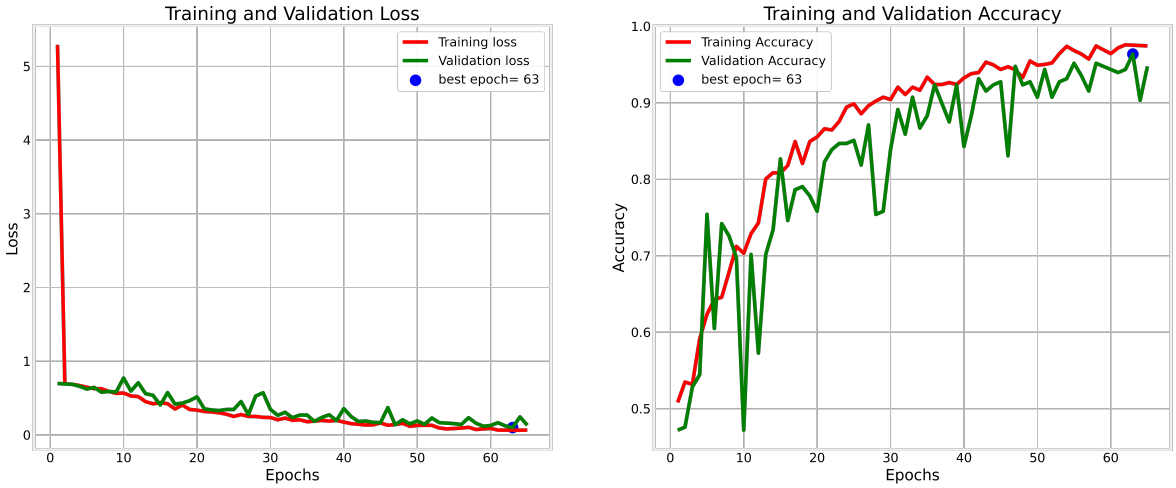


Figure 4.9: Accuracy and loss graph for CNN_model_1 without augmentation showing model loss on left side and model accuracy on right side.

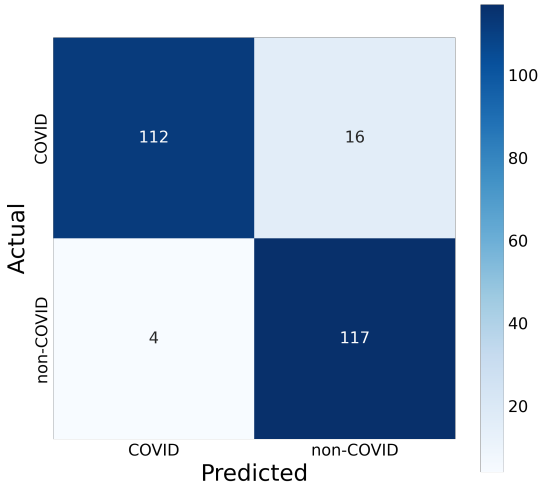


Figure 4.10: Confusion matrix for CNN_model_1 without augmentation.

4.3.4 CNN_model_2

We trained another CNN model i.e. CNN_model_2 having 7 no. of blocks. Each block contained different number of convolutional filters followed by max-pooling layer and a dropout layer. Fig. 4.11 represents the model loss and accuracy graph (left and right part of the graph respectively). From this figure, we discovered that at 52nd

epoch the CNN_model_2 produced least validation loss with highest accuracy while training the model. Upon prediction for labels of test data by using this model we attained 92.77% accuracy resulting the loss of 0.1831.

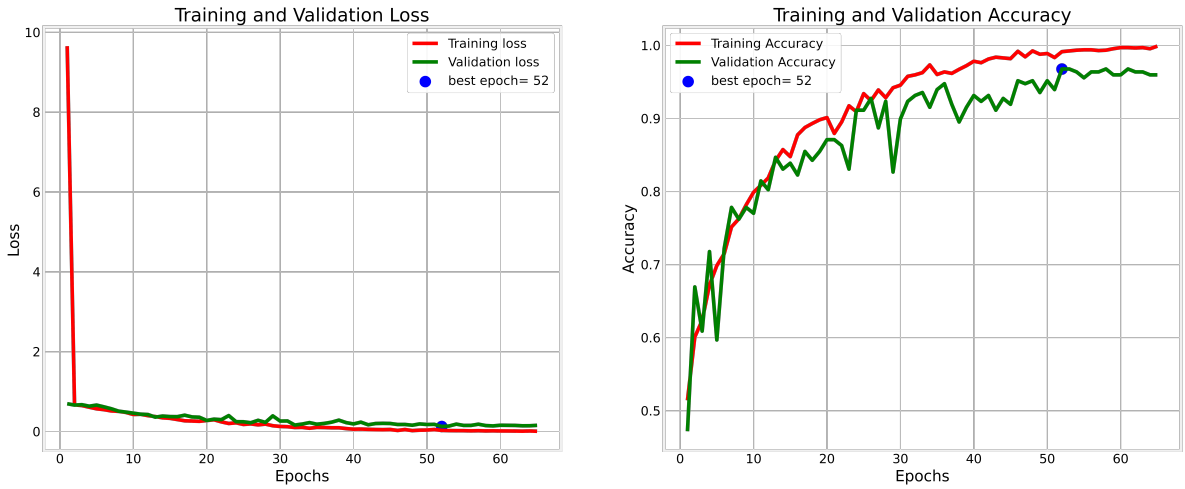


Figure 4.11: Accuracy and loss graph for CNN_model_2 without augmentation showing model loss on left side and model accuracy on right side.

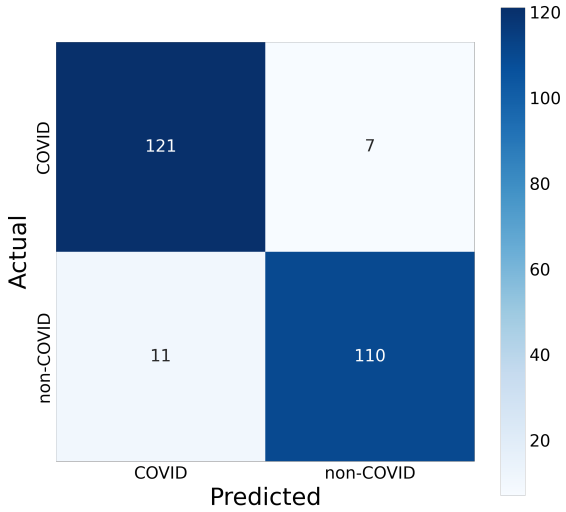


Figure 4.12: Confusion matrix for CNN_model_2 without augmentation.

Moreover, we obtained the confusion matrix representing TPR for CNN_model_2 as shown in fig. 4.12. This figure explains that 7 out of 128 actual COVID cases were

predicted as non-COVID leading the value of recall to be 0.9453. Although we witnessed the recall and accuracy for CNN_model_2 to be higher than CNN_model_1, it is still comparatively less than DenseNet and ResNet without augmentation.

4.4 Results- Supervised Models With Image Augmentation

This section explains the findings of four models i.e. DenseNet201, ResNet50, CNN_model_1 and CNN_model_2 respectively in the sub-sections 4.4.1, 4.4.2, 4.4.3 and 4.4.4. The results elaborated in the mentioned sub-sections are attained from the training and prediction of models by implementing Keras ImageDataGenerator with zoom, shift, flip and rotation to generate the training data.

4.4.1 DenseNet201

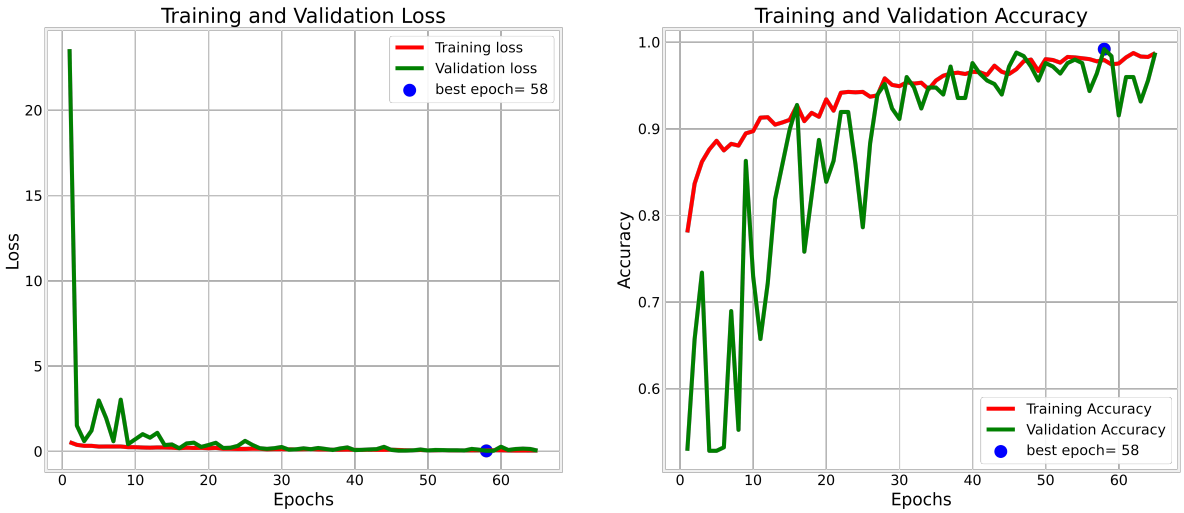


Figure 4.13: Accuracy and loss graph for DenseNet201 with augmentation showing model loss on left side and model accuracy on right side.

Fig. 4.13 demonstrates the graphs for training and validation loss (left graph) and accuracy (right graph) attained through the training of DenseNet model with aug-

mentation. When data augmentation is used on DenseNet-201 model, we acquired the lowest value for validation loss at 58^{th} epoch which incorporated the maximum validation accuracy. From the fig. 4.13, we can depict that model is gradually converging towards the reduced loss over the epochs to reach best accuracy for validation data.

From the prediction of test values using this fine-tuned model of densenet, we found the model accuracy to be 96.39% with 0.1129 loss. The recall for this experiment is determined to be 0.9609 which mis-predicted 5 out of 128 COVID cases as non-COVID as we can see in fig. 4.14. From this observation, we witnessed that DenseNet model with augmentation showed poor performance than the DenseNet model without image augmentation.

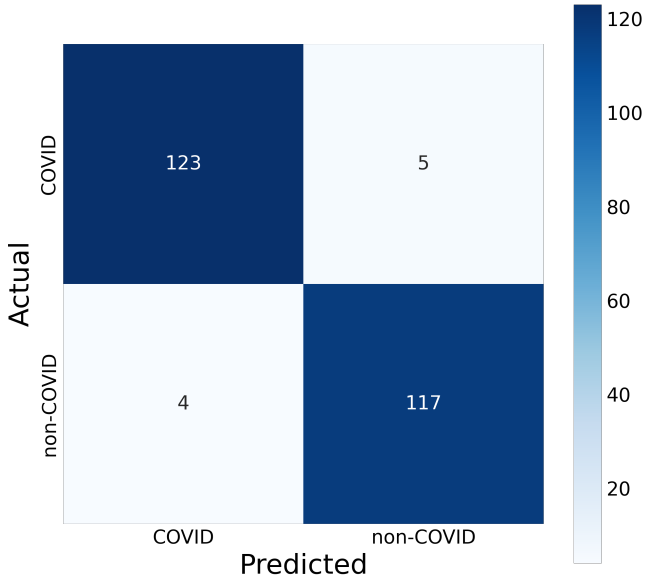


Figure 4.14: Confusion matrix for DenseNet201 with augmentation.

4.4.2 ResNet50

Fig. 4.15 demonstrates the graph of loss and accuracy for training and validation of ResNet50 model using image augmentation. From the left graph of fig. 4.15, we observed that the applied resnet model was able to achieve remarkable validation loss at 59^{th} epoch when we experimented the model with augmentation.

Figure 4.15: Accuracy and loss graph for ResNet50 with augmentation showing model loss on left side and model accuracy on right side.

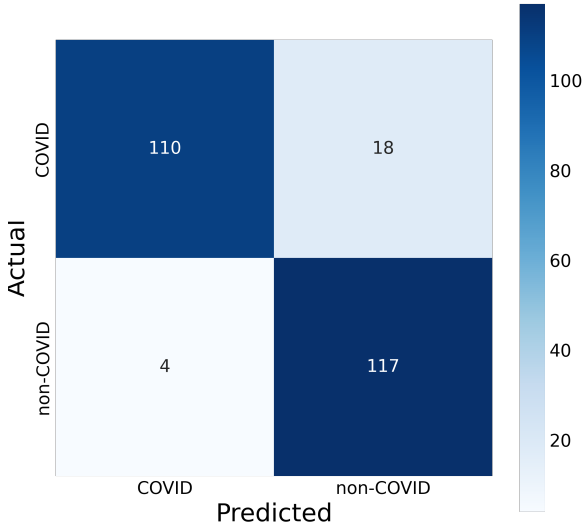


Figure 4.16: Confusion matrix for ResNet50 with augmentation.

We attained the significant result as 91.16% accuracy from the prediction of labels for test data using this model. The accuracy and recall metrics for this model were evaluated based on the confusion matrix shown in fig. 4.16. The observation of fig. 4.16 explains that total 18 COVID patients were mis-diagnosed which can cause health hazard. Thus, we achieved 0.8594 TPR/recall for this model with augmentation which

is significantly weak performance as compared to the performances of ResNet model without augmentation.

4.4.3 CNN-1

Now heading towards the experimental result analysis for the first CNN model which has 5 blocks, we attained the loss and accuracy graph from its training with augmentation as shown in fig. 4.17. The left and right sides of this graph represents the training and validation loss and accuracy respectively for the model. From fig. 4.17, it is clear that after 20^{th}, the validation loss is simultaneously increasing which resulted in the overfitting of model after 20^{th} epoch.

Figure 4.17: Accuracy and loss graph for CNN_model_1 with augmentation showing model loss on left side and model accuracy on right side.

Fig. 4.18 represents the confusion matrix acquired after performing predictions with CNN_model_1 including data augmentation. From this figure, we discovered wrong diagnosis of 78 out of 128 COVID cases limiting the model accuracy to 67.87% with high loss of 0.7866. We obtained the evaluation of recall metric as 0.3906 for this experiment. So, CNN_model_1 showed poor performance as compared to the densenet and resnet models with as well as without augmentation.

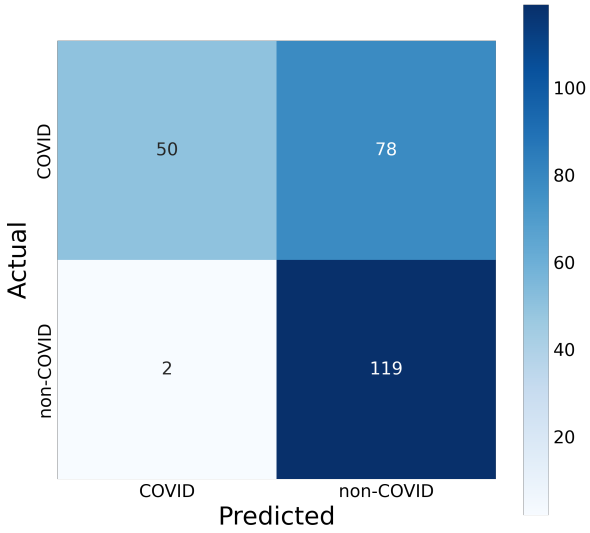


Figure 4.18: Confusion matrix for CNN_model_1 with augmentation.

4.4.4 CNN-2

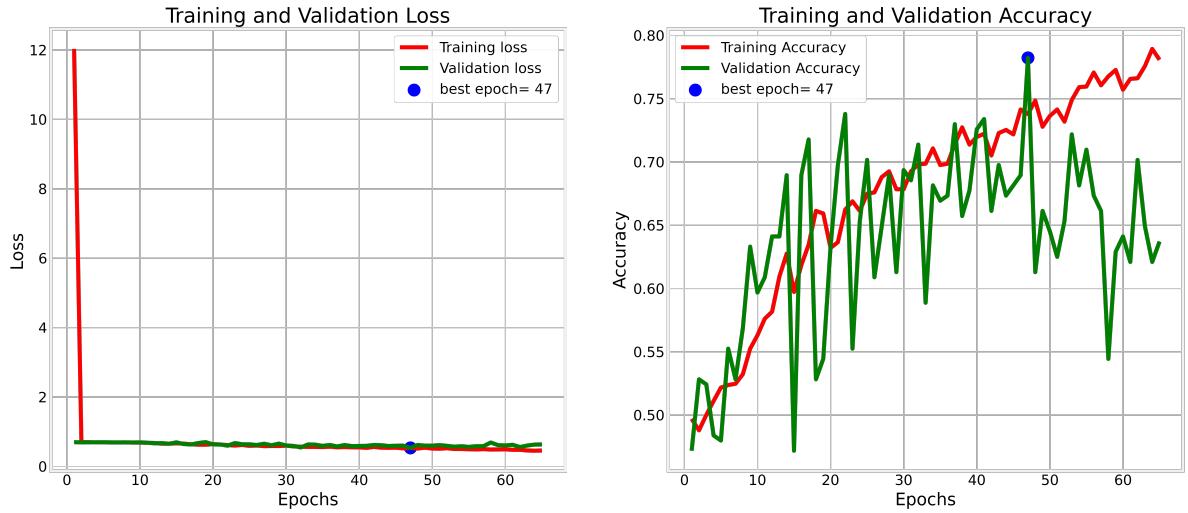


Figure 4.19: Accuracy and loss graph for CNN_model_2 without augmentation showing model loss on left side and model accuracy on right side.

The findings of CNN_model_2 with data augmentation is illustrated by fig. 4.19. The fig. 4.19 displays the loss graph (at the left side) and accuracy graph (at the right

side) for model training and validation. From this graph we can observe that after 47^{th} epoch, the model starts to overfit. Thus, best accuracy with minimum validation loss is attained at 47^{th} epoch for this model.

We also obtained the confusion matrix after predicting the test data labels using CNN_model_2 which is shown in fig. 4.20. This matrix illustrates that 82 no. of covid positive cases were predicted as negative. This is such a high number of misdiagnosis that the accuracy of this model could only reach upto 65.86% providing the loss value of 0.8323 and recall of 0.3594. We observed that this model performance was the lowest among all the models of densenet, resnet, CNN_model_1 and CNN_model_2 with and without augmentation.

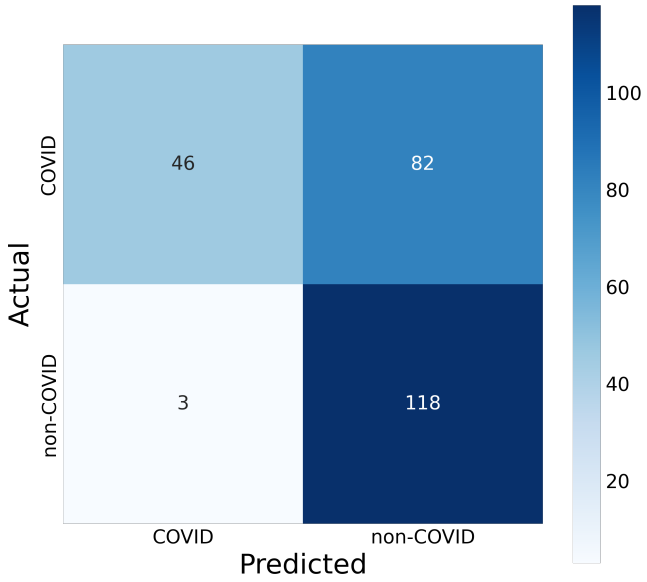


Figure 4.20: Confusion matrix for CNN_model_2 without augmentation.

4.5 Results- Supervised Vs Semi-supervised Learning

This section incorporates the explanations on the outcomes of supervised and semi-supervised learning both with and without data augmentation. In case of supervised learning, we implemented DenseNet-201 model fine-tuned with additional dense lay-

ers whereas for semi-supervised learning, we used EfficientNet-B4 with the weights of 'noisy-student'. For this experiment, we determined the constant value for Adamax learning rate [40] as 0.001 and set the batch size to 32 with 100 no. of epochs. We determined 600 samples from the original dataset to be used for this analysis which contained only 60 labeled samples. And the train-test-validation split is performed on the ratio of 80:10:10. The results without augmentation for supervised and semi-supervised learning are described in the sub-section 4.5.1 and the results with augmentation are elaborated in the sub-section 4.5.2.

4.5.1 Without Augmentation

4.5.1.1 Supervised

We took the model DenseNet201 (section 3.2.1) to be used for supervised learning as it provided the best performance among all of the supervised learning models implemented in this work. Fig. 4.21 shows the graph for training and validation accuracy and loss (right and left side of the graph respectively) for the supervised model.

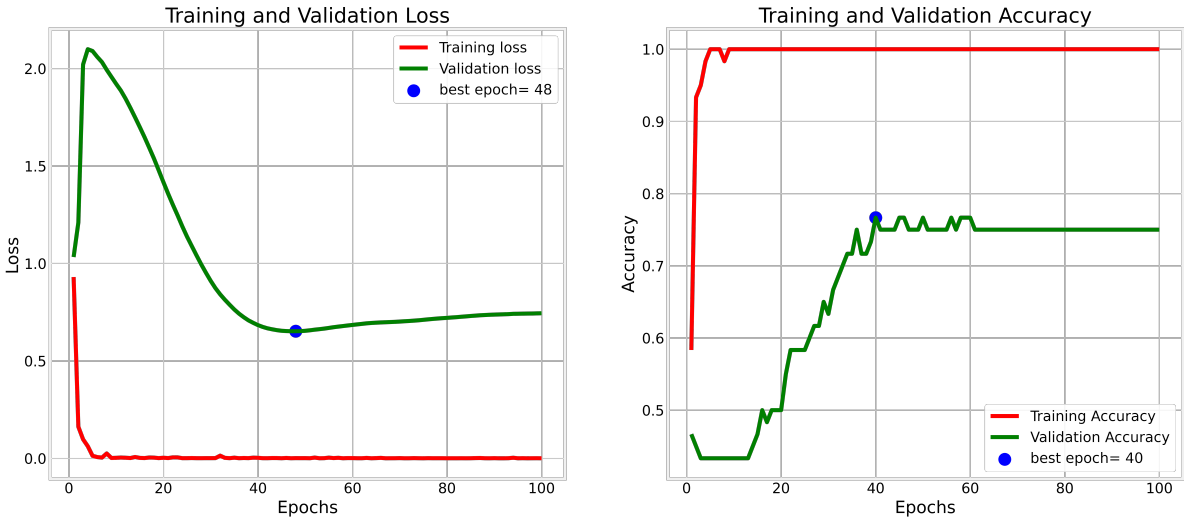


Figure 4.21: Accuracy and loss graph for supervised model without augmentation showing model loss on left side and model accuracy on right side.

From fig. 4.21, we observed over-fitting of the training and validation data as a result

of less number of samples being used in this scenario. As this model ran without augmentation for 100 epochs, we achieved the lowest validation loss at 48^{th} epoch.

We encountered the final accuracy of this supervised model without augmentation as 75% providing the loss value to be 0.9087. Fig. 4.22 determines the confusion matrix attained after the prediction on supervised model without augmentation. From this figure, it is clear that we have total 25 COVID and 35 non-COVID cases to be predicted in this scenario. We observed that among the 25 no. of COVID patients, 4 were wrongly diagnosed and the model achieved the recall value of 0.84.

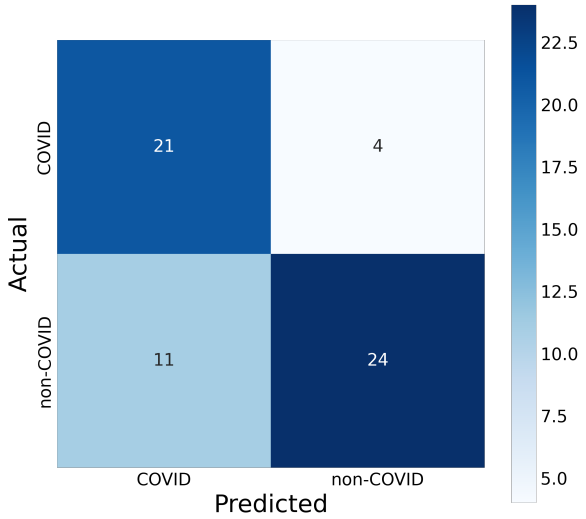


Figure 4.22: Confusion matrix for supervised model without augmentation.

4.5.1.2 Semi-supervised

For the analysis of supervised and semi-supervised learning over limited annotated data, we implemented pre-trained model of EfficientNet-B4 (section 3.3.1) with noisy-student weights as semi-supervised model. Fig. 4.23 illustrates the training result of this semi-supervised approach without data augmentation representing the training vs validation loss and accuracy at left and right graphs consecutively. Since we were using a small no. of data in this exploration, we noticed overfitting between train and validation accuracy. This graph explains that after 27^{th} epoch, the model suffered more over-fitting when trained for 100 epochs.

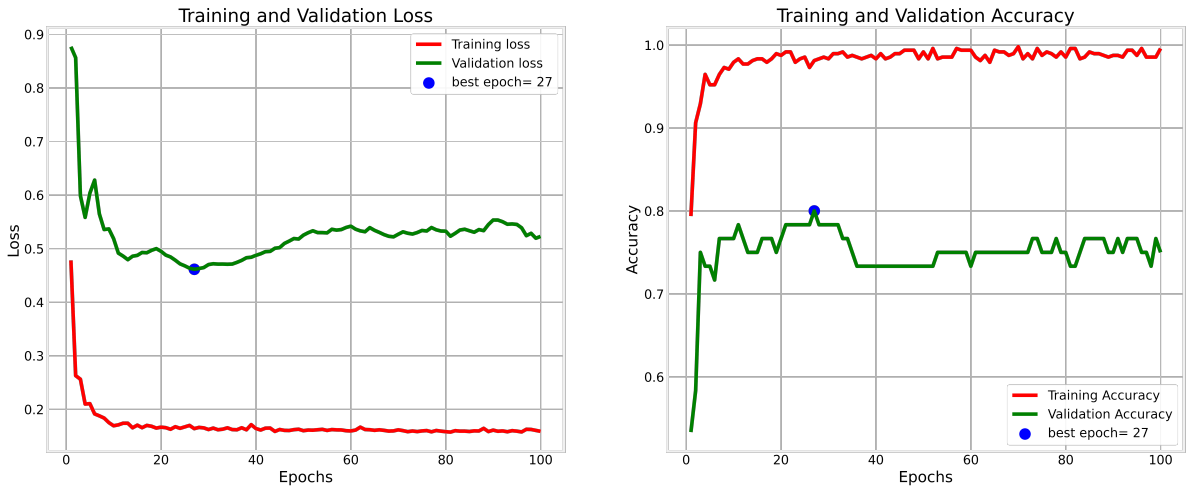


Figure 4.23: Accuracy and loss graph for semi-supervised model without augmentation showing model loss on left side and model accuracy on right side.

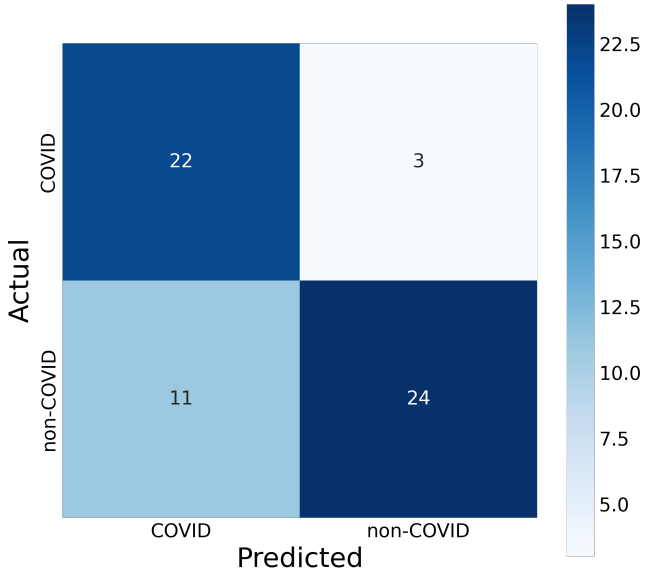


Figure 4.24: Confusion matrix for semi-supervised model without augmentation.

Likewise, from the confusion matrix shown in the fig. 4.24 we can depict that 3 no. of actual COVID cases were mis-classified as non-predicted among the total of 25 COVID test data. We found the accuracy of this semi-supervised model without augmentation to be 76.67% with 0.88 recall and 0.5578 model loss. This observation

showed that semi-supervised model gave significantly better performance than the supervised model.

4.5.2 With Augmentation

4.5.2.1 Supervised

We utilized the same supervised learning model but this time with data augmentation to generate more CT-scan images for training. Now with augmentation, we found that the minimum loss was encountered at 3rd epoch while the best accuracy for validation was witnessed at the last epoch. Over the 100 epochs, the model continuously improved its performance to reach towards the convergences of training and validation data.

Figure 4.25: Accuracy and loss graph for supervised model with augmentation showing model loss on left side and model accuracy on right side.

Fig. 4.26 represents the resulted confusion matrix of prediction from supervised learning model with the implementation of augmentation. From this figure, we can explain that out of 25 actual COVID cases, 7 were wrongly classified as non-COVID providing the model accuracy of 80%. We attained the model loss as 0.8191 with TPR of 0.72 for this model.

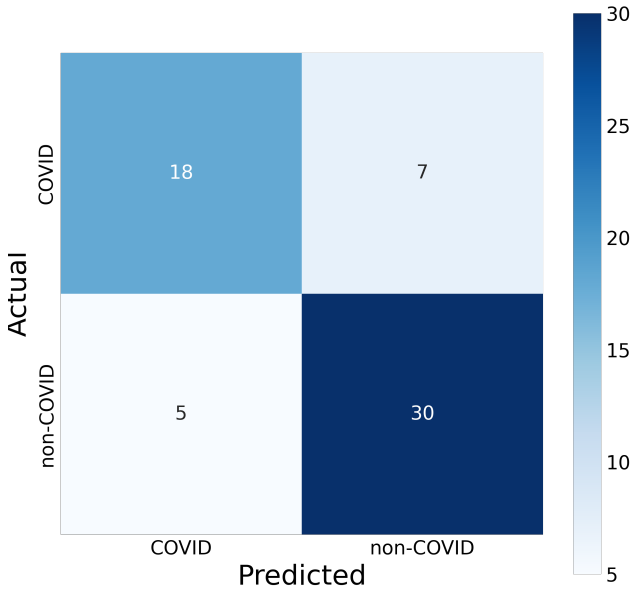


Figure 4.26: Confusion matrix for supervised model with augmentation.

4.5.2.2 Semi-supervised

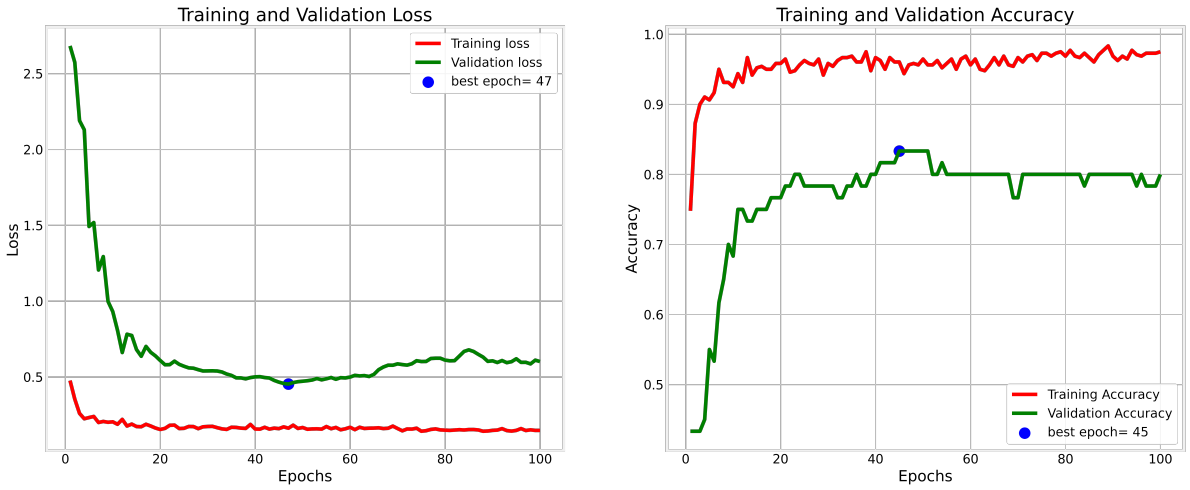


Figure 4.27: Accuracy and loss graph for semi-supervised model with augmentation showing model loss on left side and model accuracy on right side.

Using the efficientnet model with augmentation, we carried out the experiment on semi-supervised learning. From this inspection, we attained the loss and accuracy

graph showing the curves for training and validation data as shown in fig. 4.27. From this figure, we observe that after 47^{th} epoch, the model starts to over-fit as the training and validation curves gradually moves away from each-other.

Fig. 4.28 depicts the confusion matrix resulted from the prediction of test data labels performed using semi-supervised model with augmentation. From the observation of this confusion matrix, we can determine that only 4 COVID cases were mis-predicted as non-COVID among the 25 actual COVID cases. Thus, we attained the accuracy of 83.33% with 0.5379 loss and 0.84 recall for this experiment. Analyzing the performance of this model with supervised model having data augmentation, we observed that the performance of semi-supervised model was better.

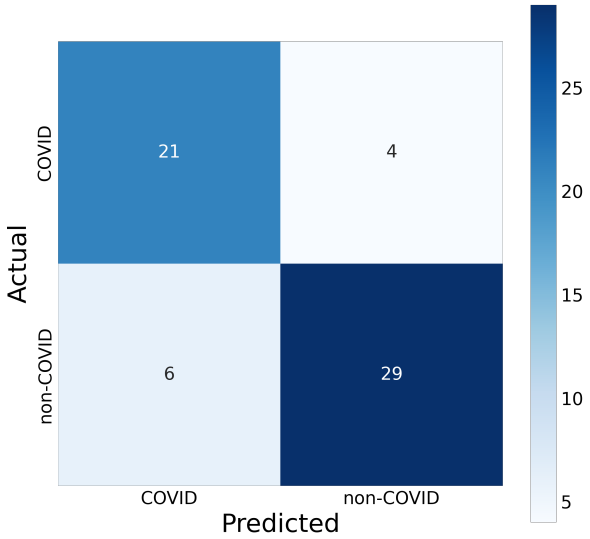


Figure 4.28: Confusion matrix for semi-supervised model with augmentation.

From the analysis of supervised and semi-supervised learning with and without data augmentation in case of limited samples with proper labels, we discovered that semi-supervised model attained significantly higher accuracy than the supervised model. In addition, the mis-classified cases were also less in semi-supervised model for both experiment with and without augmentation. Thus, we determined the semi-supervised learning to be better model than supervised for the identification of covid-19 through CT-scan images.

Chapter 5

Discussion and Further Work

This chapter is about the discussion on attained results from all the experimental deep learning models used for the identification of COVID-19 using CT-scan images. The section 5.1 contains the analysis on obtained outcomes from each model implemented in this study. Moreover, section 5.2 describes the possibility of future works that can be carried out on this topic.

5.1 Discussion

This study explored different deep learning models for the detection of coronavirus using CT-scan images. The exploration of distinct supervised models allowed to determine the model with most precise prediction for the detection of COVID-19 through CT-scans. Moreover, we also investigated on the effectiveness of semi-supervised learning for this scenario. We weighted up the performance of each model with and without implementing data augmentation. The outcomes of all supervised deep learning models implemented in this study to detect coronavirus through CT-scans are tabulated in the table 5.1. This table consists of test results based on accuracy and recall metrics for DenseNet201, ResNet50, CNN_model_1 and CNN_model_2 with and without data augmentation. We used all data (2481 samples) to perform the training and predictions on these supervised model.

Analyzing the results from this table 5.1, we realized that DenseNet-201 without aug-

mentation has the highest accuracy among all of the supervised models achieving 99.2% accuracy. We witnessed that densenet model achieved highest recall of 0.9922 among all the implemented supervised models without augmentation. We also noticed that even when image augmentation was implemented, DenseNet201 showed significantly higher performance than the other supervised models by attaining the accuracy and recall of 96.39% and 0.9609 respectively.

Table 5.1: Loss, accuracy and recall outcomes of DenseNet201, ResNet50, CNN1 and CNN2 using all samples.

Results \ Model		All data (2481 samples)			
		DenseNet-201	ResNet-50	CNN_mode_1 (5-blocks)	CNN_model_2 (7-blocks)
Without augmentation	Test loss	0.0788	0.3081	0.273	0.1831
	Test accuracy (%)	0.992	0.9558	0.9197	0.9277
	Recall	0.9922	0.9531	0.875	0.9453
With augmentation	Test loss	0.1129	0.301	0.7866	0.8323
	Test accuracy (%)	0.9639	0.9116	0.6787	0.6586
	Recall	0.9609	0.8594	0.3906	0.3594

We discovered that ResNet-50 was the second best performing model to identify COVID-19 from CT-scan images after densenet by attaining 95.58% accuracy and 0.9531 recall without augmentation. With augmentation, these accuracy and recall values dropped to 91.16% and 0.8594 consecutively however, resnet still managed to be the better model than CNN_model_1 and CNN_model_2. Moreover, from the table 5.1, we depicted the performance of CNN_model_2 containing 7 no. of blocks surpassed CNN_model_1 having 5 no. of blocks when predicted without augmentation. CNN_model_2 retrieved 92.77% accuracy with 0.9435 recall followed by CNN_model_1 having accuracy and recall of 91.97% and 0.875 simultaneously. We also realized that when augmentation was implemented, CNN_model_1 performed exceptionally better than CNN_model_2 securing the accuracy and recall as 67.87% and 0.3906 respectively. As for other supervised models, they all provided better results without augmentation. Therefore, we witnessed that DenseNet-201 outperformed all other supervised models both with and without augmentation followed by ResNet50 model.

Table 5.2 summarizes the results obtained for supervised and semi-supervised learning (SSL) models with and without augmentation over a limited no. of labeled samples. The results are based on the evaluation metrics of accuracy and recall where DenseNet-201 was taken as supervised model and EfficientNet-B4 was implemented for semi-supervised learning with noisy-student weights. Since DenseNet-201 achieved the best accuracy with minimal mis-classification of actual COVID cases as non-COVID, we used this model to be analyzed along with a semi-supervised learning model when limited labeled samples were available. For this inspection, we took 600 samples having only 60 no. of labeled data.

Table 5.2: Loss, accuracy and recall outcomes of Supervised (DenseNet201) and Semi-supervised (EfficientNetB4) learning using 600 number of samples with only 60 labeled data.

Results \ Model		600 samples	
		Supervised	Semi-supervised
Without augmentation	Test loss	0.9087	0.5578
	Test accuracy (%)	0.75	0.7667
	Recall	0.84	0.88
With augmentation	Test loss	0.8191	0.5379
	Test accuracy (%)	0.8	0.8333
	Recall	0.72	0.84

From the table 5.2, we found 76.67% accuracy and 0.88 recall for semi-supervised learning (SSL) without augmentation whereas with augmentation, the accuracy and recall of this model was 83.33% and 0.84 respectively. From this we depicted that although the accuracy of semi-supervised model with augmentation is significantly higher than the same model without augmentation, the recall rate is low for this model when implemented with data augmentation leading towards the higher no. of wrong predictions for actual COVID labels as non-COVID. Thus, the performance of semi-supervised learning without augmentation was detected to be more effective than the SSL with augmentation.

The exploration of supervised model with and without augmentation also showed

similar observation as that of SSL. We observed the accuracy of 75% and 0.84 recall for supervised model without implementing data augmentation whereas, we attained improved accuracy of 0.8% and reduced recall of 0.72 for supervised model with augmentation. We understood that decrease in the value of recall increases the false-negatives in the model. Thus, weighing the two metrics of accuracy and recall, we determined that like SSL, the supervised model also supported better performance without augmentation.

From the observation of table 5.2, we witnessed that in case of availability of limited annotated samples, semi-supervised learning approach showed better performance than the supervised model by perceiving remarkably higher accuracy and recall for both cases of with and without augmentation. And for all models we discovered that better performance was acquired for the models trained and predicted without augmentation.

5.2 Further Work

The experiments for this study were done in a limited time. Thus, time being a main limiting factor along with the system memory, there are some areas where further improvisations can be done so as to achieve better results for the detection of COVID-19 through CT-scans. The layers of CNN models can be increased to study its effect on coronavirus identification using CT-scan images. Since increasing the no. of layers in a deep learning model can aid in the improvement in model accuracy, this can be extended as further study on CNN with distinct no. of layers to diagnose COVID-19 with CT-scans.

Moreover, due to the limitations on the system memory we could only use 600 samples for the investigation of supervised and semi-supervised learning with 60 numbers of labeled images. Thus, this study on semi-supervised technique can be broadened with the arrangement of additional physical or virtual memory to the system. Likewise, we can also experiment with different optimization tools such as L1 and L2 regularization to optimize the learning rate in the models. With proper regularization, the model can achieve proper balance between overfitting and underfitting. Thus, implementing the regularization technique can be taken as a future work to study the enhancement in model accuracy for COVID detection through CT-scan images.

Chapter 6

Conclusion

Coronavirus impacts on the respiratory system of people and the lack of its timely diagnosis can result in the high risk to humankind due to its fast spread in large group. Utilization of medical imagining technology such as CT-scans with deep learning models have shown progressive performance on achieving the intended accuracy for COVID detection over multiple researches. This study explores the different supervised deep learning models (DenseNet201, ResNet50, CNN_model_1 and CNN_model_2) including the extended inspection on semi-supervised learning approach for the COVID-19 diagnosis with CT-scans. All of these models were trained once without using data augmentation and again with the implementation of augmentation.

The analysis of attained results showed that DenseNet201 performed as best model even with and without the image augmentation among all the applied supervised models. DenseNet achieved the highest accuracy of 99.2% and 0.9922 recall without augmentation. However, when few labeled samples were provided with large unlabeled data, semi-supervised learning technique that we implemented as EfficientNetB4 model with noisy-student weight outperformed the accuracy of supervised model i.e. DenseNet201. We witnessed that in all experiments, the model without augmentation reached significantly higher accuracy than the respective models with data augmentation.

To conclude, DenseNet201 among all supervised models provided best performance to identify coronavirus using CT-scans. However, when explored with limited annotated samples, semi-supervised learning approach achieved relatively better accuracy.

Bibliography

[1] Y. Oh, S. Park, and J. C. Ye, "Deep learning covid-19 features on cxr using limited training data sets," *IEEE transactions on medical imaging*, vol. 39, no. 8, pp. 2688–2700, 2020.

[2] P. Silva, E. Luz, G. Silva, *et al.*, "Covid-19 detection in ct images with deep learning: A voting-based scheme and cross-datasets analysis," *Informatics in medicine unlocked*, vol. 20, p. 100 427, 2020.

[3] P. Huang, T. Liu, L. Huang, *et al.*, "Use of chest ct in combination with negative rt-pcr assay for the 2019 novel coronavirus but high clinical suspicion," *Radiology*, 2020.

[4] Y. Fang, H. Zhang, J. Xie, *et al.*, "Sensitivity of chest ct for covid-19: Comparison to rt-pcr," *Radiology*, vol. 296, no. 2, E115–E117, 2020.

[5] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, and J. Liu, "Chest ct for typical coronavirus disease 2019 (covid-19) pneumonia: Relationship to negative rt-pcr testing," *Radiology*, vol. 296, no. 2, E41–E45, 2020.

[6] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[10] G. D. Rubin, C. J. Ryerson, L. B. Haramati, *et al.*, "The role of chest imaging in patient management during the covid-19 pandemic: A multinational consensus statement from the fleischner society," *Radiology*, vol. 296, no. 1, pp. 172–180, 2020.

[11] Y. Lee, Y.-S. Kim, D.-i. Lee, *et al.*, "The application of a deep learning system developed to reduce the time for rt-pcr in covid-19 detection," *Scientific reports*, vol. 12, no. 1, p. 1234, 2022.

[12] R. Chauhan, K. K. Ghanshala, and R. Joshi, "Convolutional neural network (cnn) for image detection and recognition," in *2018 first international conference on secure cyber computing and communication (ICSCCC)*, IEEE, 2018, pp. 278–282.

[13] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, 2010, pp. 242–264.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[16] S. S. Keh, "Semi-supervised noisy student pre-training on efficientnet architectures for plant pathology classification," *arXiv preprint arXiv:2012.00332*, 2020.

[17] R. Alizadehsani, D. Sharifrazi, N. H. Izadi, *et al.*, "Uncertainty-aware semi-supervised method using large unlabeled and limited labeled covid-19 data," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 3s, pp. 1–24, 2021.

[18] C. H. Han, M. Kim, and J. T. Kwak, "Semi-supervised learning for an improved diagnosis of covid-19 in ct images," *PLoS One*, vol. 16, no. 4, e0249450, 2021.

[19] T. Rahman, A. Khandakar, Y. Qiblawey, *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104 319, 2021.

[20] E. Benmalek, J. Elmhamdi, and A. Jilbab, "Comparing ct scan and chest x-ray imaging for covid-19 diagnosis," *Biomedical Engineering Advances*, vol. 1, p. 100 003, 2021.

[21] X. Zhang, Y. Wang, N. Zhang, D. Xu, and B. Chen, "Research on scene classification method of high-resolution remote sensing images based on rfpnet," *Applied Sciences*, vol. 9, no. 10, p. 2028, 2019.

[22] P. Tumuluru, P. Srinivas, R. B. Devabhaktuni, K. V. Attili, P. M. Ramesh, and B. R. P. Kalyan, "Detection of covid disease from ct scan images using cnn model," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, IEEE, 2022, pp. 178–184.

[23] N. Rajawat, B. S. Hada, M. Meghawat, S. Lalwani, and R. Kumar, "C-covidnet: A cnn model for covid-19 detection using image processing," *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 10 811–10 822, 2022.

[24] A. Rosebrock, *Imagenet: Vggnet, resnet, inception, and xception with keras-pyimagesearch.[online] pyimagesearch*, https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/, Accessed: 2022-10-06, 2017.

[25] M. Shaha and M. Pawar, "Transfer learning for image classification," in *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, IEEE, 2018, pp. 656–660.

[26] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*, Springer, 2019, pp. 191–202.

[27] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, "A generic intelligent tomato classification system for practical applications using densenet-201 with transfer learning," *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.

[28] D. Singh, V. Kumar, and M. Kaur, "Densely connected convolutional networks-based covid-19 screening model," *Applied Intelligence*, vol. 51, pp. 3044–3051, 2021.

[29] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on densenet model," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1651, 2020, p. 012 143.

[30] M. Farooq and A. Hafeez, "Covid-resnet: A deep learning framework for screening of covid19 from radiographs," *arXiv preprint arXiv:2003.14395*, 2020.

[31] DataRobot, *Semi-supervised learning.[online] datarobot*, https://www.datarobot.com/blog/semi-supervised-learning/, Accessed: 2023-02-12, 2020.

[32] L. Nwosu, X. Li, L. Qian, S. Kim, and X. Dong, "Semi-supervised learning for covid-19 image classification via resnet," *arXiv preprint arXiv:2103.06140*, 2021.

[33] Bharatdhyani, *Efficient nets with noisy student training.[online] towards data science*, https://towardsdatascience.com/efficient-nets-with-noisy-student-training-5ac6e239ff14, Accessed: 2023-02-26, 2021.

[34] Antoreepjana, *Tf keras pretrained model weights.[online]*, https://www.kaggle.com/datasets/antoreepjana/tf-keras-pretrained-model-weights, Accessed: 2023-02-21, 2021.

[35] F. Chollet *et al.* "Keras." (2018), [Online]. Available: https://github.com/keras-team/keras/blob/v2.12.0/keras/applications/densenet.py#L395-L417.

[36] F. Chollet *et al.* "Keras." (2018), [Online]. Available: https://github.com/keras-team/keras/blob/v2.12.0/keras/applications/resnet.py#L499-L533.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[38] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in neural information processing systems*, vol. 2, 1989.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[41] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification," *MedRxiv*, pp. 2020–04, 2020.

[42] Plameneduardo, *Sars-cov-2 ct-scan dataset.[online]*, www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset, Accessed: 2023-01-28, 2021.

[43] M. KA, *Covid-19 detection using vgg16 98%acc.[online]*, https://www.kaggle.com/code/marioska/covid-19-detection-using-vgg16-98-acc/notebook, Accessed: 2023-01-15, 2022.

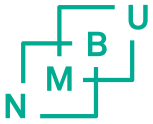
[44] T. Ai, Z. Yang, H. Hou, *et al.*, "Correlation of chest ct and rt-pcr testing for coronavirus disease 2019 (covid-19) in china: A report of 1014 cases," *Radiology*, vol. 296, no. 2, E32–E40, 2020.

[45] P. Angelov and E. Soares, "Towards explainable deep neural networks (xdnn)," *Neural Networks*, vol. 130, pp. 185–194, 2020.

[46] M. Grewal, M. M. Srivastava, P. Kumar, and S. Varadarajan, "Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, IEEE, 2018, pp. 281–284.

[47] A. Basu, K. H. Sheikh, E. Cuevas, and R. Sarkar, "Covid-19 detection from ct scans using a two-stage framework," *Expert Systems with Applications*, vol. 193, p. 116 377, 2022.

Thank you.



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway