COVID-DenseNet: A deep learning architecture to detect COVID-19 from chest radiology images

Laboni Sarker[1], Md. Mohaiminul Islam[2], Tanveer Hannan[3], and Zakaria Ahmed[4]

[1]laboni@uap-bd.edu, [2]mmiemon@uap-bd.edu, [3]tanveer.hannan@campus.lmu.de, [4]zakaria.ahmed@enosisbd.com
[1 2 3 4]B.Sc. from Department of CSE, Bangladesh University of Engineering and Technology
[1 2]Lecturer, University of Asia Pacific, Bangladesh, [3] M.Sc.Student, Ludwig Maximilian University of Munich, Germany
[4]Software Engineer, Enosis Solutions, Bangladesh
[*]All authors contributed equally and share the first-authorship of this paper.

Keywords: Deep learning, CNN, DenseNet, COVID-19, Transfer learning.

***Abstract*—Coronavirus disease 2019 (COVID-19) is a pandemic infectious disease that has a severe risk of spreading rapidly. The quick identification and isolation of the affected persons is the very first step to fight against this virus. In this regard, chest radiology images have proven to be an effective screening approach for COVID-19 affected patients. Several AI-based solutions have been developed to make the screening of radiological images faster and more accurate in detecting COVID-19. In this study, we are proposing a deep learning-based approach using Densenet-121 to effectively detect COVID-19 patients. We incorporated the transfer learning technique to leverage the information regarding the radiology image learned by another model (CheXNet), which was trained on a large radiology dataset of 112,120 images. We have trained and tested our model on the COVIDx dataset containing 13,800 chest radiography images across 13,725 patients. To check the robustness of our model, we performed both two-class and three-class classifications and achieved 96.49% and 93.71% accuracy, respectively. To further validate the consistency of our performance, we performed patient-wise k-fold cross-validation and achieved an average accuracy of 92.91% for three class tasks. Moreover, we performed an interpretability analysis using Grad-CAM to highlight the most significant image regions in making a prediction. Besides ensuring trustworthiness, this explainability can also provide new insights about the critical factors regarding COVID-19. Finally, we developed a website that takes chest radiology images as input and generates probabilities of the presence of COVID-19 or pneumonia and a heatmap highlighting the probable infected regions. Source code for reproducing results and model's weights are available.** [1]

1. Introduction

On February 11, 2020, the World Health Organization (WHO) defined the novel coronavirus (2019-nCoV) as Coronavirus Disease 2019 (COVID-19) as an epidemic disease. The 2019-nCoV is a new member of the Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV) family and is defined as SARS-CoV-2. Though it started in Wuhan, Hubei Province, China, it had spread nationwide within a very short period and turned into an outbreak [1]. Being concerned by the frightening levels of spread and severity, WHO characterized COVID-19 as a pandemic in the month of March, and it became a global issue as there are no specific vaccines or treatments available for this virus. As it can infect people easily and can spread from person-to-person very spontaneously, the quick identification and isolation of the affected person is the very first step to fight against this virus. Polymerase chain reaction (PCR) is the primary method for detecting COVID-19 cases. It can detect SARS-CoV-2 RNA from respiratory specimens such as nasopharyngeal or oropharyngeal swabs [2]. Though this method is the most effective one, it is very time consuming and intensive lab work is required after the collection of the samples to get the result.

Another approach is the examination of chest radiography imaging (e.g., radiology or computed tomography (CT) imaging), which can be conducted faster but an expert analysis is needed to interpret the subtle differences. For removing this bottleneck, many AI-based systems have been proposed to detect COVID-19 from radiography images. Moreover, AI solutions are much faster than traditional methods where radiologists need to examine the images by hand. Some previous works used AI solutions with CT images to detect COVID-19 [3] [4]. But CT scans are more costly and in most cases CT image dataset is not publicly available. On the other hand, X-rays are more widespread, quicker and cheaper alternative. Therefore, we choose chest X-ray images in our study. We used publicly available COVIDx dataset [5] to train a deep learning model which can efficiently detect COVID-19 from chest X-ray images.

In our work, we have used the Dense Convolutional Network (DenseNet) [6] of 121 layers as our model. DenseNet makes the training of deep learning models manageable by alleviating the vanishing gradient problem, increasing feature reuse, and decreasing parameter usage. It has attained state-of-the-art performance in several computer vision tasks. Moreover, DenseNet has been used successfully in disease prediction from radiology images. In paper [7], DenseNet-121 was used to detect 14 kinds of diseases from chest radiology images (CheXNet), and it achieved better performance than practicing academic radiologists. Paper [8] also used

[1] Code for reproducing is results available at https://github.com/mmiemon/COVID-DenseNet and models' weights can be found at https://bit.ly/2YZwyk3

DenseNet-121 for disease prediction from radiology images of the ChestRadiology-14 dataset and further improved the performance achieved by paper [7]. Being motivated by the excellent performance of DenseNet on radiology images (e.g. paper [7] and [8]), we used DenseNet-121 as our deep learning model. Moreover, we initialized our model's weights by the weights of CheXNet [7]. Our intuition of using this transfer learning technique was the utilization of the information regarding radiology images present in the CheXNet pre-trained model. Because CheXNet was trained on ChestRadiology-14 [9] dataset which contains 112,120 frontal view radiology images from 30,805 unique patients.

We trained our model on the COVIDx dataset [5] containing 13,800 chest radiography images across 13,725 patients. We tested our model for two-class classification (COVID-19 and non-COVID-19) and three-class classification (COVID-19, Pneumonia, and Normal). We achieved 96.49% accuracy for two-class and 93.71% accuracy for three-class classification. These results show that our model is capable of differentiating COVID-19 radiology images not only from those of a healthy person but also from those of other pneumonia patients. To check the robustness and consistency of our model, we performed 10-fold cross-validation where no two folds contain COVID-19 images from the same patients (patient-wise cross-validation) and achieved an average accuracy of 92.91%.

We used Gradient-weighted Class Activation Mapping (Grad-CAM) [10] to visualize how our model works. Using Grad-CAM, we created a heatmap for each input image, highlighting the most significant region for which our model makes a prediction. This feature ensures interpretability as well as the trustworthiness of our model. It also works as a safeguard that our model is not making predictions based on inappropriate portions of the input radiology image. Moreover, this will help doctors and clinicians to visualize the most significant features and give insights about the critical factors of COVID-19 patients.

It is important to develop a tool for allowing users to use our model and generate predictions effortlessly. We developed a web application [11], which adapts our model to provide real-time predictions. We used TensorFlow.js for converting our model to work in the browser. The web application also generates heatmaps of the radiology images. A RESTful API is implemented using Flask micro web framework, which is used to create the heatmaps.

The rest of the paper is organized as follows. Section 2 provides the literature review on some state-of-the-art methods developed for detecting COVID-19 in recent times. Section 3 presents our proposed method, which is followed by a discussion on the experimental results in section 4. Finally, section 6 concludes the paper.

2. Related Works

Computer vision [12] helps us building autonomous systems to perform tasks similar to the human visual system and, in some cases, better performance than human vision. One of the significant contributions of computer vision is in better diagnosing, treatment, and prediction of diseases using medical imaging data [13]. Deep neural network(DNN) has a great capability in the image classification task [14] and convolutional neural network(CNN, or ConvNet) [15] is one of the most popular classes of DNN. AlexNet [16], VGG [17], Inception [18], ResNet [19], DenseNet [6] are some of the popular convulational networks.

AlexNet [16] architecture is composed of five convolutional layers, followed by three fully connected layers. Instead of the standard tanh or sigmoid function, it uses ReLU(Rectified Linear Unit) for the non-linear part after each convolutional and fully connected layers. ReLU is much faster in case of training than the sigmoid function. It also solved the problem of over-fitting by introducing the idea of a drop-out layer.

VGG16 and VGG19 architecture are from the VGG group. Instead of large kernel-sized filters used in AlexNet, VGG16 and VGG19 use multiple 3X3 kernel-sized filters consecutively. This multiple stacked smaller size kernel works better than AlexNet because it increases the depth of the network and provides the chance to learn more complex features at a low cost. VGG16 contains 16 weight layers where VGG19 has 19. In the VGG group, convolutional layers are followed by fully connected layers. Also, all the hidden layers are equipped with ReLU.

Inception is initially known as GoogleNet. Though VGG is a good model, it takes an extensive computational cost in terms of memory and time. Inception reduces the cost by introducing a bottleneck layer(1X1 convolutional filter). Also, it uses convolutions of different sizes like 5X5, 3X3, 1X1 to capture the details. It also reduces the total number of parameters by replacing the fully-connected layers with a global average pooling after the last convolutional layer.

ResNet is a deeper network than VGG16(with 16 layers) and VGG19(with 19 layers) but smaller because of the use of global average pooling instead of the fully-connected layers(like inception model). By adding some connections directly to the output skipping training from a few layers, it tries to handle the problem of vanishing gradient descent. This is called a residual network. That means with the help of this type we can train very deep networks. ResNet50 is from this group with 50 weight layers.

DenseNet architecture is designed in such a way that all the layers are directly connected ensuring maximum information flow in the network. Also unlike ResNet, here features are concatenated. This architecture requires less parameters and computation to get state-of-art performace.

Numerous works have been done in detecting COVID-19 from radiography images. Different model architectures have been used for accurate detection of the disease. COVID-Net [5] introduced a deep convolutional neural network design for detecting COVID-19 using the COVIDx dataset, which comprises 13,975 chest X-ray images. COVID-Net network architecture uses projection-expansion-projection-extension (PEPX) design pattern (Figure 1). They utilized a human-machine collaborative design strategy. This strategy

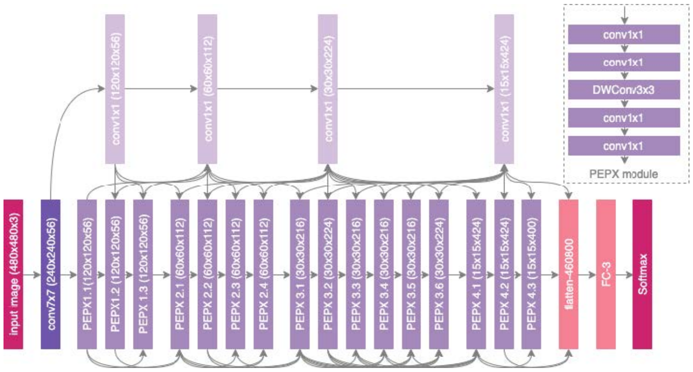

Figure 1: COVID-Net [5] Architecture

combines human-driven principled network design prototyping and machine-driven design exploration. In the final detection, they have used 4 class classifications: Normal, Bacterial, Non-COVID-19 Viral, and COVID-19 Viral.

Table I: Sensitivity(Recall) of COVID-Net [5]

Normal	Bacterial	Non-Covid19 Viral	COVID19 Viral
73.9	93.1	81.9	100.0

Table II: Precision of COVID-Net [5]

Normal	Bacterial	Non-Covid19 Viral	COVID19 Viral
95.1	87.1	67.0	80.0

From Table I and II, it is clear that the COVID-Net is very good at detecting COVID-19 infection as sensitivity(recall) is 100%. A small portion of radiology images is misclassified as COVID-19. But for other classes, both the sensitivity(recall) and positive predictive value(precision) rate can be improved. So, there is a lot more to contribute to properly detect the COVID-19 from other respiratory infections as they are all very similar. The COVID-Net model has achieved a test accuracy of 93.3%.

COVNet [4] has differentiated COVID-19 from Community-Acquired Pneumonia(CAP) from chest CT images. The dataset was collected from 6 hospitals and is not publicly accessible. COVNet is a 3D deep learning framework(can extract both 3D global and 2D local representative features) and contains a ResNet50 [19] as the backbone. They have used U-net [20] to segment the lung region from the chest radiology images. The training dataset contains 1165 images of COVID-19, 1560 from CAP, and 1193 of non-pneumonia CT scans. They have trained their model with both CAP and non-pneumonia CT images to check the robustness of how efficiently their model can differentiate between COVID-19 and other similar lung diseases. Table III gives us an overview of the performance of their model, which seems very promising but not for public use.

Table III: Sensitivity(Recall) & Specificity of COVNet [4]

	COVID-19	CAP	Non-pneumonia
Sensitivity	90	87	94
Specificity	96	92	96

Transfer learning is a technique where the knowledge gained from solving a specific problem is transferred to solve a different but similar problem. Transfer learning can provide great results in detecting various irregularities in small medical image datasets. Paper [21] adopted a transfer learning technique to evaluate the performance of some state-of-the-art convolutional neural network architectures. They used two different datasets in this experiment. Table IV presents a summary of their datasets.

Table IV: Datasets used in paper [21]

	COVID-19	Pneumonia	Normal
Dataset 1	224	700	504
Dataset 2	224	714	504

The author evaluated five CNN models which are VGG19 [22], MobileNet v2 [23], Inception [24], Xception [25], and Inception ResNet v2 [24]. Among these models, MobileNet v2 [23] provided the best results in terms of specificity in their particular datasets. Table V presents the results of MobileNet v2 [23] on Dataset 2. The results from Table V is promising. But this experiment was performed on a particular small dataset. For practical medical use, especially in a pandemic like COVID-19, this model needs to perform well on large datasets as well.

DarkCovidNet is another deep learning model proposed in paper [26]. The author used Darknet-19 [27] model as their base model and designed DarkCovidNet architecture. DarkCovidNet has 17 convolutional layers in contrast to the 19 convolutional layers in Darknet-19.

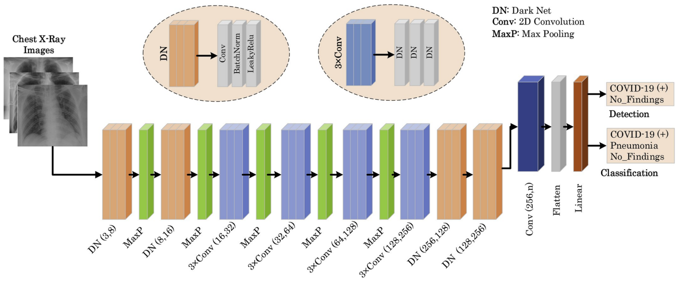

Figure 2: DarkCovidNet [26] Architecture

DarkCovidNet used a dataset of 1125 chest X-ray images, which comprises 125 images that were diagnosed with COVID-19, 500 images with pneumonia, and 500 images were

Table V: Results of MobileNet v2 [23] on Dataset 2

Accuracy		Sensitivity	Specificity
(2-class)	(3-class)		
0.9678	0.9472	0.9866	0.9646

normal. The result of their experiment is presented in Table VI.

Table VI: Experiment results of DarkCovidNet [26]

Classification	Sensitivity	Specificity	Precision	F1-score	Accuracy
2-class	0.9513	0.953	0.9803	0.9651	0.9808
3-class	0.8535	0.9218	0.8996	0.8737	0.8702

From the table, it is clear that DarkCovidNet is very good at detecting COVID-19 for 2-class classification. But there are room for improvements for 3-class classification. This method needs more contribution to detect COVID-19 from other respiratory infections as well.

3. Methodology

As our dataset contains a small amount of COVID-19 radiography images, learning a deep learning model can be very problematic in this scenario because deep learning models require a large number of data for training from scratch. Transfer learning can used as a viable solution to this problem. In transfer learning algorithms, information learned in one domain is utilized to perform another task in other domains. For example, it a common practice to initialize deep learning models with weights learned by the ImageNet dataset [28] in computer vision domains. ImageNet is an enormous dataset containing 3.2 million images from different sources. The main advantage of such transfer learning algorithms is that pretrained models on dataset like ImageNet have already learned different features of different images. Later, these learned features can be utilized for any other domain-specific tasks by fine-tuning the pretrained model on the dataset of that domain. However, if the dataset used for pretraining is similar to the dataset of a specific task, it is expected that the pretraining process will have more relevant and useful features for that task. From this intuition, we utilized the CheXNet model [7] trained on chest X-ray images for pretraining instead of using AlexNet, VGG, inception or ResNet which are pretrained on ImageNet(with 1000 categories of images but with no chest X-ray images). The CheXNet model is basically the DenseNet-121 model, which was trained on the ChestRadiology-14 dataset [9] containing 112,120 chest radiology images from 30,805 unique patients to detect 14 different diseases from radiography images. As the ChexNet was trained on a huge dataset of radiography images, it is expected that the ChexNet has learned various features relevant to the radiography images. To utilize those learned features related to the radiography image, we used transfer learning from CheXNet by initializing our model by the weights of CheXNet. We used DenseNet-121 [6] as a deep learning model for feature extraction because this model has several advantages over other deep learning models for the image domain, which is explained in subsection 3(Model architecture).

The complete workflow proposed method is shown schematically in Figure 3. First, we load the pretrained DenseNet-121 model with the CheXNet model for feature extraction. Then we remove the last layer of the CheXNet model and replace it with a classifier specific to our task. For 3-class classification (COVID-19, Pneumonia, and Normal), our classifier is a fully connected layer with three neurons. For 2-class classification (COVID-19 and non-COVID-19), it is a fully connected layer with three neurons. Then we train our model (COVID-DenseNet) with the radiography images of COVIDx dataset [5] containing 13,800 radiography images of 13,725 unique patients. In the testing phase, this trained COVID-DenseNet model is efficiently used to predict the radiography image class. Finally, a gradient-based localization algorithm (Grad-Cam) [10] is used to identify the significant image regions that contribute to the prediction decisions.

3.1. Data Generation

Radiology images of COVID-19 infected patients are rare. We used COVIDx dataset assembled by [5]. We downloaded the images on 7th April, 2020. They combined open source databases with chest radiology or CT images from [29], [30], [31]. We only used X-ray images to train our model and no CT scan images were used. The total number of COVID-19 infected Chest images are only 238. This number is extremely small compared to the number of radiology images available for pneumonia infected and healthy persons, which are 6045 and 8851 respectively. So the data is highly skewed because of the scarcity of images of COVID-19 patients. To deal with this unbalanced dataset, we augmented only the COVID-19 images in the training set. The following Table IV shows the distribution of the dataset before and after augmentation.

Table VII: Class Distribution.

Augmentation/ Class	Normal	Pneumonia	COVID-19
No	8851	6045	238
Yes	8851	6045	11416

The train-test split ratio is fixed at 0.9. We also stratified the train, validation, and test split so that the proportion is maintained in each set. We augmented the training data in six different methods. These are width shift, height shift, horizontal flip, rotation, brightness change, and zoom in or zoom out. We created 9 different images randomly for each category. So each COVID-19 radiology image in the training set has a total of 54 augmentations. For validating the result, the dataset was prepared for 10-fold cross-validation keeping the proportions of the class labels the same for each fold. We maintained augmentation leakage by creating an indexing system so that the augmentation of images in one fold does not fall in another one. We also maintained an index for patient ids' so that no two folds have images of the same patient. Each patient a has variable number of images. So dividing the patients randomly among 10-folds would create an imbalance in terms of the number of images in each fold. So we had to maximize both the number of patients and images for each fold at the same time. We thus reduced the correlation between train and test images.

The COVID-19 dataset is currently growing. We created a new data injection method to add new images to our dataset.

This method also performs all the balancing acts to reduce the correlation of images between each fold.

3.2. Preprocessing

We used minimal preprocessing of the dataset before it is fed to our model. The only preprocessing was resizing every image to a similar dimension. We used images of height 224 pixels, width 224 pixels, and the number of channel 3 (224*224*3). Minimal preprocessing makes our inference process faster, so when testing, we can generate the model's output (prediction and heatmap) in real-time.

3.3. Model Architecture

Our model is comprised of two parts, feature extractor, and classifier. For the feature extractor, we used Densenet-121 [6], and for the classifier, we used a fully connected layer with softmax activation function.

The main building block of DenseNet-121 is DenseBlock [6]. These DenseBlocks consist of Convolution Layers. In general, CNN architectures are hierarchical, so feature maps of $(l-1)$th layer are input to the lth layer. But in DenseNet, feature-maps off all preceding layers are concatenated and used as input for any particular layer. Also, it's own feature-maps are used as inputs for all subsequent layers. So, for lth layer, features maps of all preceding layers $X_0, X_1, ..., X_l - 1$ are concatenated and used as it's input.

$$X_l = H_l([X_0, X_1, ..., X_l - 1]) \tag{1}$$

Here H_l represents the lth layer, X_l is the output of the lth layer, and $[X_0, X_1, ..., X_l - 1]$ represents the concatenation operation.

This special design improves information flow through the network and alleviates vanishing gradient problem. Moreover, DenseNet enhances feature reuse and parameter efficiency and provides each layer the collective knowledge of the network. Another important reason for choosing DenseNet as our architecture is that dense connection has a regularization effect, and it reduces over-fitting on training with smaller data sets [6], which is our case.

DenseNet-121 has four dense blocks and a transition layer between every two dense blocks (Figure 4). Each dense block consists of several convolution layers, and each transition layer consists of a batch normalization, a convolution, and an average pooling layer. To increase nonlinearity ReLU activation function is used in DenseNet, which can be described as:

$$ReLu(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{2}$$

In our model, the final layer of the Dense-121 is a global average pooling layer that generates the features from the input image. These features are used by the classifier to make the final prediction. For the classifier, we used a fully connected layer, followed by a softmax activation function. For 3-class classification, we used a fully connected layer of three units, and for 2-class classification, we used a fully connected layer of two units. The softmax activation normalizes the output of the fully connected layer and generates a probability distribution over the predicted output classes. The equation of the softmax function can be written as follows:

$$\sigma(\vec{z}_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \tag{3}$$

Here, $\vec{z}$ is the input vector of the softmax function, z_i values are the components of the input vector, and K is the number of classes.

3.4. Model Implementation

DenseNet-121 consists of 121 densely connected convolutional layers with a fully connected(FC) layer of 1000 units as its final output layer. We removed the final layer and used it as our feature extractor. Then we added a classifier that consists of an FC layer and a softmax activation. We initialized our models weights with the weights of CheXNet [7], which was trained on ChestRadiology-14 [9] dataset of 112,120 chest radiology images. Since CheXnet was already trained to extract features from chest radiology images, we used this transfer learning method to leverage the pretrained model.

The network was trained end-to-end with a backpropagation algorithm to minimize the loss function. We used categorical cross-entropy as the loss function of our model. The loss function can be written as the the following equation:

$$L(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^i \ln h_\theta(x^i)_k\right] \tag{4}$$

Here, n is the number of training samples, k is the number of classes, θ is the model parameter, y_k^i is the actual level of ith training sample, $h_\theta(x^i)_k$ is the output at kth node for ith training sample.

Adam optimizer [32] was used to update the model weight θ. Weight update equation can be written as follows:

$$\theta_t = \theta_{t-1} - \alpha\frac{m_t}{\sqrt{v_t} + \epsilon} \tag{5}$$

where,

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t \tag{6}$$

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2 \tag{7}$$

Here, θ_t is weight at time t, θ_{t-1} is weight at time $t-1$, α is the learning rate, g_t is gradient of the weights with respect to the loss function, m_t is the first moment estimate and v_t is the second raw moment estimate, β_1 and β_2 are hyperparameters. We used ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The initial learning rate was .00001, and it was reduced by the factor of 0.1 when the validation loss plateaued. Because when loss begins to plateau, reducing the learning rate helps the optimizer to find the minimum in the loss surface more efficiently. We used early stopping with patience 5, which means if validation set

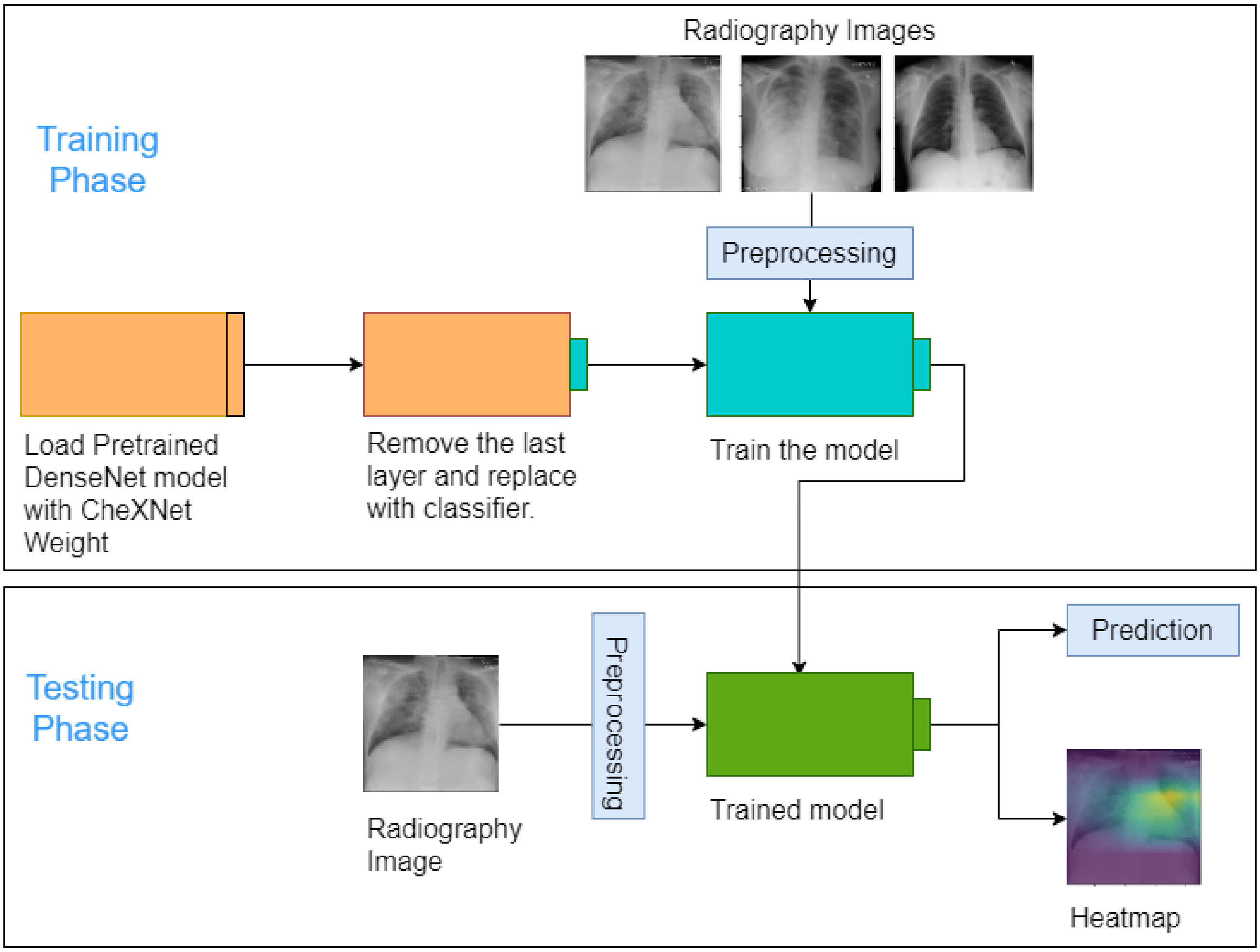

Figure 3: Schematic diagram of the complete workflow.

performance does not improve for 5 epochs, then the training will be stopped, and the model's parameter with the best validation set performance will be restored. This strategy helps to stop the over-fitting of the model.

3.5. Prediction and Heatmap generation

In Section 4, we described the implementation of our model at the training phase. After training, we get a trained model with a learned weight that we can use at the testing phase to make a prediction. In the testing phase, our model receives a chest radiography image of a patient and does minimal preprocessing (resizing in shape 224*224). Then the image is fed to the trained model to generate final predictions.

Besides making the prediction, our method also generates a heatmap of the input image. This heatmap highlights the significant regions in the input image that contributed most to a particular prediction. This can help doctors identifying the critical areas of an affected patient's chest, which may lead the model to identify him as a COVID-19 affected person.

We used Gradient-weighted Class Activation Mapping (Grad-CAM) [10] to generate the heatmap. This heatmap is a coarse localization map which highlights the important regions in the input image for making the prediction. Grad-CAM exploits the last convolution layer of a CNN architecture to generate the activation map. The intuition behind choosing the last convolution layer is that the deeper CNN layers capture the high-level information most. Early CNN layers cannot capture the high-level information and the later fully-connected layers loses the spatial information. Therefore, last CNN is a good choice which captures both spatial and high-level information. In this approach, to generate the heatmap of width u and height v, we computed the gradient of the target class with respect to the feature maps of the final convolutional layer. These gradients are average-pooled over width and height dimension to generate the neuron importance weights for the target class.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\delta y^c}{\delta A_{ij}^k} \tag{8}$$

Here, c is the target class, y^c is the gradient of c before softmax, A^k is the feature map of the last convolution layer, summation over i and j represents the average-pool operation, and α_k^c is the generated importance weights for class c.

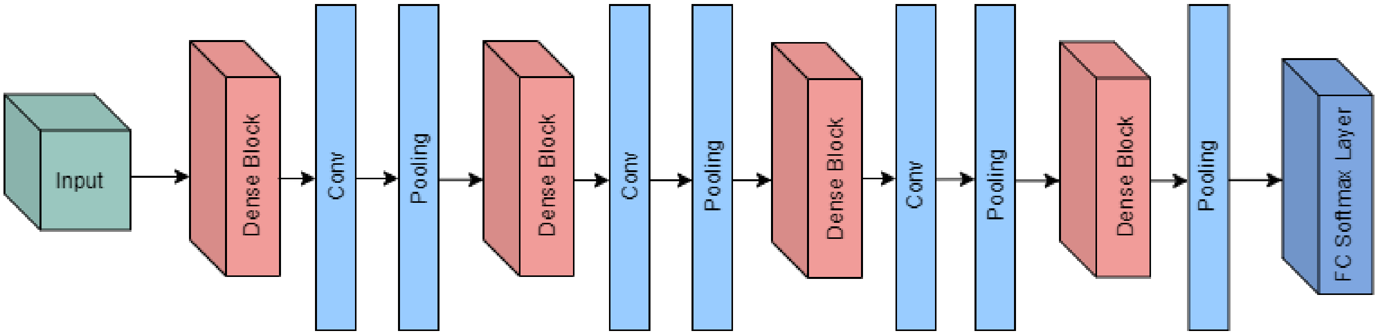

Figure 4: DenseNet-121 with 4 dense blocks and 3 transition layers.

After that, a weighted combination of ReLU activation is applied. This produces a coarse localization map or heatmap of the size of the final convolution layer. The ReLU activation function is applied because it emphasises the features that have a positive influence on the final prediction.

$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha^c_k A^k\right) \tag{9}$$

Here, $L^c_{Grad-CAM}$ is the generated heatmap, α^c_k is the generated importance weights for class c, and A^k is the feature map of the last convolution layer.

Finally, this heatmap is resized and superimposed on the input image, which generates the final heatmap like Figure 6. Figure 6 shows the actual Chest-Xray images along with heatmaps of a COVID-19 affected, a pneumonia affected, and a normal person. We can see that our model is mainly emphasizing on the lung areas in detecting COVID-19 or pneumonia.

4. Results and Discussion

As the dataset of COVID-19 cases is not that much available, to be assured about the performance of our model, we performed both 2-class classification (COVID-19 and non-COVID-19) and 3-class classification (COVID-19, Pneumonia, and Normal). Moreover, we performed patient-wise 10-fold cross-validation to guarantee the robustness of our model. Finally, in the qualitative analysis, we analyzed the decision-making behavior of our model to ensure interpretability and trustworthiness.

4.1. Quantitative Analysis

To show this particular analysis, we calculated the test accuracy, precision, recall, and f-score of each experimental setup. As we have an imbalanced class distribution, accuracy alone cannot provide a proper performance overview. So we also included the other metrics mentioned above to assess our model. Recall is the fraction of test instances of a class that has been correctly predicted whereas precision is the fraction of correctly classified object assigned to the class. F-score is simply the harmonic mean of these two values. The equation for calculating these values are:

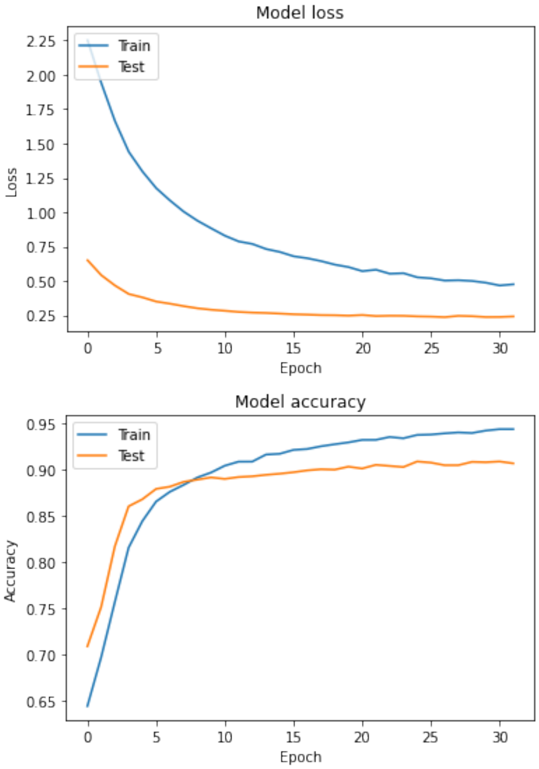

Figure 5: Accuracy vs epoch and loss vs epoch for train and validation set (Experiment 1).

$$recall = \frac{TP}{TP + FN} \tag{10}$$

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$fscore = \frac{2 * recall * precision}{recall + precision} \tag{12}$$

Here, TP, FN, and FP are true positives, false negatives, and false positives. For multi-class prediction, f-score can be computed in two different ways: micro-average f-score and macro-average f-score. In the micro-average f-score, the larger classes get more weight and hence give the same evaluation as accuracy. However, in the macro-average f-score, the smaller classes get the same importance as the bigger ones. We choose this macro-average to evaluate our models.

4.1.1. Three class classification

In this experiment, we performed a 3-class classification (COVID-19, Pneumonia, and Normal). We split our dataset in train, validation, and test set in an 80%-10%-10% ratio. There was no common image among the three sets, and augmentation was performed separately in each set. Results are shown in Table VIII and IX. The results show 95% accuracy for normal or healthy people where the model correctly predicted pneumonia and COVID-19 with 93% and 87% accuracy. The low accuracy for COVID-19 cases is due to the limited amount of training images. Overall accuracy is 94%, which is quite good. As our dataset is imbalanced, we analyzed precision, recall and f-score. We can see that all performance matrices are more that 90% for pneumonia and normal and 87% for COVID-19. These results indicates our model is capable of classifying COVID-19, pneumonia, and normal chest radiography images with high confidence.

4.1.2. Patient wise cross validation

To check the robustness and performance consistency of our model, we have done patient-wise 10-fold cross-validation as the data-set contains multiple radiology images of different days for the same patient. Each fold has images of different patients, and augmentation was performed separately in each of them. Table X shows the average accuracy, precision, recall, and f-score of all folds. The result is quite similar to the 3-class classification performance, which indicates that the performance of the model does not deviate too much even if we change the train and test instances. So the model is robust and can work quite well with new images.

4.1.3. Two class classification

The same setup (train, validation, test split in 80%-10%-10% ratio) as the first experiment with only 2 class labels (COVID-19 and Non-COVID-19) was used in this experiment. Results are shown in Table XI and Table XII. We see improvement in detecting COVID-19 cases with binary classification as expected. The model can classify better in this setup with an overall accuracy of 96%. Overall precision, recall and f-score is also 96%. We also analyzed accuracy, precision, recall and f-score separately for COVID-19 and Non- COVID-19 and all matrices is more that 90%. These results ensures that, this model can serve very well in case one only wants the successful detection of COVID-19.

4.1.4. Comparison with standard computer vision algorithm

We compared our model other state of the art computer vision models. Particularly, we compared with inception-v3, resnet50, and VGG-16 with the same setup as three class classification. Initial weights were set to the trained models on ImageNet, which is a large database of images for classification tasks. Weights of the trained models are available, and we trained the radiology dataset with the initial weights of these trained models, which is a transfer learning approach. The comparative performance is shown in Table XV. Overall accuracy for COVID-DenseNet is 94%, while accuracy for resnet50, inception-v3, and VGG-16 are 92%, 90%, and 93%. COVID-DenseNet has higher accuracy for Pneumonia and Normal cases also. The results clearly indicate that our model outperforms other computer vision models.

4.1.5. Different initial weights

We trained Densenet with initial weights set to the trained model from ChestXNet. The reason for performing transfer learning with this weight is that the necessary features associated with radiology images have already been extracted, and we might use that information in the model to efficiently change the problem set to our use case with COVID-19 images. We trained our model with image-net and random weight initialization to verify our claim of a performance boost with ChestXNet transfer learning. The results are shown in Table XIV. The overall performance increased with weight initialization from ChestXNet.

4.1.6. Comparison with state of the art methods

We have encountered some state-of-the-art methods developed for detecting COVID-19. The accuracy comparison with these methods is shown in Table XV. COVID-Net [5] introduced a deep convolutional neural network for detecting COVID-19 using COVIDx dataset which comprises 13,975 chest X-ray images and achieved an accuracy of 93.3%. Apostolopoulos2020 [21] adopted transfer learning to evaluate some state-of-the-art CNN architectures and obtained an accuracy of 96.78% for 2-class classifications and an accuracy of 94.72% for 3-class classifications using MobileNets v2 [23]. Ozturk [26] used DarkNet [33] model as their classifier and proposed DarkCovidNet model. They obtained 98.08% accuracy for 2-class and 87.02% accuracy for multi-class classifications. ResNet50 plus SVM [34], A deep learning based methodology achieved an overall accuracy of 95.38% and ResNet50 [35], a deep convolutional neural network model, achieved 98% accuracy for 2-class classifications. COVNet [4], which is a 3D deep learning model and uses ResNet50, obtained 93.24% accuracy for 3-class classifications.

Class/ Metric	**Accuracy**	**Precision**	**Recall**	**f-score**
Overall	0.94	0.94	0.94	0.94
COVID-19	0.87	0.87	0.87	0.87
Pneumonia	0.93	0.95	0.93	0.94
Normal	0.95	0.94	0.96	0.95

Table VIII: Three-class classification results.

4.2. Qualitative Analysis

As described in section 5, we used Gradient-weighted Class Activation Mapping (Grad-CAM) [10] to analyze our model's

[2] This metric is not mentioned directly in the paper. We calculated it from their Sensitivity and Specificity results.

Predicted/ Actual	COVID-19	Pneumonia	Normal
COVID-19	27	3	1
Pneumonia	2	93	3
Normal	2	4	96

Table IX: Confusion matrix for three-class classification.

Class/ Metric	Accuracy	Precision	Recall	f-score
Overall	0.93	0.92	0.92	0.92
COVID-19	0.86	0.77	0.85	0.81
Pneumonia	0.90	0.93	0.91	0.92
Normal	0.95	0.94	0.94	0.94

Table X: Patient-wise cross-validation results.

Class/ Metric	Accuracy	Precision	Recall	f-score
Overall	0.96	0.96	0.96	0.96
COVID-19	0.93	0.90	0.94	0.92
Non COVID-19	0.96	0.97	0.97	0.97

Table XI: Two-class classification results.

Predicted/ Actual	COVID-19	Non COVID-19
COVID-19	29	3
Non COVID-19	2	97

Table XII: Confusion matrix for two-class classification.

output. It produces a coarse localization map highlighting the significant regions in the input image for making the prediction. From the generated heatmap, we can approximately localize the possible affected regions. Figure 6 shows the actual Chest-Xray images along with heat-maps of a COVID-19 affected, a pneumonia affected, and a normal person. We can see that our model is mainly emphasizing on the lung areas in detecting COVID-19 or Pneumonia as expected. This indicates the infected lung areas which led the model to predict COVID-19 or Pneumonia. On the other hand, heatmap of a normal person shows that our model is not emphasizing on any particular region. This indicates there is no affected region and the person is healthy.

This qualitative analysis is important for a number of factors:

- **Interpretability:** One of the major drawbacks of many deep learning models is the lack of interpretability. With Grad-CAM, we tried to make our model interpretable and explainable. The generated heat-maps show us insights about how our model makes predictions.
- **Trustworthiness:** From the heatmaps, we can see the important regions of the images that lead to classification decisions. Consequently, we can verify that our model is not making decisions based on inappropriate regions of the radiology images.
- **Possible critical factors:** Our approach can provide new insights and visual indicators about critical factors of COVID-19 disease.
- **Misclassification Cases:** The heatmap provides visual explanation of the misclassified examples. In figure 7 we showed some false positive and false negative cases

Models	COVID-19	Pneumonia	Normal	Overall
COVID-DenseNet	0.87	0.93	0.95	0.94
Resnet50	0.90	0.93	0.91	0.92
InceptionV3	0.84	0.87	0.93	0.90
VGG16	0.77	0.87	0.98	0.93

Table XIII: Comparison of accuracy with other Computer Vision models.

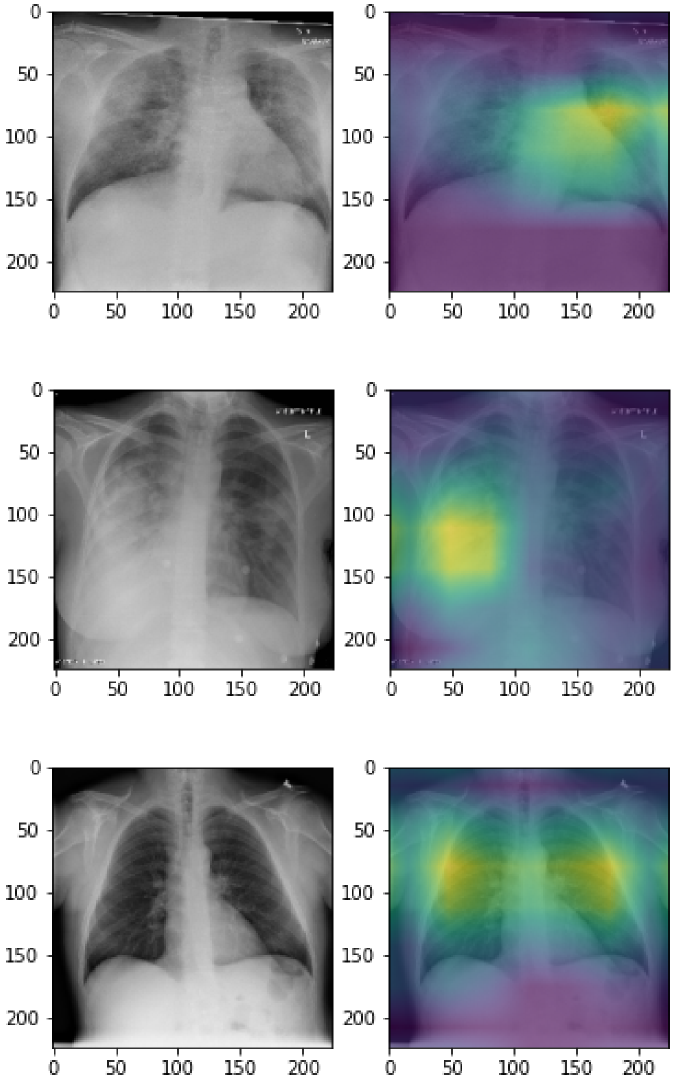

Figure 6: Actual Chest-Xray images along with heatmaps of a COVID-19 affected, a pneumonia affected, and a normal person (top to bottom respectively).

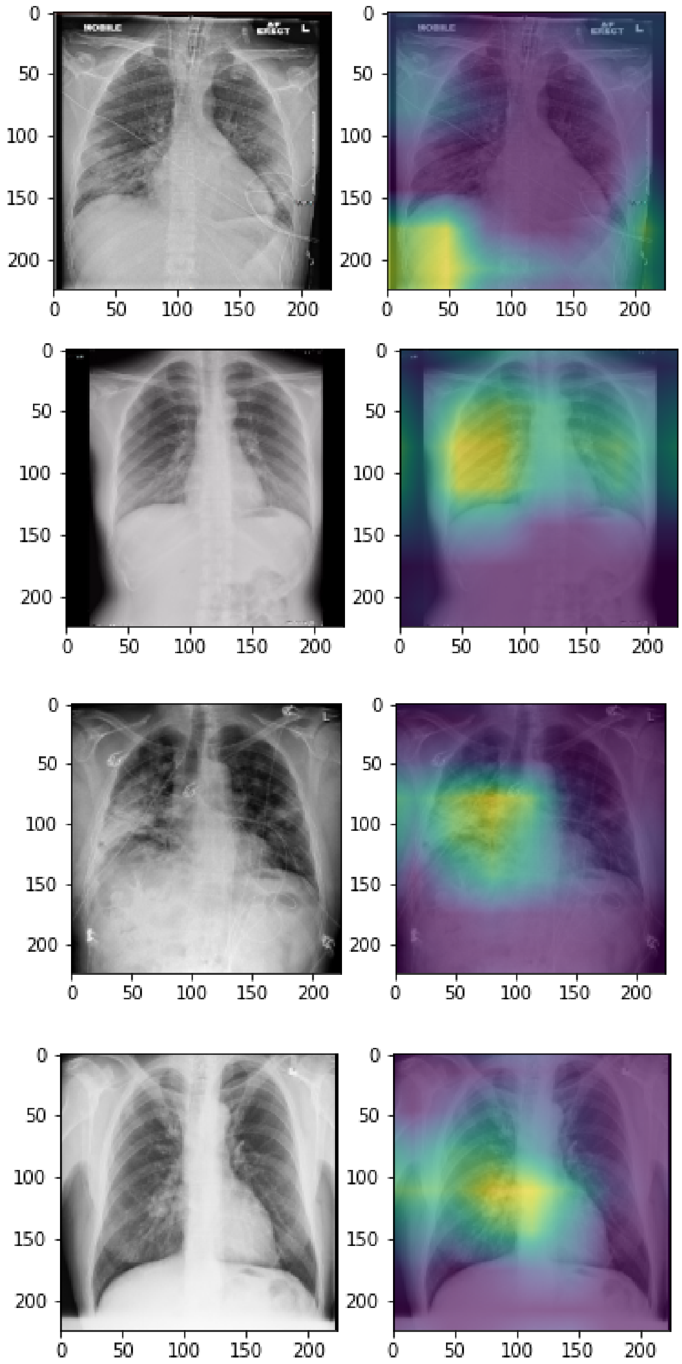

Figure 7: Actual Chest-Xray images along with heatmaps of a COVID-19 misclassified as Pneumonia, COVID-19 misclassified as normal, pneumonia misclassified as COVID-19, and a normal misclassified as COVID-19 (top to bottom respectively).

Weight Initialization	COVID-19	Pneumonia	Normal	Overall
Random	0.87	0.82	0.95	0.90
ImageNet	0.84	0.90	0.96	0.93
CovXNet	0.87	0.93	0.95	0.94

Table XIV: Comparison of accuracy of Densenet with different weight initializations.

Classification	Models	Accuracy
3-class	COVID-DenseNet	0.94
	CovidNet [5]	0.933
	Apostolopoulos2020 [21]	0.947
	DarkCovidNet [26]	0.87
	COVNet [4]	0.93[2]
2-class	COVID-DenseNet	0.96
	Apostolopoulos2020 [21]	0.968
	DarkCovidNet [26]	0.98
	ResNet50 plus SVM [34]	0.95
	ResNet50 [35]	0.98

Table XV: Comparison of accuracy with state-of-the-art methods for 3-class(COVID-19, Pneumonia, and Normal) and 2-class(COVID-19 and Normal) classifications.

of COVID-19 cases. From the visual representation of the critical region we can see the problems of detection associated with different kind of images.

Generally X-ray of healthy patient has no critical region. But our model detected healthy person as COVID-19 positive by finding a critical region in the right lung. The case where normal image was falsely classified as COVID-19 positive, we can see that there is no visible critical region in the lung region. So our model was unable to detect COVID-19.

The distinction of Pneumonia from COVID-19 is a bit complicated than normal images, because in this case there are some critical regions in the lung. The critical region looked quiet similar for both Pneumonia and COVID-19 in some cases. So the model falsely classified the images in this kind of situations.

5. Limitations and future works

Some limitations and future works are listed below.

- **Lack of data:** The biggest limitation of our model is the small number of radiology images of COVID-19 patients. With the limited amount of images, our model worked quite well. But to increase the accuracy and make the model diverse, more images are needed. If more public data is available in future, we can train our model with more data and improve the performance of our model.
- **Different types of pneumonia:** We could work with 14 different types of pneumonia as in [7], which we have merged as one class and train our model. So our model is extendable to these cases easily with slight modification.
- **Progression of COVID-19 disease:** Our dataset contains radiology images of COVID-19 patients of different stages (1-14 days). If more data is available, we can analyze images of same patients at different days to study how COVID-19 gradually develop.

- **Severity of COVID-19 disease:** Currently, COVID-DenseNet is capable of detecting COVID-19 disease from chest radiology images. In future, we are interested in predicting the severity of the detected COVID-19 disease by analyzing the radiology image.

6. Conclusion

In this work, we showed a novel transfer learning-based approach to detect COVID-19. To assure that our model can differentiate COVID-19 radiology images from both healthy persons and pneumonia patients, we performed both 2-class and 3-class classifications. To guarantee the robustness and consistency of our model, we implemented patient-wise 10-fold cross-validation. Moreover, we performed an explainability analysis to interpret and visualize how our model works. Our extensive experiments suggest that COVID-DenseNet can be used effectively for detecting COVID-19 from chest radiology images.

References

[1] The Novel Coronavirus Pneumonia Emergency Response Epidemiology Team. Vital surveillances: The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (covid-19) — china, 2020, February 2020.

[2] Wang. Detection of sars-cov-2 in different types of clinical specimens. *JAMA*, 2020.

[3] Ophir Gozes, Maayan Frid-Adar, Hayit Greenspan, Patrick D Browning, Huangqi Zhang, Wenbin Ji, Adam Bernheim, and Eliot Siegel. Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. *arXiv preprint arXiv:2003.05037*, 2020.

[4] Lin Li, Lixin Qin, Zeguo Xu, Youbing Yin, Xin Wang, Bin Kong, Junjie Bai, Yi Lu, Zhenghan Fang, Qi Song, Kunlin Cao, Daliang Liu, Guisheng Wang, Qizhong Xu, Xisheng Fang, Shiqin Zhang, Juan Xia, and Jun Xia. Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. *Radiology*, 296:200905, 03 2020.

[5] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. 03 2020.

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Weinberger. Densely connected convolutional networks. 07 2017.

[7] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[8] Joseph Paul Cohen, Paul Bertin, and Vincent Frappier. Chester: A web delivered locally computed chest x-ray disease prediction system, 2019.

[9] X Wang, Y Peng, L Lu, Z Lu, M Bagheri, and RM Summers. Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE CVPR*, 2017.

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[11] Website. https://plez.herokuapp.com.

[12] T Huang. Computer vision: Evolution and promise. 10 2020.

[13] Chi Chen. *AN INTRODUCTION TO COMPUTER VISION IN MEDICAL IMAGING*, pages 1–16. 01 2014.

[14] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. pages 1–9, 01 2013.

[15] Saad Albawi, Tareq Abed Mohammed, and Saad ALZAWI. Understanding of a convolutional neural network. 08 2017.

[16] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. In *In the Proceedings of the 25th International Conference on Neural Information Processing Systems*, page 1097–1105, 2012.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. pages 1–9, 06 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] Olaf Ronneberger, Philipp Fischer, Thomas Brox, N Navab, J Hornegger, W Wells, and A Frangi. Medical image computing and computer-assisted intervention–miccai 2015. *Cham: Springer*, pages 234–241, 2015.

[21] Ioannis D. Apostolopoulos and Tzani A. Mpesiana. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2):635–640, April 2020.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[23] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[24] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[25] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[26] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U. Rajendra Acharya. Automated detection of COVID-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine*, 121:103792, June 2020.

[27] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[29] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.

[30] figure1. https://github.com/agchung/Figure1-COVID-chestxray-dataset. Accessed: 2020-04-29.

[31] rnsa. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge. Accessed: 2020-04-29.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

[34] Prabira Kumar Sethy and Santi Kumari Behera. Detection of coronavirus disease (COVID-19) based on deep features. March 2020.

[35] Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, 2020.