



WARSAW UNIVERSITY OF TECHNOLOGY
**Faculty of Mathematics
and Information Science**



Artificial Intelligence Fundamentals

**Project of
COVID-19 Classification Detection System Based on Dense
Convolutional Network with SENet Module**

Done by:

Amir Ali & Nader Tavana

Computer Science and Information Systems

02 June 2021

Responsibilities

Name	Tasks
Amir Ali	<ul style="list-style-type: none">● System Architecture● Data Generation● Data Preprocessing● Model Architecture● Model Implementation● Result and Discussion
Nader Tavana	<ul style="list-style-type: none">● Introduction● Related Work● Conclusion● Presentation

Table of Contents

- Abstract
- 1. Introduction
- 2. Some Existing Solution
- 3. Methodology
 - 4.1 Data Generation
 - 4.2 Data Preprocessing
 - 4.3 Model Architecture
 - 4.4 Model Implementation
 - 4.5 Prediction and Heatmap generation
- 4. Result and Discussion
 - 5.1 Two-class Classification
 - 5.2 Three-class Classification
 - 5.3 Compare Accuracy with other Model
- 5. Conclusion
- 6. Future Work
- References

Abstract

Coronavirus known as Covid-19 starts from Wuhan city of China and affects rapidly around all over the world. In March 2021, According to the Statistic of WHO approximately 120M cases reports and deaths are 2.66M which is extremely frightened. It has caused a devastating effect on both community health and the worldwide economy. This study proposes a computer vision method that can detect COVID-19 from chest X-ray images. For this purpose, we will use the Dense Convolutional Neural Network (DenseNet) as the basic building block module for the classification of COVID-19 Computed Tomography (CT) Chest Images.

1. Introduction

COVID-19 is a type of virus. It is called Novell because it is a new strain of the virus and its combination with the number 19 since it appeared in 2019. COVID-19 is a transferable virus caused by SARS-CoV-2. The most common symptoms of the virus are fever, dry cough, tiredness, difficulty breathing, and Chest Pain.

Coronavirus first case reports in November 2019 in Wuhan city of China and affected thousands of peoples. Then rapidly spread around all over the world. Now currently 2.66M cases Reports and Death around 2.6M.

The Coronavirus is first and foremost reach between people during close contact through sneezing, handshaking, eating, etc. The virus approaches the cell membrane which very dangerous for the human body. 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 COVID19 [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 COVID 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.

2. Some Existing Solution

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 convolutional 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 layer. 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 performance.

Numerous works have been done in detecting COVID-19 from radiography images. Different model architectures have been used for accurate detection of the disease. COVIDNet [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-projectionextension (PEPX) design pattern (Figure 1). They utilized a human-machine collaborative design strategy. This strategy

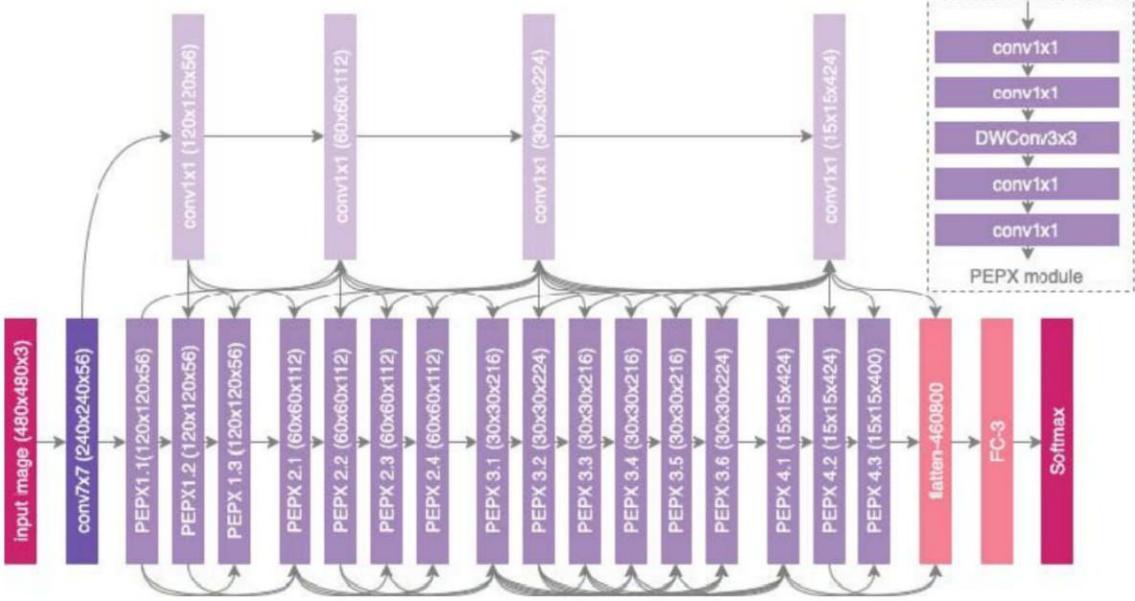


Figure 1: COVID-Net [5] Architecture

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

Table 1: Sensitivity (Recall) of COVID-Net [5]

Normal	Bacterial	Non-Covid Viral	COVID Viral
73.9	93.1	81.9	100

Table 2: Precision of COVID-Net [5]

Normal	Bacterial	Non-Covid Viral	COVID Viral
95.1	87.1	67.0	80.0

From Table 1and 2, 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 3 gives us an overview of the performance of their model, which seems very promising but not for public use.

Transfer learning is a technique where the knowledge gained from solving a specific problem is transferred to solve a different but similar problem.

Table 3: Sensitivity (Recall) & Specificity of COVNet [4]

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

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 4 presents a summary of their datasets.

Table 4: 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 5 are 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.

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

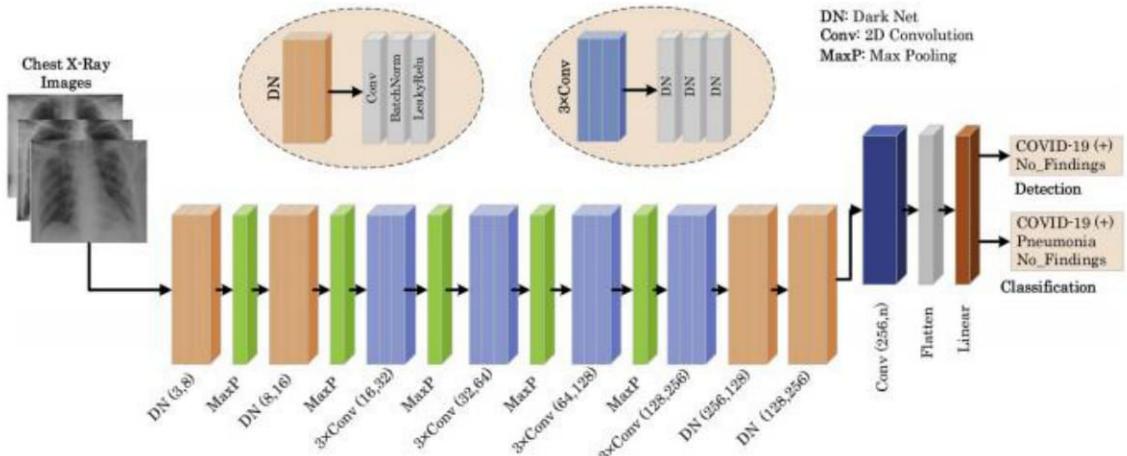


Figure 2: Dark Covid Net [26] Architecture

Dark Covid Net 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 5: Results of MobileNet v2 [23] on Dataset 2

Accuracy		Sensitivity	Specificity
2 Class	3 Class		
0.96	0.94	0.98	0.96

Normal. The result of their experiment is presented in Table 6.

Table 6: Experiment results of Dark Covid Net [26]

Classification	Sensitivity	Specificity	Precision	F-1 Score	Accuracy
2 class	0.95	0.95	0.98	0.96	0.98
3 class	0.85	0.92	0.89	0.87	0.87

From the table, it is clear that Dark Covid Net is very good at detecting COVID-19 for 2-class classification. But there are rooms 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 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 domains. For example, it is 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 datasets 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]. 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 4 shows the distribution of the dataset before and after augmentation.

Table 6: Class Distribution

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 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, it's own featuremaps are used as inputs for all subsequent layers. So, for lth layer, features maps of all preceding layers X_0, X_1, \dots, X_{l-1} are concatenated and used as its input.

$$X_l = H_l ([X_0, X_1, \dots, X_{l-1}]) \quad (1)$$

Here H_l represents the lth layer, X_l is the output of the lth layer, and $[X_0, X_1, \dots, X_{l-1}]$ 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 batch normalization, a convolution, and an average pooling layer. To increase nonlinearity ReLU activation function is used in DenseNet, which can be described as:

$$\text{ReLU}(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2)$$

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:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3)$$

Here, 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 following equation:

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

Here, n is the number of training samples, k is the number of classes, θ is the model parameter.

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

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

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

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

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 performance does not improve for 5 epochs,

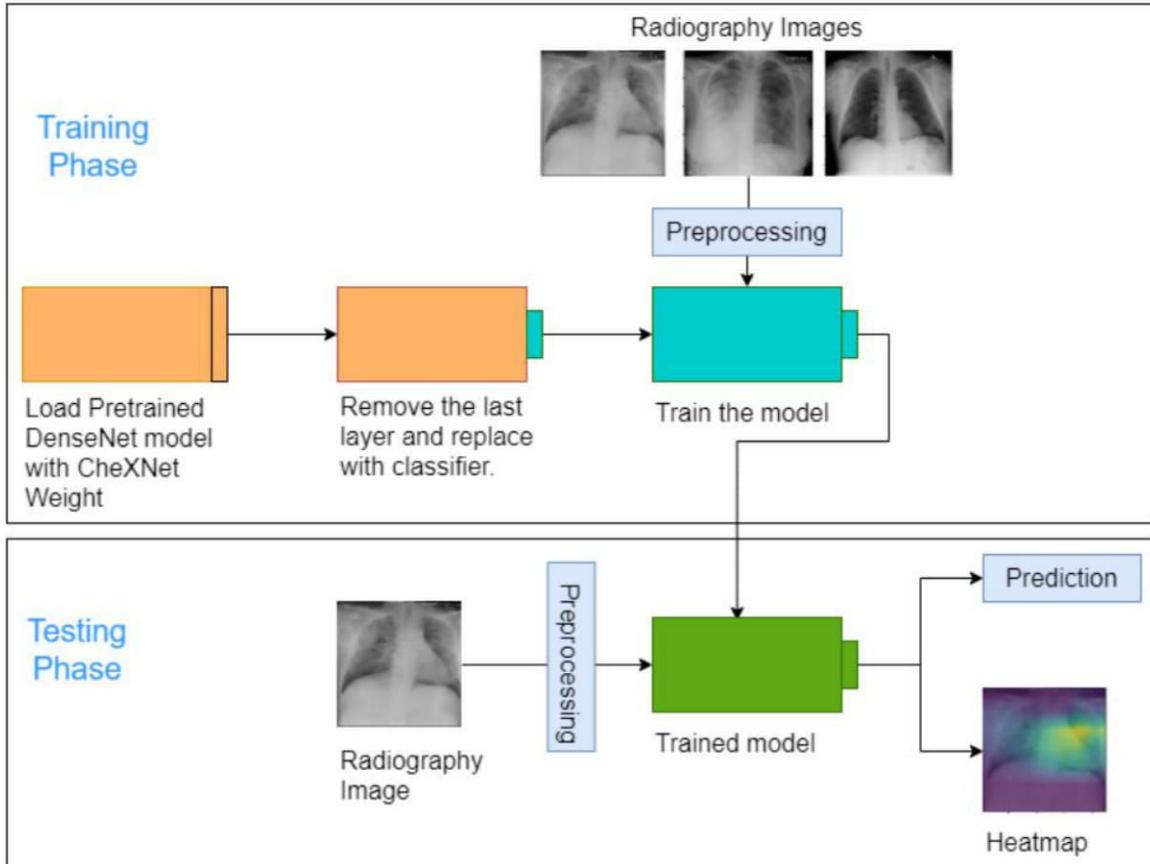


Figure 3: Schematic diagram of the complete workflow

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 this Section, 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.

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

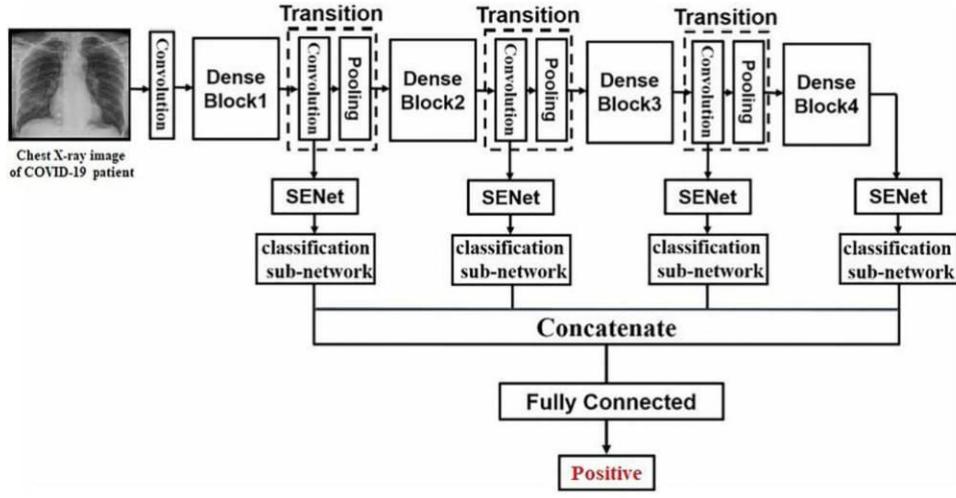


Figure 4: DenseNet-121 with 4 dense blocks and 3 transition layers.

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.

$$L_{Grad-CAM}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right) \quad (9)$$

4. Result 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 nonCOVID-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 decisionmaking behavior of our model to ensure interpretability and trustworthiness.

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

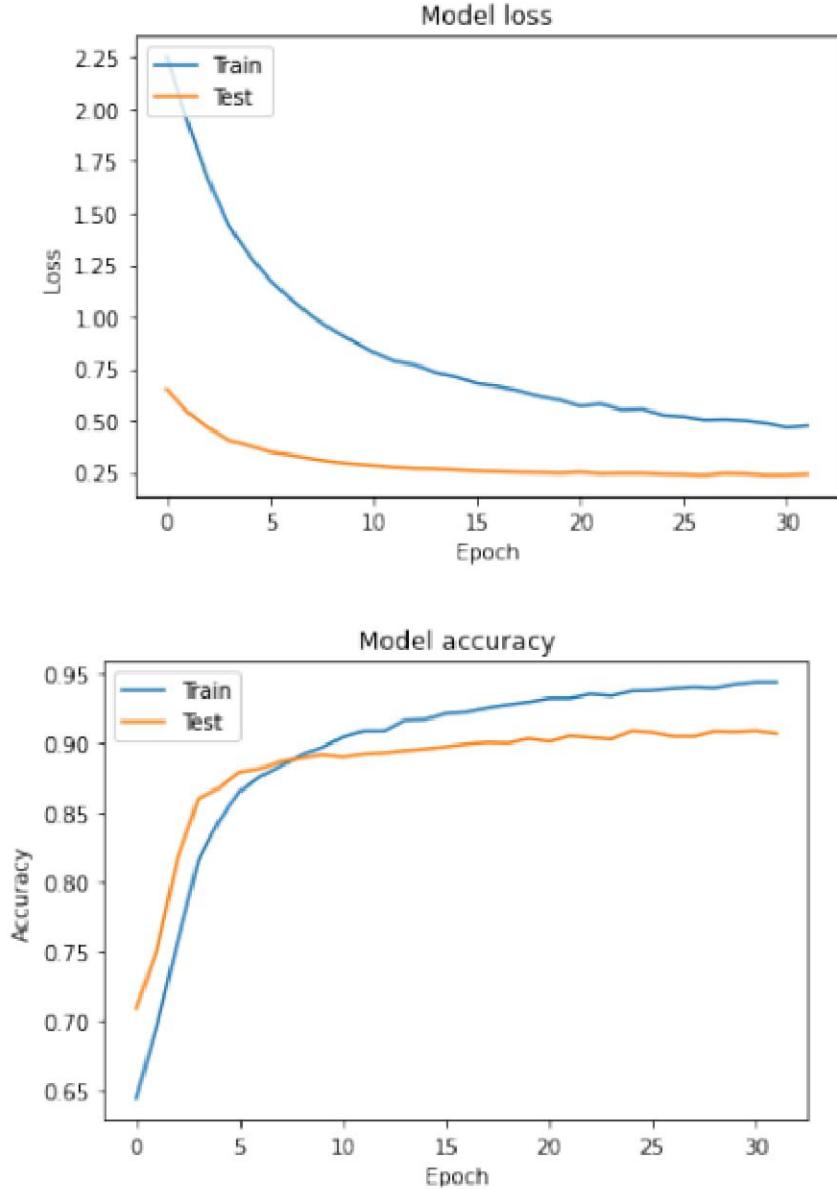


Figure 5: Accuracy vs epoch and loss vs epoch for train and validation set

4.1. 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 fscore 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 than 90%. These results ensure that, this model can serve very well in case one only wants the successful detection of COVID-19.

Table 7: Two-class Classification Result

Class	Accuracy	Precision	Recall	F-Score
Overall	0.96	0.96	0.96	0.96
Covid-19	0.93	0.90	0.94	0.92
Normal	0.96	0.97	0.97	0.97

4.2. 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 than 90% for pneumonia and normal and 87% for COVID19. These results indicate our model is capable of classifying COVID-19, pneumonia, and normal chest radiography images with high confidence.

Table 7: Three-class Classification Result

Class	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	0.94	0.94	0.96	0.95

4.3. Compare Accuracy with other Computer vision model

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.

Table 8: Compare Accuracy with other model

Classification	Model	Accuracy
Three Class	DenseNet	0.94
	CovidNet [5]	0.93
	DarkCovidNet [26]	0.87
	COVNet [4]	0.93
Two Class	DenseNet	0.96
	DarkCovidNet [26]	0.98
	ResNet50 plus SVM [34]	0.95
	ResNet50 [35]	0.98

5. 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.

6. Future Work

Currently, COVIDDenseNet 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.

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, MichaelCogswell, Abhishek Das, RamakrishnaVedantam, DeviParikh, and DhruvBatra. 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 coro navirus disease (covid-19) using x-ray images and deep convolutional neural networks, 2020