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

Md Mohaiminul Islam[*1] Tanveer Hannan[*2] Laboni Sarker[3] Zakaria Ahmed[4]

[1] University of North Carolina at Chapel Hill, `mmiemon@cs.unc.edu`
[2] Ludwig Maximilian University of Munich, `tanveer.hannan@campus.lmu.de`
[3] University of California, Santa Barbara, `labonisarker@ucsb.edu`
[4] Enosis Solutions, `zakaria.ahmed@enosisbd.com`

Abstract. COVID-19 has a severe risk of spreading rapidly, the quick identification of which is essential. In this regard, chest radiology images have proven to be a practical screening approach for COVID-19 affected patients. This study proposes a deep learning-based approach using Densenet-121 to detect COVID-19 patients effectively. We have trained and tested our model on the COVIDx dataset and performed both 2-class and 3-class classifications, achieving 96.49% and 93.71% accuracy, respectively. By successfully utilizing transfer learning, we achieve comparable performance to the state-of-the-art method while using 15x fewer model parameters. Moreover, we performed an interpretability analysis using Grad-CAM to highlight the most significant image regions at test time. Finally, we developed a website that takes chest radiology images as input and detects the presence of COVID-19 or pneumonia and a heatmap highlighting the infected regions. Source code for reproducing results and model weights are available.

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

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 short period and turned into an outbreak [1]. Concerned by the frightening levels of spread and severity, the WHO characterized COVID-19 as a pandemic, 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 spread from person to person very spontaneously, the quick identification and isolation of the affected person is the 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

---

[*] Contributed equally and share the first-authorship of this paper.

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 collecting 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. Many AI-based systems have been proposed to detect COVID-19 from radiography images to remove this bottleneck. 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, the CT image dataset is not publicly available.

On the other hand, X-rays are a more widespread, quicker, and cheaper alternative. Therefore, we choose chest X-ray images in our study. We used the 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 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 is utilizing 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 can differentiate 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%.

Having only 9 million parameters, our model shows competitive performance compared to Apostolopoulos2020 [10] which has 138 million parameters. They

used the same dataset as we use, making this comparison fair and unbiased. We also perform better than some methods [4] [11] [12] who used more data samples than us. It shows the efficiency of our technique which mostly comes from the successful pre-training we incorporate into our training pipeline.

We used Gradient-weighted Class Activation Mapping (Grad-CAM) [13] 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 visualize the most significant features and give insights into the critical factors of COVID-19 patients.

We developed a web application [14], which adapts our model to provide real-time predictions.

The core contributions of this paper are as follows

1. We present a successful transfer learning technique that made use of a large number of existing non-COVID chest radiology images.
2. With a significantly small number of model parameters (15x fewer) and a smaller data set, our method achieves comparable results to the state-of-the-art method.
3. We present an interpretable model that demonstrates the model's validity and explains regions of interest in the patients' lungs.
4. We made our model's weight and training code[5] public, as well as hosted a website, to make it easy to use our method.

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

2.1   Deep Learning in Computer Vision

Computer vision [15] helps us build 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 [16]. Deep neural network(DNN) has an excellent capability in the image classification task [17] and convolutional neural network(CNN, or ConvNet) [18] is one of the most popular classes of DNN. AlexNet [19], VGG [20], Inception [21], ResNet [22], DenseNet [6] are some of the popular convulational networks.

---

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

2.2   Detection of COVID-19

Numerous works exist for detecting COVID-19 from radiography images. Different model architectures have been used for the 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 a projection-expansion-projection-extension (PEPX) design pattern. They combine 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. COVID-Net is very good at detecting COVID-19 infection with high recall. A small percentage of radiology images are incorrectly classified as COVID-19. However, both sensitivity (recall) and positive predictive value (precision) can be increased for other classes. So, there is much more to contribute to properly detecting the COVID-19 from other respiratory infections as they are all very similar. 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. Covent is a 3D deep learning framework(can extract both 3D global and 2D local representative features) and contains a ResNet50 [22] as the backbone. They have used U-net [23] to segment the lung region from the chest radiology images. 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. Paper [10] 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. The author evaluated five CNN models which are VGG19 [24], MobileNet v2 [25], Inception [26], Xception [27], and Inception ResNet v2 [26]. Among these models, MobileNet v2 [25] provided the best results in terms of specificity in their particular datasets. DarkCovidNet is another deep learning model proposed in paper [28]. The author used Darknet-19 [29] model as their base model and designed DarkCovidNet architecture. DarkCovidNet has 17 convolutional layers compared to the 19 convolutional layers in Darknet-19. They used a dataset of 1125 chest X-ray images, which comprises 127 images that were diagnosed with COVID-19, 500 images with pneumonia, and 500 normal images. Khan et al. [30] proposed CoroNet, a deep neural network model which can detect COVID-19 from chest X-ray images. Xception [27] architecture is used as the base model in this proposed model, which is pre-trained on the ImageNet dataset. At the end of the Xception model, they added a dropout layer and two fully-connected layers. The modified model was then retrained using Adam optimizer. CoroNet has 33,969,964 parameters, and transfer learning is used to initialize these parameters to overcome overfitting. Murugan and Goel [11] proposed E-DiCoNet, which is a deep learning network with ELM based classifier (DLN-ELM). A pre-trained CNN-ResNet50 model is used as a starting point. DLN is used for feature extraction, which follows the ELM classifier. The modi-

fied model is then trained with a total of 2700 images which were collected from six publicly available datasets.

This work is more efficient than existing methods because it uses a smaller model architecture and fewer data samples. Despite the limitations, our method successfully detects the various cases with accuracy comparable to larger models or methods using larger data sets.

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 be 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 disciplines. For example, it is a common practice to initialize deep learning models with weights trained on the ImageNet dataset [31] 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 pre-trained models on datasets like ImageNet have already learned various features of additional images. Later, by fine-tuning the pre-trained model on the domain dataset, other domain-specific tasks can utilize the learned features. 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 pre-trained on ImageNet(with 1000 categories of images but with no chest X-ray images). The CheXNet model is 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 3.3(Model architecture).

The complete workflow proposed method is shown schematically in Figure 1. First, we load the pre-trained 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. Our classifier is a fully connected layer with three neurons for 3-class classification (COVID-19, Pneumonia, and Normal). 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

Table 1: Class Distribution.

Augmentation	Normal	Pneumonia	COVID-19
✗	8851	6045	224
✓	8851	6045	11416

radiography images of 13,725 unique patients. This trained COVID-DenseNet model is efficiently used to predict the radiography image class in the testing phase. Finally, a gradient-based localization algorithm (Grad-Cam) [13] 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 the 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 [32], [33], [34]. 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 is only 224. This number is tiny compared to the number of radiology images available for pneumonia infected and healthy persons, 6045 and 8851, respectively. So the data is highly skewed because of the scarcity of images of COVID-19 patients. We augmented only the COVID-19 images in the training set to deal with this unbalanced dataset. Table 1 shows the distribution of the dataset before and after augmentation.

The train-test split ratio is fixed at 0.9. We also stratified the train, validation, and test split to maintain the proportion 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 out. We created nine different images randomly for each category. So each COVID-19 radiology image in the training set has a total of 54 augmentations. The dataset was prepared for 10-fold cross-validation, keeping the class labels' proportions 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 has a variable number of images. So dividing the patients randomly among 10-folds would create an imbalance in the number of images in each fold. So we had to maximize both the number of patients and images for each fold simultaneously. 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 was fed to our model. The only preprocessing was resizing every image to a similar dimension. We used 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 a softmax activation function.

The main building block of DenseNet-121 is DenseBlock [6]. These Dense-Blocks 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, its 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 its input.

$$X_l = H_l([X_0, X_1, ..., X_l - 1]) \qquad (1)$$

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 unique design improves information flow through the network and alleviates the vanishing gradient problem. Moreover, DenseNet enhances feature reuse and parameter efficiency and provides each layer with the network's collective knowledge. 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 2). 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:

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

In our model, the final layer of the Dense-121 is a global average pooling layer that generates the features from the input image. The classifier uses these features to make the final prediction. We used a fully connected layer for the classifier, 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
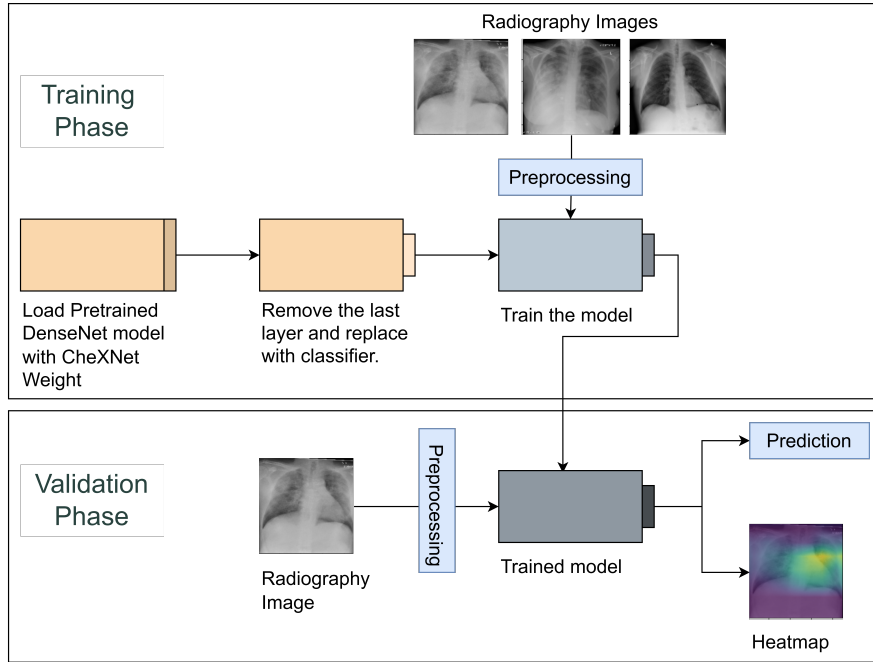
Fig. 1: Schematic diagram of the complete workflow.

over the predicted output classes. The equation of the softmax function can be written as follows:

$$\sigma(\vec{z_i}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$$ (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 pre-trained 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 following equation.
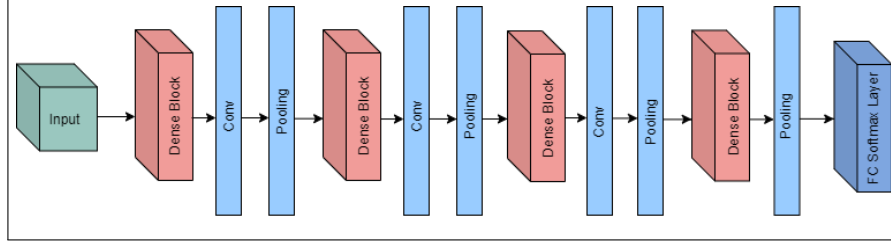
Fig. 2: DenseNet-121 with 4 dense blocks and 3 transition layers.

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

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 [35] was used to update the model weight θ. The weight update equation can be written as follows.

$$\begin{aligned} \theta_t &= \theta_{t-1} - \alpha\frac{m_t}{\sqrt{v_t} + \epsilon}\\ m_t &= \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t\\ v_t &= \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2 \end{aligned} \qquad (5)$$

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 find the minimum in the loss surface more efficiently. We used early stopping with patience 5, which means if validation set performance does not improve for five 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. The detailed training dynamics with loss and accuracy are depicted in Figure 3.

3.5   Prediction and Heatmap generation

In Section 3.4, 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
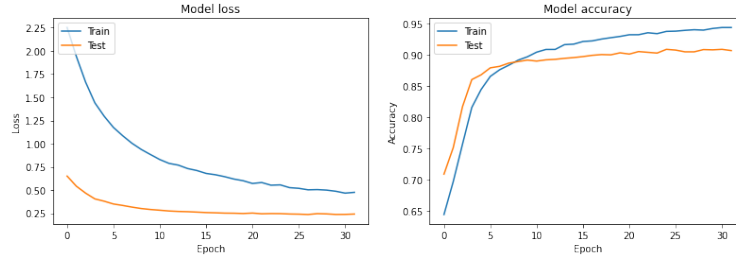
Fig. 3: Accuracy vs epoch and loss vs epoch for train and validation set (Experiment 1).

(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 that highlights the significant regions in the input image that contributed most to a particular prediction. The heatmap can help doctors identify the critical areas of an affected patient's chest, leading the model to identify him as a COVID-19 affected person.

We used Gradient-weighted Class Activation Mapping (Grad-CAM) [13] to generate the heatmap. This heatmap is a coarse localization map that highlights the critical 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 lose the spatial information. Therefore, last CNN is a good choice that captures 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 dimensions to generate the neuron importance weights for the target class.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\delta y^c}{\delta A_{ij}^k} \tag{6}$$

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.

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 emphasizes the features that positively influence the final prediction.

$$L_{Grad-CAM}^{c} = ReLU \left(\sum_{k} \alpha_{k}^{c} A^{k} \right) \qquad (7)$$

Here, $L_{Grad-CAM}^{c}$ is the generated heatmap, α_{k}^{c} 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 4. Figure 4 shows the actual Chest-Xray images along with heatmaps of a COVID-19 affected, pneumonia affected, and a normal person. We can see that our model is mainly emphasizing the lung areas in detecting COVID-19 or pneumonia.

4   Results and Discussion

Because the dataset of COVID-19 cases is limited, we performed both 2-class classification (COVID-19 and non-COVID-19) and 3-class classification (COVID-19, Pneumonia, and Normal) to ensure the performance of our model. 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.

We calculated each experimental setup's test accuracy, precision, recall, and f-score. 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. The recall is the fraction of test instances of a class that has been correctly predicted, whereas precision is the fraction of correctly classified objects assigned to the class. F-score is simply the harmonic mean of these two values. Equation 8 defines the performance matrices.

$$\begin{aligned} recall &= \frac{TP}{TP + FN} \\ precision &= \frac{TP}{TP + FP} \\ fscore &= \frac{2 * recall * precision}{recall + precision} \end{aligned} \qquad (8)$$

Here, TP, FN, and FP are true positives, false negatives, and false positives. For multi-class prediction, f-score can be computed in two ways: micro-average f-score and macro-average f-score. In the micro-average f-score, the larger classes get more weight and give the same value as accuracy itself. 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   Main results

We have encountered several state-of-the-art methods developed for detecting COVID-19. The accuracy comparison with these methods is shown in Table 2.

Table 2: 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.

	2-Class ↑	3-Class ↑	Params(M)↓	Cases
COVID-DenseNet	**0.94**	**0.96**	9	224
Apostolopoulos2020 [10]	0.94	0.96	138	224
E-DiCoNet [11]	0.94	-	23	900
COVIDNet [5]	0.93	-	11.7	266
COVNet [4]	0.93^6	-	23	1165
CoroNet [30]	0.89	-	34	290
DarkCovidNet [28]	0.87	0.98	1.2	127
ResNet50 plus SVM [36]	-	0.95	23	50
ResNet50 [12]	-	0.98	23	341

We achieved the same performance as Apostolopoulos2020 [10] having only 9 million parameters, whereas their model has 138 million parameters. Moreover, we perform better with a much smaller number of train data samples for COVID cases than E-DiCoNet [11], COVNet [4], and CoroNet [30]. Our model can perform better than most state-of-the-art methods with a small model size and limited training data. The successful and efficient utilization of transfer learning is the key to this improved performance with the constraints mentioned.

4.2   Detailed results

Three class classification In this experiment, we performed a 3-class classification (COVID-19, Pneumonia, and Normal). We split our dataset into train, validation, and test set in an 80%-10%-10% ratio. There was no common image among the three sets, and each set performed augmentation separately. Results are shown in Table 3(a)(b).

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 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 than 90% for pneumonia and normal and 87% for COVID-19. These results indicate our model can classify COVID-19, pneumonia, and normal chest radiography images with high confidence.

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 3(c)(d). 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 are also 96%. We also analyzed accuracy, precision, recall, and f-score separately for COVID-19 and Non- COVID-19

Table 3: Detailed Results.

(a) Three-class Classification.

	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

(b) Three-class Confusion Matrix.

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

(c) Two-class Classification.

	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

(d) Two-class Confusion Matrix.

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

and all matrices are more than 90%. These results ensure that this model can serve very well if one only wants the successful detection of COVID-19.

4.3 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 dataset 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 4(a) 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 model's performance does not deviate too much even if we change the train and test instances. So the model is robust and can work exceptionally well with new images.

4.4 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 4(b). The overall performance increased with weight initialization from ChestXNet.

4.5 Comparison with standard computer vision models

We compared our model with other state-of-the-art computer vision models. Particularly, we compared inception-v3, resnet50, and VGG-16 with the same setup as the three-class classification. Initial weights were set to the trained models

Table 4: Ablation studies.

(a) Patient-wise cross-validation.

	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

(b) Accuracy with Different weight.

	COVID-19	Pneumonia	Normal	Overall
CovXNet	**0.87**	**0.93**	**0.95**	**0.94**
ImageNet	0.84	0.90	0.91	0.90
Random	0.80	0.82	0.95	0.87

on ImageNet, 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, a transfer learning approach. The comparative performance is shown in Table 5. Overall accuracy for COVID-DenseNet is 94%, while accuracy for resnet50, inception-v3, and VGG-16 are 90%, 89%, and 88%. COVID-DenseNet has higher accuracy for Pneumonia and Normal cases also. The results indicate that our model outperforms other computer vision models.

Table 5: Comparison of accuracy with other Computer Vision models.

	COVID-19	Pneumonia	Normal	Overall
COVID-DenseNet	**0.87**	**0.93**	**0.95**	**0.94**
Resnet50	0.84	0.93	0.91	0.90
InceptionV3	0.84	0.87	0.93	0.89
VGG16	0.77	0.87	0.93	0.88

4.6   Qualitative Analysis

As described in section 3.5, we used Gradient-weighted Class Activation Mapping (Grad-CAM) [13] to analyze our model's output. It produces a coarse localization map highlighting the significant regions in the input image to predict. From the generated heatmap, we can approximately localize the possible affected regions. Figure 4 shows the actual Chest-Xray images along with heatmaps of a COVID-19 affected, pneumonia affected, and a normal person. We can see that our model is mainly emphasizing 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, the heatmap of a normal person shows that our model is not emphasizing any particular region. This indicates there is no affected region, and the person is healthy.

This qualitative analysis is important for several 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 heatmaps show us insights into how our model makes predictions.
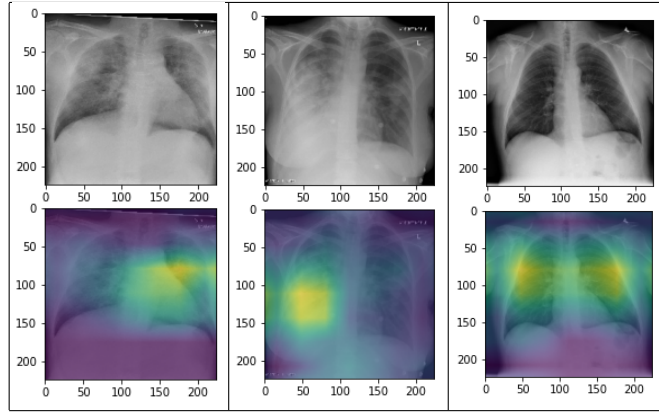
Fig. 4: Actual Chest-Xray images along with heatmaps of a COVID-19 affected, a pneumonia affected, and a normal person (left to right respectively).

– **Trustworthiness:** From the heatmaps, we can see the critical 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 the critical factors of COVID-19 disease.
– **Misclassification Cases:** The heatmap provides a visual explanation of the misclassified examples. In figure 5 we showed some false positive and false negative cases of COVID-19 cases.

   Generally, an X-ray of a healthy person has no critical region. But our model detected a healthy person as COVID-19 positive by finding a critical region in the right lung. In the case where the 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 more complicated than normal images because, in this case, there are some critical regions in the lung. In some cases, the critical region looked quite similar for both Pneumonia and COVID-19. So the model falsely classified the images in this kind of situation.

5   Limitations and future works

Some limitations and future works are listed below.

– **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.
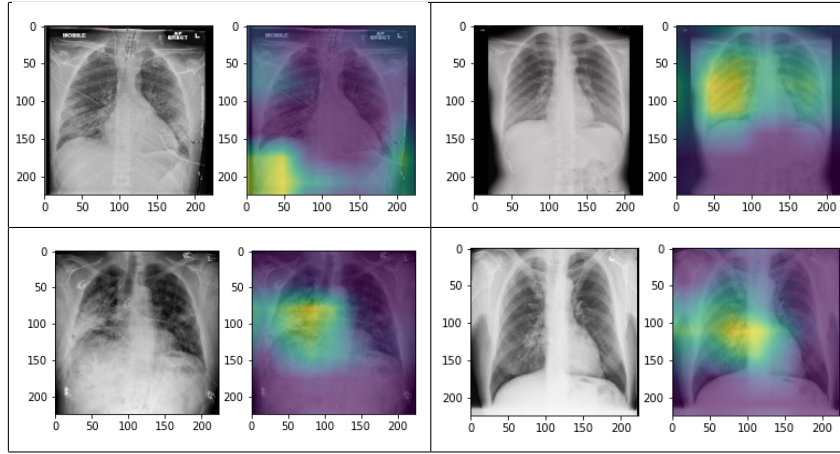
Fig. 5: Actual Chest-Xray images along with heatmaps of a COVID-19 misclassified as Pneumonia (Top Left), COVID-19 misclassified as normal (Top Right), pneumonia misclassified as COVID-19 (Bottom Left), and a normal misclassified as COVID-19 (Bottom Right).

– **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 the same patients on different days to study how COVID-19 gradually develops.
– **Severity of COVID-19 disease:** Currently, COVID-DenseNet is capable of detecting COVID-19 disease from chest radiology images. In the future, we are interested in predicting the severity of the detected COVID-19 disease by analyzing the radiology image.

6   Conclusion

This work shows a novel transfer learning-based approach to detect COVID-19. We perform competitively with state-of-the-art methods with notably a smaller model size and fewer data samples. 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 and efficiently 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. 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.
11. R. Murugan and Tripti Goel. E-diconet: Extreme learning machine based classifier for diagnosis of covid-19 using deep convolutional network. *Journal of Ambient Intelligence and Humanized Computing*, 12(9):8887–8898, Sep 2021.
12. Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, 2020.
13. 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.
14. Website. https://plez.herokuapp.com.
15. T Huang. Computer vision: Evolution and promise. 10 2020.
16. Chi Chen. *AN INTRODUCTION TO COMPUTER VISION IN MEDICAL IMAGING*, pages 1–16. 01 2014.
17. Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. pages 1–9, 01 2013.

18. Saad Albawi, Tareq Abed Mohammed, and Saad ALZAWI. Understanding of a convolutional neural network. 08 2017.
19. 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.
20. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.
21. 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.
22. 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.
23. 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.
24. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
25. 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.
26. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.
27. François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.
28. 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.
29. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
30. Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasir Bhat. Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, 196:105581, 2020.
31. 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.
32. Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.
33. figure1. https://github.com/agchung/Figure1-COVID-chestxray-dataset. Accessed: 2020-04-29.
34. rnsa. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge. Accessed: 2020-04-29.
35. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
36. Prabira Kumar Sethy and Santi Kumari Behera. Detection of coronavirus disease (COVID-19) based on deep features. March 2020.