CNN-Based Multiclass Classification for COVID-19 in Chest X-ray Images

S.K.Towfek [*1]

[1] Computer Science and Intelligent Systems Research Center, Blacksburg 24060, Virginia, USA

Emails: sktowfek@jcsis.org

Abstract

Managing the increasing number of patients requiring first screening can be significantly aided by real-time automated detection of COVID-19. It's feasible that Deep CNN models that have been trained on sufficiently large datasets will emerge as the most promising options for achieving the goal. This study aims to automatically detect and classify COVID-19 and viral pneumonia infections in chest X-ray images using a deep CNN model. Our proposed model relies on multiclass labeling to accomplish our aims. Design and Organization: Using the ImageNet pre-trained weights, the proposed model is built on top of the VGG16 framework. Additional custom layers were used to fine-tune the model and produce a better overall performance that is more specific to the goal. In terms of its subjects and methods, this study uses 15,153 samples in total. There are X-rays of the lungs from patients with COVID-19, those with viral pneumonia, and healthy volunteers. The entire dataset was split into an 80:20 split for the train and test sets before the model was trained. Image preprocessing and augmentation were used to enhance crucial parts of the photos before they were sent to the model in batches. We measure the model's efficacy with accuracy, precision, recall, and the F1 score. The analysis that was performed statistically was. The model's output is compared to the results of other recent research that have set the standard in the field. The proposed model has a 98% accuracy in multiclass classification on the test dataset, as measured by 98% precision, 96% recall, and 97% F1 score. Receiver operating characteristic curve area scores of 0.99 were achieved in all three multiclass classification situations. Finally, the proposed categorization model may show to be highly useful in the first diagnosis of COVID-19 and viral pneumonia patients, especially when dealing with heavy workloads and large volumes.

Keywords: Keywork one; Keywork two; Keywork three; Keyword four; ….

1. Introduction

Respiratory infections caused by viral pneumonia or COVID-19 may become severe, especially for old age people, people already having a chronic medical illness, or people with a weak immune system. Pneumonia is an infectious disease that causes lung inflammation, which viruses, bacteria, fungi, or other germs may cause. The tiny sacs of the lungs are filled with pus and fluid, which makes breathing painful due to limited oxygen intake [1]. Pneumonia can also be a complication of COVID-19, the illness caused by the new coronavirus known as SARS-CoV-2. The symptoms of COVID-19 may be similar to the other kinds of viral pneumonia; therefore, it is hard to discover the cause of infection without a test [2]. In recent years, a large number of studies and research work have been carried out on applying machine learning models for automated detection of various diseases such as diabetic retinopathy,[3,4] breast cancer,[5,6] pulmonary nodules diagnosis,[7] and lung cancer [8]. Deep learning methods reveal very subtle features in images that are not visible otherwise. Specifically, the ability of convolutional neural networks (CNN) in deep feature

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

extraction and learning has made them the leading choice among researchers for classification-related tasks in medical imaging problems [9]. Both pre-trained deep CNNs and CNNs trained from scratch can produce very high results if trained and fine-tuned robustly.

This study aims to apply deep transfer learning capabilities from pre-trained deep learning models to classify COVID-19 infections and viral pneumonia cases. In this study, we engineered and trained a deep CNN model based on the VGG16 model with custom layers. The model is trained and validated on an open public dataset. The dataset contains images for COVID-19, viral pneumonia, and typical cases discussed in the next section.

2. Methods

Deep architectures in CNN models allow for robust feature extraction, leading to high performance in image classification. Since pneumonia frequently appears foggy on X-rays, radiologists may have trouble diagnosing it correctly. It may also be a waste of time because the traits may coincide with those of other harmless oddities. As a result of these divergences of opinion, radiologists may arrive at different pneumonia diagnoses for different patients [10]. For this research, we create deep CNN models. Inputs to these models include frontal chest X-ray images, and outputs include binary and multiclass classifications indicating the presence or absence of pneumonia and COVID-19 abnormalities.

A. Data Source

After the COVID-19 pandemic broke out, several scientists and doctors began releasing publicly available datasets comprising X-ray, CT, and ultrasound images of lung diseases and their corresponding labels. We used X-ray images of COVID-19, viral pneumonia, and regular patients from a prize-winning dataset compiled by researchers and healthcare practitioners [11,12]. The photos included in the collection were culled from numerous open-access databases, websites, and previously published studies [13]. Our group used various images from the updated version of this dataset, including those of COVID-19, viral pneumonia, and regular chest X-rays. There are a total of 15,153 images throughout the three different types. The collection also contains images depicting lung opacity. However, we have decided against using them. There are 10,192 photographs of healthy individuals, 1345 photographs of individuals with viral pneumonia, and 3616 photographs of individuals who test positive for COVID-19. The PNG format is used for all the images in the collection, and each has an exact width and height of 299 pixels.

The images are available on the Kaggle website. The pictures used for Figure 1 were randomly picked from the entire set. Before training the model, the entire dataset was split 80/20 between a training set and a validation set. The images were solely utilized for the test. Thus, the model did not have access to them during training. To assist a dispassionate evaluation of the model, these images were first isolated from one another and stored separately. Afterward, a ratio of 80:20 was applied to further partition the train set into train and validation sets. In conclusion, are 9697 images in the training set, 2425 in the validation set, and 3031 in the test set.

Figure 1: Sample X-ray images from dataset: (a) COVID-19, (b) viral pneumonia, (c) normal

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

B. Image Preprocessing

Some offline preprocessing operations and online image augmentation methods enhanced crucial image features and lessened unwanted distortions in the training set. Preprocessing required reducing the size of each image to 224 by 224 pixels, the default input size for the VGG16 model. We also employed a VGG16-specific preprocess input function on both the training and validation sets before feeding the data to the model. This technique converts the photos from RGB to BGR and centers each color channel at zero about the images in the ImageNet database. In this procedure, scaling is not used at any point. To fine-tune the VGG16 model, which had been trained on ImageNet, we converted single-channel chest X-ray images to multi-channel ones with three colors. Image augmentation was also used to expand the quantity of photos and the variety of those photos by randomly rotating each image in the training set by an angle between 0 and 5 degrees. Multiple versions of a post-production-enhanced image are shown in Figure 2.

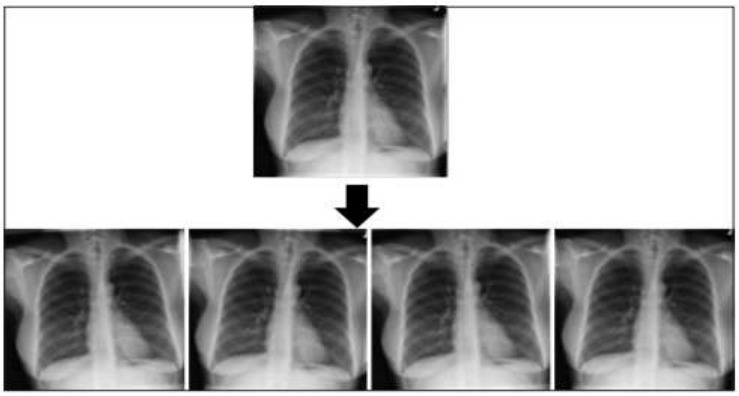

Figure 2: Image variants after augmentation

C. Model Architecture

Deep convolutional neural network (CNN) models are the best alternative for advancing healthcare research because of their ability to extract complex lower-level information in images from the original training dataset. By examining X-rays of the lungs, we proposed a deep CNN model for distinguishing between COVID-19, viral pneumonia, and healthy individuals. As part of that work, we came up with this model. The proposed model is built with pre-trained weights for the ImageNet database and the VGG16 underlying architecture. The model's degree of detail was improved by using extra custom layers. Simonyan and Zisserman developed the VGG16 convolutional neural network (CNN), and it is a simple and widely used CNN. VGG16 earns an accuracy score of 92.7% on the ImageNet dataset, which contains over 14 million images from 1000 categories. The presented model can be used as a diagnostic tool for primary screening and distinguishing between COVID-19, viral pneumonia, and healthy patients. Figure 3 depicts the overall design of the provided model.

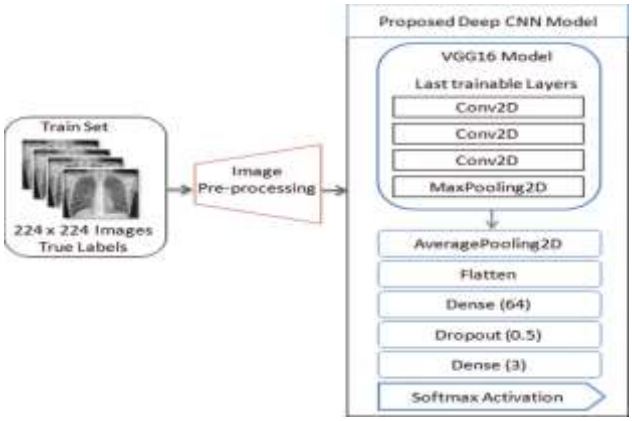

Figure 3: Architecture of proposed deep convolutional neural network model

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

The provided deep CNN model is based on the VGG16 pre-trained model. At first, we made the entire model trainable to retrain the network's pre-trained parameters. However, this did not lead to the expected outcomes. Then, we began methodically removing preconceptions and assessing the results. We found that the model performed best when the last four layers were left open (trainable) and the remaining ones were frozen (pre-trained). We built a few layers of original code on top of this framework. These extra layers allowed the model to generate far better results that were more specific to our objective.

Figure 3 demonstrates the five specific layers applied to the base model. After that, we used the AveragePooling2D technique with a pool size of (4, 4) to down-sample the image. The flattened layer produced a 1D array of elements, eliminating all dimensions. Next, a dense layer was applied with 64 output neurons, followed by a dropout layer to ignore half of the neurons in a random sample, and finally, a third dense layer was applied with three output neurons. The final classification result was generated using the softmax activation function. The model weights were first assigned randomly, and subsequent updates were made according to the gradient of the loss as it was computed about the truth. It was intended that the loss function be minimized optimally. Because the model does not acquire knowledge from this information, the validation set has only a secondary effect on the model's training. We used a validation set to reduce the likelihood of the model being overfit and to fine-tune its hyperparameters. The optimal final weight was recorded and used for further evaluation in the test set that followed model training. Since it was created independently of the original dataset and used for testing purposes alone, the test set contains images that the model has never seen before (during training or validation). In Table 1, we show our chosen values for the model's hyperparameters.

Table 1: Parameters during the training

Parameter	Value
Image size	222 X 224 pixel
Optimizer	Adam
Batch size	8
Initial Learning rate	0.0001
Weight decay	0.0005
Loss function	Categorical cross-entropy

D. Performance Metric

Accuracy, sensitivity, specificity, precision, recall, F1 score, and confusion matrix are the commonly used metrics for measuring the performance of classification models. These metrics are derived using four values: true positive, true negative, false positive, and false negative. We used precision, recall, and F1 scores to measure the performance of our model. We also plotted the receiver operating characteristic (ROC) curve, considered the de facto standard for classification tasks in medical images.

The following equations show the formula of accuracy, precision, recall, and F1 score to measure the model's performance for multiclass classification.

$$Accuracry = \frac{TP_n + TP_c + TP_p}{Total\ Samples} \quad (1)$$

$$Percision = \frac{TP_c + TP_p}{TP_c + FP_c + TP_p + FP_p} \quad (2)$$

$$Recall = \frac{TP_c + TP_p}{TP_c + TP_p + FN_c + FN_p} \quad (3)$$

$$F1\ Score = \frac{2\ Precision \times Recall}{Precision + Recall} \quad (4)$$

Where

TP_c: COVID-19 positive patients predicted to be positive

TP_p: Viral pneumonia-positive patients predicted to be pneumonia positive

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

TP_n: Normal patients expected to be normal

FP_c: Normal patients expected to be COVID-19 positive

FP_p: Normal patients expected to be viral pneumonia positive

FN_c: COVID-19 patients expected to be normal

FN_p: Viral pneumonia-positive patients expected to be regular patients

E. Experimental Setup

Training a deep neural network is highly resource-intensive due to the need to perform matrix multiplication on the input and weight arrays. The graphics processing unit (GPU) is increasingly replacing CPUs as the go-to piece of hardware for deep learning model training because of its larger memory bandwidth and more significant number of cores. More traditional approaches utilizing central processing units (CPUs) could achieve this, but it would take too long, especially for models with billions of parameters. In this investigation, we use GPU nodes from the Kshitij-5 high-performance computing cluster to conduct our training, validation, and testing.

3. Results

The proposed model is verified by applying it to a test set consisting of 317 images. These images are never utilized in the model's training and have never been seen by it before. A CNN model is created with 49339 parameters as shown in figure 4.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_16 (Conv2D)          (None, 62, 62, 32)        896

 conv2d_17 (Conv2D)          (None, 60, 60, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 20, 20, 32)       0
 2D)

 conv2d_18 (Conv2D)          (None, 18, 18, 40)        11560

 conv2d_19 (Conv2D)          (None, 16, 16, 32)        11552

 max_pooling2d_9 (MaxPooling  (None, 5, 5, 32)         0
 2D)

 flatten_4 (Flatten)         (None, 800)               0

 dense_8 (Dense)             (None, 20)                16020

 dropout_4 (Dropout)         (None, 20)                0

 dense_9 (Dense)             (None, 3)                 63

=================================================================
Total params: 49,339
Trainable params: 49,339
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Model parameters

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

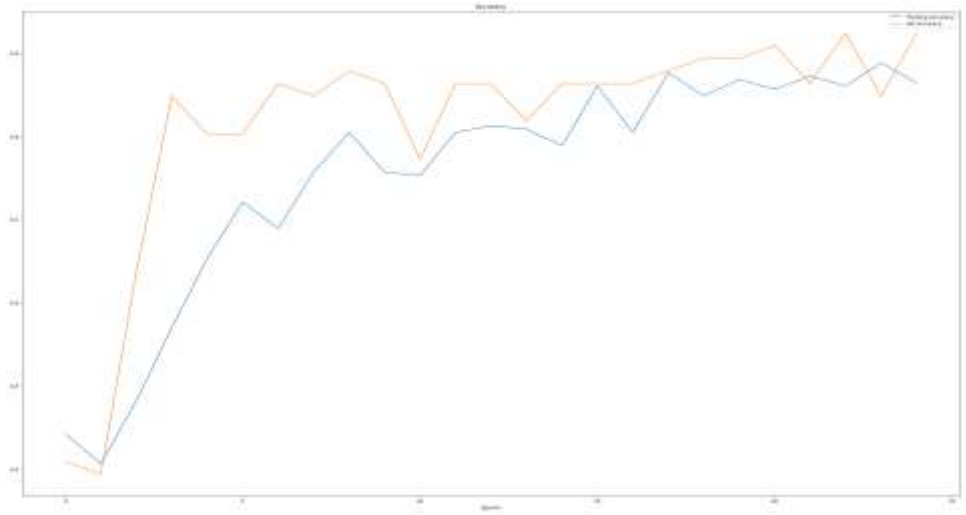

Figure 5: The training accuracy and the validation accuracy

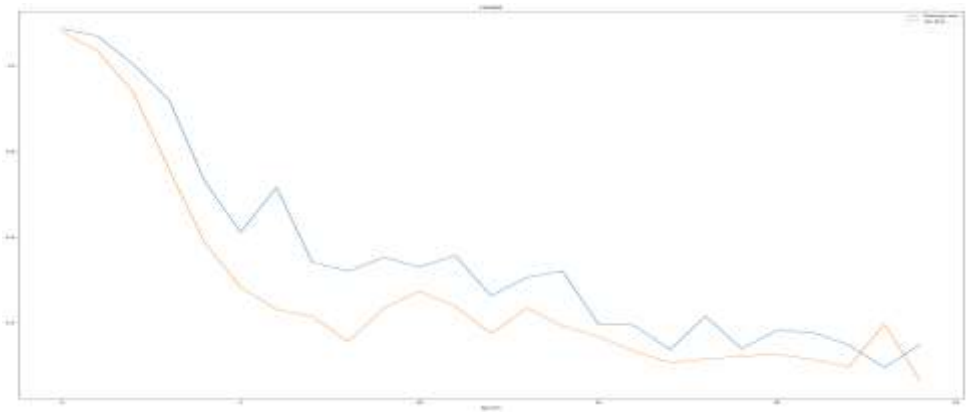

Figure 6: The training loss and the validation loss

Table 2: The performance of cnn model

	accuracy	loss	val_loss	val_accuracy
0	0.442231089	1.086398125	1.079683304	0.409090906
1	0.406374514	1.069277525	1.033983827	0.393939406
2	0.482071728	1.002218962	0.938227773	0.636363626
3	0.569721103	0.919577777	0.756598532	0.848484874
4	0.653386474	0.729153991	0.585980356	0.803030312
5	0.72111553	0.611150563	0.482193589	0.803030312
6	0.689243019	0.715445638	0.430938035	0.863636374
7	0.756972134	0.541456819	0.414068162	0.848484874
8	0.8047809	0.519967496	0.355484754	0.878787875
9	0.756972134	0.552279294	0.434084624	0.863636374
10	0.75298804	0.530174613	0.473259598	0.772727251
11	0.8047809	0.556382537	0.438408434	0.863636374
12	0.812749028	0.462405145	0.37415728	0.863636374
13	0.808764935	0.505527318	0.433781236	0.818181813
14	0.788844645	0.520447135	0.392340392	0.863636374

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

15	0.860557795	0.396957964	0.366987735	0.863636374
16	0.8047809	0.393620551	0.33377862	0.863636374
17	0.87649405	0.33672002	0.306430936	0.878787875
18	0.848605573	0.41489476	0.315303177	0.893939376
19	0.868525922	0.339415193	0.32115373	0.893939376
20	0.856573701	0.381623626	0.326183051	0.909090936
21	0.872509956	0.377552658	0.313632339	0.863636374
22	0.860557795	0.347534359	0.29675284	0.924242437
23	0.888446212	0.295143008	0.396822453	0.848484874
24	0.864541829	0.349391341	0.263835341	0.924242437

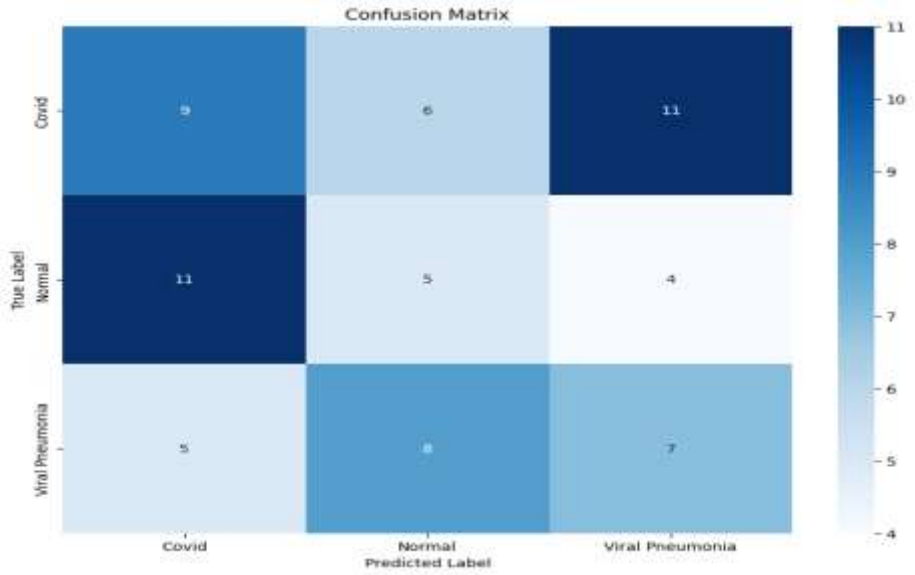

Figure 7: The confusion matrix CNN model

6. Conclusion

In this study, we sought to automate the process of classifying chest X-ray pictures for the presence of COVID-19 and other viral pneumonia using a deep CNN model built on top of the popular deep learning framework VGG16. The proposed methodology could aid clinicians in making rapid, preliminary diagnoses of COVID-19 and viral pneumonia. The model's results are noteworthy because it successfully classifies the three cases with high recall and precision. The model can also perform this function. The next step of this research will involve adding hospital images to the training dataset to make the model more accurate and dependable. Furthermore, one of our objectives is to identify the infectious parts of the images by creating class activation maps.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Pneumonia. Available from: https://www.who.int/news-room/ fact-sheets/detail/pneumonia.

[2] Seladi-Schulman J. What to Know About COVID-19 and Pneumonia. Healthline. Available from: https://www.healthline.com/health/ coronavirus-pneumonia#vs-regular-pneumonia.

[3] Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402-2410, 2016.

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023

[4] Gargeya R, Leng T, Automated identification of diabetic retinopathy using deep learning. Ophthalmology, 124(7), 962-969, 2017.
[5] Amrane M, Oukid S, Gagaoua I, Ensari T, Breast Cancer Classification Using Machine Learning. 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), Istanbul, 1-4, 2018.
[6] Hussain L, Aziz W, Saeed S, Rathore S, Rafique M. Automated Breast Cancer Detection Using Machine Learning Techniques by Extracting Different Feature Extracting Strategies. 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12 IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 327-31, 2018.
[7] Xie Y, Xia Y, Zhang J, Song Y, Feng D, Fulham M, et al, Knowledge-based collaborative deep learning for benign-malignant lung nodule classification on chest CT. IEEE Trans Med Imaging, 38(4), 991-1004, 2019.
[8] Wu Q, Zhao W., Small-Cell Lung Cancer Detection Using a Supervised Machine Learning Algorithm. 2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC), Budapest, 88-91, 2017.
[9] Krizhevsky A, Sutskever I, Hinton GE., Imagenet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, 1097-105, 2012.
[10] Neuman MI, Lee EY, Bixby S, Diperna S, Hellinger J, Markowitz R, et al, Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children. J Hosp Med, 7, 294-298, 2012.
[11] Chowdhury ME, Rahman T, Khandakar A, Mazhar R, Kadir MA, Mahbub ZB, et al, Can AI help in screening viral and covid-19 pneumonia?, IEEE Access, 8, 65-76, 2020.
[12] Rahman T, Khandakar A, Qiblawey Y, Tahir A, Kiranyaz S, Kashem SB, et al, Exploring the effect of image enhancement techniques on covid-19 detection using chest X-ray images. Computers in Biology and Medicine, 132, 2021.
[13] BIMCV-COVID19, Datasets Related to COVID19's Pathology Course. Available from: https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1 590858128006-9e640421-6711. [Last accessed on 2021 Mar 06].
[14] COVID-19-image-repository. Available from: https://github.com/ ml-workgroup/covid-19-image-repository/tree/master/png.
[15] COVID-19 Database. Available from: https://sirm.org/category/ senza-categoria/covid-19/.
[16] "Eurorad.org." Eurorad. Available from: http://www.eurorad.org/. [Last accessed on 2021 Mar 06].
[17] "COVID-19 Image Data Collection." Cohen2020covid, 2020. Available from: http://github.com/ieee8023/covid-chestxray-dataset.
[18] Arman H, Mahdiyar MM, Seokbum K. COVID-19 Chest X-Ray Image Repository. Figshare Dataset; 2020. Available from: https://doi. org/10.6084/m9.figshare. 12580328.
[19] "Armiro/COVID-CXNet." Armiro, 2020. Available from: https:// github.com/armiro/COVID-CXNet.
[20] RSNA Pneumonia Detection Challenge. Available from: https://www. kaggle.com/c/rsna-pneumonia-detection-challenge/overview. [Last accessed on 2021 Mar 06].

Doi: https://doi.org/10.54216/JAIM.060105
Received: February 18, 2023 Revised: May 13, 2023 Accepted: November 17, 2023