Novel Application of Deep Learning in Detection of COVID-19

https://youtu.be/Zxu9vf93VLs

Kefei Wang, Wuguo Chen, Yi Gu, Xidu Qiu
George Institute of Technology, Atlanta, GA, USA

Abstract

This project introduces the use of X-ray imagery for automated classification of emerging COVID-19 using deep learning. We used a novel approach of applying a convolutional neural network (CNN) including ResNet50, DenseNet121, VGG19 and InceptionV3 on X-ray imagery. Both binary and multiple classification was then performed to determine whether COVID-19 infection was present or not, when compared to other conditions. We found that Inspection V3 performs the best for binary data (99%） and DenseNet performs the best for multi-classes data (77%).

1. Introduction

Coronavirus Disease 2019 (COVID-19) is a very infectious disease caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), that has resulted in an ongoing pandemic worldwide through person-to-person transmission. The typical clinical symptoms include fever, cough, shortness of breath and other vital signs. It is reported that the cumulative number of COVID-19 cases worldwide are more than 30 million and the number of deaths has already exceeded 1 million by the 1st of October 2020 [1]. Currently, real-time RT-PCR is one of the most widely used laboratory methods for detecting the COVID-19. Antibody tests are also a useful complementary method for diagnosis.

2. Problem Description

Due to limitations of specificity or sensitivity, laboratory methods sometimes produce unexpected false results. Therefore, chest radiological imaging such as computed tomography (CT) and X-ray has been officially applied in clinical diagnosis and play key roles in early infection stage and even during treatment of this disease [2]. Moreover, chest X-rays are the preferred initial imaging modality when pneumonia is suspected as the radiation dose is lower than the radiation dose of chest CT scans. However, there are many similar features between medical images of COVID-19-infected lung tissue and pneumonia which is caused by other viral infections such as Influenza A. distinguishing medical images is challenging for radiologist experts [3].

3. Related Work

DeepLearning algorithms have shown unprecedented success in the reliable analysis of medical images. Properly implemented, these approaches can be scalable, accurate and convenient to implement in clinical diagnosis, such as classification of medical images with highly similar features [4-9]. In this study, we used CNN to train models to classify X-ray images. Since the invention of AlextNet[17] in 2012, lots of CNNs have been developed and shown significant performance on image classification problems. We adopted some of these neural networks in our project, for example; ResNet50, Vgg, DenseNet121 and DenseNet.

- **ResNet50** ResNet is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was that it allowed us to train extremely deep neural networks with 150+ layers successfully. ResNet uses a skip connection approach to add the output from an earlier layer to a later layer. This helps it mitigate the vanishing gradient problem. ResNet50 is a convolutional neural network that is 50 layers deep, consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet50 has over 23 million trainable parameters. [10].
- **DenseNet121** DenseNet is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each dense block). For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers. DenseNet121 has 121 layers but it still makes the training of deep learning models tractable. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters [11].
- **VGG19** VGG is a convolutional neural network model which improves on AlexNet by replacing large kernel-sized filters. It focuses on having convolution layers of 3x3 filter with a stride 1 and always uses the same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 fully connected layers followed by a softmax for output. The VGG19 has 19 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters [12, 13].
- **InceptionV3** It is a widely-used image recognition model which is a convolutional neural network with 48 layers. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during ImageNet Recognition Challenge [14]. It is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

In addition to the neural networks above, and given the fact that we have a relatively small dataset (1196 images), we also applied transfer learning in our model. Transfer learning is an approach where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems[18].

4. Experimental Setup

4.1 Data

COVID-19 X-ray images were derived from a public database developed by Cohen JP [14], which collects X-ray and CT images of patients from different public sources. We extracted a total of 196 X-ray images from the database that contained confirmed infections with COVID-19. 500 pneumonia X-ray images which excluded COVID-19 infection were used as a positive control along with 500 normal X-ray images from healthy people were used as a negative control. These two control datasets come from the National Institute of Health (NIH) Chest X-ray dataset [15]. All the datasets are in posteroanterior (PA) view.

4.2 Software and Hardware

For our ETL pipeline, we used an Amazon Web Service (AWS) EMR cluster, version 5.32.0 with Apache Spark 2.4.5 via PySpark. We used a Spark cluster with one driver (on-demand m5.xlarge instance) and seven worker nodes (also on-demand, m5.2xlarge). To train CNNs on image data requires substantial resources, so initial setup was performed on a PC with a GPU. This PC had an Intel(R) i7-7700 CPU, 16 GB RAM as well as an NVIDIA GeForce GTX 1080 GPU. For final training and tuning, we used an AWS EC2 p3.8xlarge instance with 4 NVIDIA Tesla V100 GPUs, 64 GB of GPU memory. Beside standard Spark libraries, we will also used Python libraries, including, but not limited to Numpy, Pillow, TensorFlow, Keras, Scikit-learn, Pandas, Scipy, Maplotlib and OpenCV-Python (cv2).

4.3 Approach

For this project, we first created a Spark ETL job to preprocess the image data, then we trained the Convolutional Neural Network models on a separately GPU-equipped machine (either our personal desktops or AWS EC2 instances).

The ETL Spark job ingested images from all three data types (COVID-19, pneumonia, and normal), transformed the images to RGB channel ordering, resized the images to 224*224 pixels, scaled pixels to the range from 0 to 1, flattened/reshaped the matrix to one-dimensional arrays, saved it to a Spark DataFrame, and tagged with labels. Finally, we stored the data to S3 for our Convolutional Neural Network models.

On a separate machine, we read in the preprocessed, flattened CSVs and reshaped the image data back to a 224*244 numpy array. The label data was encoded to a one-vs-all fashion. Images and label data were then split into training and test sets. We reserved 80% of data for training and 20% of data for testing. Data augmentation with a random rotation of 15 degrees, either clockwise or counterclockwise, was set to ensure the models generalized.

Our models were trained in two phases (not counting ImageNet application). In the first phase, we applied transfer learning and used the features that the initial CNN models (ResNet50, DenseNet121, VGG19, InceptionV3) leveraged from the ImageNet database [19]. Those features were frozen and would not be updated in this phase. On top of the frozen layers, we added some trainable, fully connected layers which turned the old features into predictions on our dataset. Each CNN model now has different top layers and different dropout rates depending on the output shape of pre-trained layers. Please see Fig. 1 for the structure of each CNN model.

In the second phase, we used the same structure as phase one, but un-froze the pre-trained layers and set all the parameters trainable, then trained the whole model with a smaller learning rate. This further improved our model performance. For both phases, we used 30 epoches and batch size of 3. In phase one, we used 0.001 as the learning rate and 0.00001 in phase two.

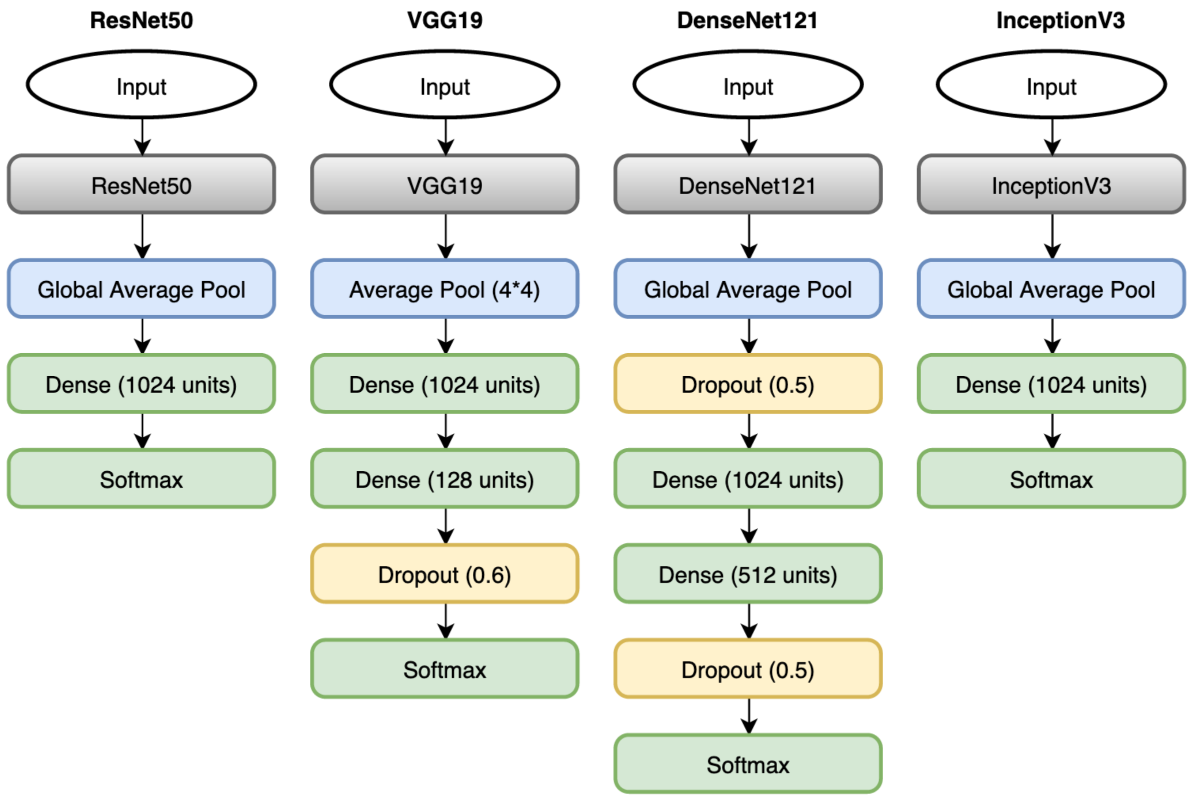


Fig. 1 CNN model diagram

4.4 Evaluation metrics

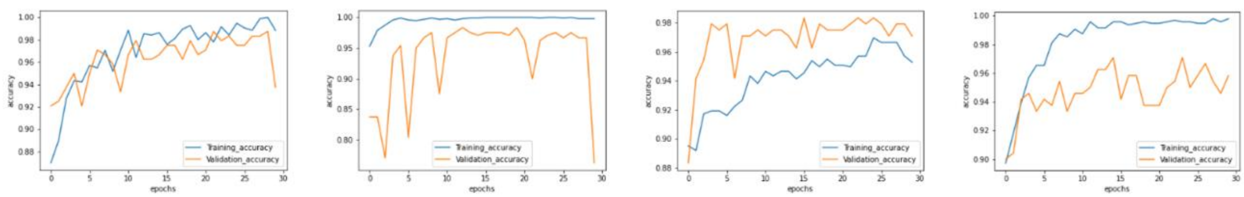
With the above models, we generated training and validation accuracy plots from the training output and evaluated their performance on a validation set of images. Accuracy, precision, recall and F1 score, along with the confusion matrix, were derived from the test dataset.

In addition, for intuitive visualization, we created heatmaps using a Gradient-weighted Class Activation Mapping (Grad-Cam) approach, which applies the gradients of any target concept flowing into the final convolutional layer of model, and produces a coarse localization map highlighting the import regions in the image for predicting the concept [20].

5. Experimental Results

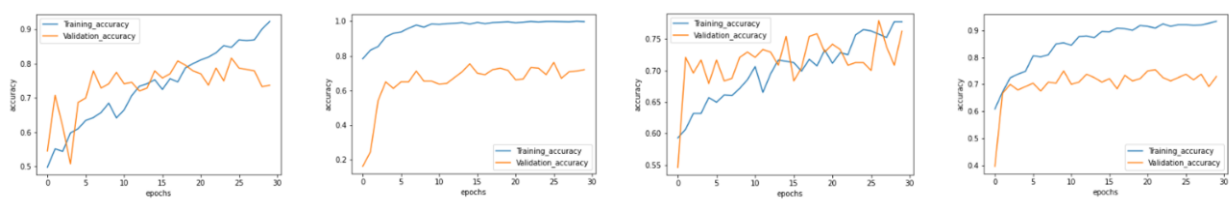
At first, we performed binary classification between X-ray chest images of COVID-19 and non-COVID19 (pneumonia and normal) by using CNN models as above. For this, we evaluated the classification accuracy and confusion matrix for the CNN pre-trained models (VGG19, ResNet50, DenseNet121, and InceptionV3) after fine tuning. We found that the accuracy of the InceptionV3 network was the highest among the four CNN models, close to 0.99. For the other three models, we found the accuracies are also high except that the stability was weaker in the ResNet50 network (Fig. 2).

Furthermore, we performed multiple classification evaluations for images of COVID-19, pneumonia and healthy tissue imagery using similar methods. To our surprise, the accuracy of all the validation data was a little lower than binary classification (about 0.7). For Vgg19 and DenseNet121, the accuracies of the training dataset increased while the validation data fluctuated around 0.75 during the period of epochs. For ResNet50 and InceptionV3, the accuracies of training (about 0.95) are higher than validation datasets (about 0.65), indicating model overfit.(Fig. 3).



(a). Vgg19 (b). ResNet50 (c). DenseNet121 (d). InceptionV3

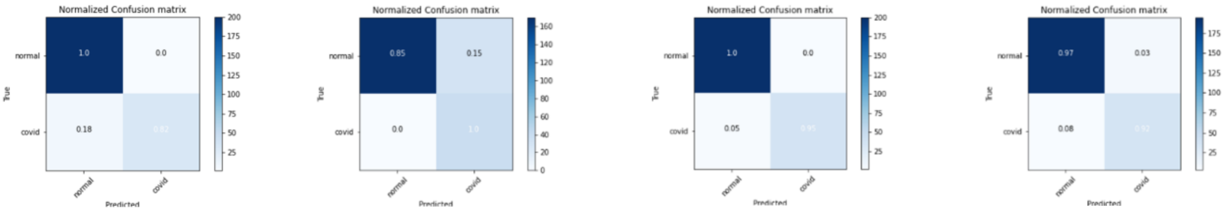
Fig. 2 The learning curve accuracy for binary classification



(a). Vgg19 (b). ResNet50 (c). DenseNet121 (d). InceptionV3

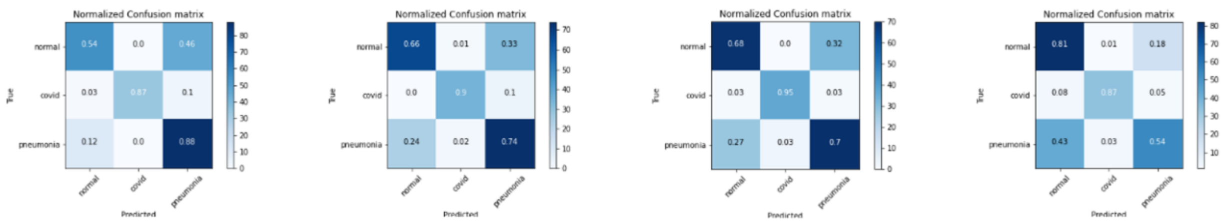
Fig. 3 The learning curve accuracy for multiple classification

From the results of the confusion matrix, all the models achieve perfect binary classification (Fig. 4). Among the four models, Inception V3 reached the highest testing accuracy of 99%. However, for multiple classification, the accuracies are lowerDenseNet121 has the highest accuracy among them, which might be related to the variances existing within three datasets (Fig. 5). Finally, we derived the results from the confusion matrix as shown in Table 1 & 2 which had similar trends.



(a). Vgg19 (b). ResNet50 (c). DenseNet121 (d). InceptionV3

Fig. 4 The confusion matrix for binary classification.



(a). Vgg 19 (b). ResNet50 (c). DenseNet121 (d). InceptionV3

Fig. 5 The confusion matrix for multiple classification

Table 1: Results from different models-Binary Classification.

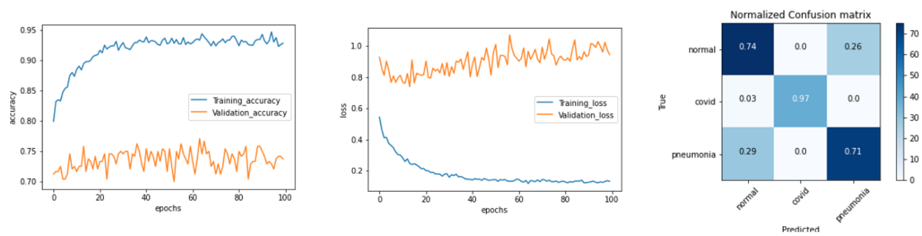
	Vgg19				**ResNet50**				**DenseNet121**				**InceptionV3**			
	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support
no cov.	0.97	1.00	0.98	201	1.00	0.85	0.92	201	0.98	0.97	0.98	201	0.99	1.00	0.99	201
covid19	0.97	0.82	0.89	39	0.56	1.00	0.72	39	0.86	0.92	0.89	39	0.97	0.95	0.96	39
acc.	0.97	0.97	0.97	0.97	0.87	0.87	0.87	0.87	0.96	0.96	0.96	0.96	0.99	0.99	0.99	0.99
ma. avg	0.97	0.91	0.93	240	0.78	0.92	0.82	240	0.92	0.95	0.93	240	0.98	0.97	0.98	240
wei. avg	0.97	0.97	0.97	240	0.93	0.87	0.88	240	0.96	0.96	0.96	240	0.99	0.99	0.99	240

Table 2: Results from different models-Multiple Classification.

	Vgg19				**ResNet50**				**DenseNet121**				**InceptionV3**			
	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support	prec.	recall	f1-sc.	support
normal	0.81	0.54	0.65	101	0.74	0.66	0.70	101	0.64	0.81	0.72	101	0.71	0.68	0.70	101
covid19	1.00	0.87	0.93	39	0.92	0.90	0.91	39	0.89	0.87	0.88	39	0.93	0.95	0.94	39
pneumonia	0.64	0.88	0.74	100	0.67	0.74	0.70	100	0.73	0.54	0.62	100	0.68	0.70	0.69	100
acc.	0.74	0.74	0.74	0.74	0.73	0.73	0.73	0.73	0.71	0.71	0.71	0.71	0.73	0.73	0.73	0.73
mac. avg	0.82	0.77	0.77	240	0.77	0.77	0.77	240	0.76	0.74	0.74	240	0.77	0.78	0.77	240
wei. avg	0.77	0.74	0.73	240	0.74	0.73	0.73	240	0.72	0.71	0.70	240	0.73	0.73	0.73	240

Since DenseNet121 has the highest accuracy among all the multiple classification models, we further fine-tuned DenseNet121. Our final model used three different learning rates. In the first phase, we used the learning rate of 0.001. At this stage, we froze all the parameters in the DenseNet structure and only trained the top layers. This phase can be seen as the “warm up” for the model and only has 10 epochs. In the second phase we unfroze all the layers and trained the model on a smaller learning rate (0.0001). This stage will run for 100 epochs. We can see that the validation accuracy stopped increasing after 70 epochs. So we added another stage with a smaller learning rate (0.00001). This stage will run for 100 epochs as well.

After further fine-tuning, we can see a slight improvement in validation accuracy. Accuracy from test data went from 0.71 to 0.77. COVID-19 cases precision and recall are both close to 1. Pneumonia recall went from 0.54 to 0.71 (Fig. 6).



	prec.	recall	f1-sc.	support
normal	0.71	0.74	0.73	101
covid19	1.00	0.97	0.99	39
pneumonia	0.73	0.71	0.72	100
acc.	0.77	0.77	0.77	0.77
mac. avg	0.82	0.81	0.81	240
wei. avg	0.77	0.77	0.77	240

Fig. 6 DenseNet121 second fine-tuned multiple classification results -

learning curve accuracy, learning curve accuracy, confusion matrix and other metrics

In addition to the above metrics, we visualize the final layer weight as a heat map to show which parts of an image our CNN model is looking at to make a decision. Below images contain examples of X-ray heatmaps along with the labels, which are based on two models of DenseNet121 and VGG19 respectively (Fig. 7). The highlighted heatmaps of lung regions potentially contain most features of COVID-19 infection which are extracted with Grad-Cam method, and the red arrow shows that most of the infected area was annotated by our board certified radiologist (on the right).

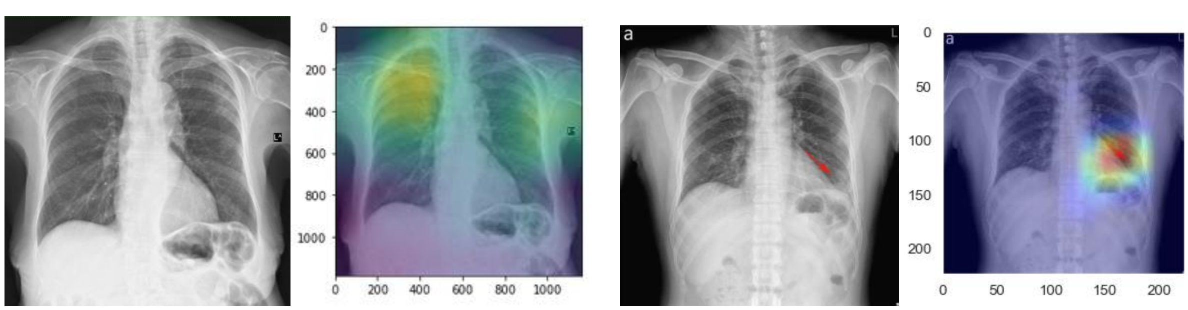


Fig. 7 Heat map Visualization of DenseNet121 (left) and VGG19 (right)

6. Discussion

In this study, we applied deep learning algorithms on X-ray imagery for classification of emerging COVID-19. By using various Convolutional Neural Network (CNN) models, we are able to achieve high classification accuracy from most of the models. However, leveraging CNNs for multiple classification is a very complicated procedure. Although the model can detect COVID cases from the other two classes, it has difficulty distinguishing between healthy cases and pneumonia, possibly because the model put higher weights on the features that can detect COVID cases and lower weights to classify normal and pneumonia cases. Therefore, there is a lot of room for improvement and fine tuning. For example, we can see that the training accuracy stabilized very quickly while validation accuracy was fluctuating as the number of epochs grew. This can be an indicator of overfitting, so reducing the number of epochs or using a smaller learning rate may be something to investigate. Another potential approach to improve the model performance is to construct a customized CNN structure. Although the pretrained CNN models (DenseNet, ResNet, etc.) can be a good starting point, the fixed model structure left us with limited space for hyperparameter tuning as only the top layers can be adjusted. For future work, we can design our own CNN structure where each filter and pooling layer can be tuned. In addition to the classification model, we created heatmaps to visualize the features that the models learned. We used Grad-Cam method for creating a heatmap, which is helpful for distinguishing COVID-19 infection with x-ray chest image in our study, however, the heatmap sometimes is not stable for non-COVID-19 images, which seems as signal noise so we need to further optimize and exclude. The setup of the running environment is inevitably restricted by the huge amount of complicated computation, besides using CUDA-based GPU in personal computers, we think run in AWS which could greatly contribute to the efficiency of the parameters tuning.

7. Conclusion

In this study, we applied transfer learning on CNN models to detect COVID-19 cases from normal cases and other pneumonia cases using X-ray images. Our model reached 99% accuracy for binary classification tasks and 77% accuracy for 3-class tasks respectively. Furthermore, we visualized the important features learned from models, which can provide valuable insights for medical diagnostics.

References

1. Wu F, A et al. new coronavirus associated with human respiratory disease in China. Nature, 579:265:269.
2. Zu ZY et al. Coronavirus disease 2019 (COVID-19): A perspective from China. Radiology, 200490.
3. Dai WC et al. CT Imaging and differential diagnosis of COVID-19. Canadian Association of Radiologists Journal. 2020;71:195-200.
4. Ozturk T, et al. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine 2020;121:1-11.
5. Sohaib A, et al. Classification of COVID-19 from chest X-ray images using deep convolutional neural networks. MedRxiv preprint. 2020.
6. Abbas A, et al. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. Applied Intelligence 2020, Springer.
7. Khuzani AZ, et al. COVID-Classifier: an automated machine learning model to assist in the diagnosis of COVID-19 infection in chest X-ray images. MedRxiv preprint. 2020.
8. Ozturk S, et al. Classification of Coronavirus (COVID-19) from X-ray and CT images using shrunken features. International Journal of Imaging Systems and Technology. 2020;1-11.
9. Mahdy LN et al. Automatic X-ray COVID-19 lung image classification system based on multi-level thresholding and support vector machine. MedRxiv preprint. 2020.
10. Minaee S et al. Deep-CoVID : Predicting COVID-19 from chest X-ray images using deep transfer learning. Medical Image Analysis. 2020:1-9.
11. Sarker M et al. COVID-DenseNet: A deep learning architecture to detect COVID-19 from chest radiology images. Preprints (www.preprints.org). 2020.
12. Zhang et al. Dive into Deep Learning. 2020. Release 0.14.4.
13. Shibly KH et al. COVID faster R-CNN: A novel frame work to diagnose novel Coronavirus disease (COVID-19) in X-ray images. Informatics in Medicine Unlocked. 2020:1-9.
14. Das D et al. Truncated inception net: COVID-19 outbreak screening using chest X-rays. Physical and Engineering Sciences in Medicine. 2020:915-925.
15. Cohen JP et al. COVID-19 image data. https://github.com/ieee8023/covid-chestxray-dataset. 2020.
16. Wang et al. Summers, Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: Proceedings of the IEEE conference on computer vision and pattern Recognition (CVPR) , 2017: 2097–2106.
17. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. InNIPS, 2012
18. Francois Chollet, Transfer learning & fine-tuning, https://keras.io/guides/transfer_learning/
19. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale HierarchicalImage Database. InCVPR09, 2009.
20. R. R. Selvaraju. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. arXiv:1610.02391v4, 2019