Tutorial

Healthc Inform Res. 2021 January;27(1):82-91.
https://doi.org/10.4258/hir.2021.27.1.82
pISSN 2093-3681 • eISSN 2093-369X

HIR
Healthcare Informatics Research

Codeless Deep Learning of COVID-19 Chest X-Ray Image Dataset with KNIME Analytics Platform

Jun Young An[1,*], Hoseok Seo[2,*], Young-Gon Kim[1,3,*], Kyu Eun Lee[1,4], Sungwan Kim[1,3,5], Hyoun-Joong Kong[2,3]

[1]Institute of Medical & Biological Engineering, Medical Research Center, Seoul National University College of Medicine, Seoul, Korea
[2]Department of Biomedical Engineering, Chungnam National University College of Medicine, Daejeon, Korea
[3]Transdisciplinary Department of Medicine and Advanced Technology, Seoul National University Hospital, Seoul, Korea
[4]Department of Surgery, Seoul National University College of Medicine, Seoul, Korea
[5]Department of Biomedical Engineering, Seoul National University College of Medicine, Seoul, Korea

Objectives: This paper proposes a method for computer-assisted diagnosis of coronavirus disease 2019 (COVID-19) through chest X-ray imaging using a deep learning model without writing a single line of code using the Konstanz Information Miner (KNIME) analytics platform. **Methods:** We obtained 155 samples of posteroanterior chest X-ray images from COVID-19 open dataset repositories to develop a classification model using a simple convolutional neural network (CNN). All of the images contained diagnostic information for COVID-19 and other diseases. The model would classify whether a patient was infected with COVID-19 or not. Eighty percent of the images were used for model training, and the rest were used for testing. The graphic user interface-based programming in the KNIME enabled class label annotation, data preprocessing, CNN model training and testing, performance evaluation, and so on. **Results:** 1,000 epochs training were performed to test the simple CNN model. The lower and upper bounds of positive predictive value (precision), sensitivity (recall), specificity, and f-measure are 92.3% and 94.4%. Both bounds of the model's accuracies were equal to 93.5% and 96.6% of the area under the receiver operating characteristic curve for the test set. **Conclusions:** In this study, a researcher who does not have basic knowledge of python programming successfully performed deep learning analysis of chest x-ray image dataset using the KNIME independently. The KNIME will reduce the time spent and lower the threshold for deep learning research applied to healthcare.

Keywords: COVID-19, Mass Chest X-Ray, Diagnosis, Computer Assisted, Deep Learning, KNIME

Submitted: January 4, 2021
Accepted: January 25, 2021

Corresponding Author
Hyoun-Joong Kong
Transdisciplinary Department of Medicine & Advanced Technology, Seoul National University Hospital, 101 Daehak-ro, Jongno-gu, Seoul 03080, Korea. Tel: +82-2-2072-4492, E-mail: gongcop@melab.snu.ac.kr (https://orcid.org/0000-0001-5456-4862)
*These authors contributed equally to this work.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ⓒ 2021 The Korean Society of Medical Informatics

I. Introduction

According to a market report, artificial intelligence (AI) in healthcare is expected to grow rapidly in the next 5 years [1]. Healthcare-related AI publications have been increasing by an average of 17% every year since 1995 [2,3]. Future studies are expected to use machine learning or deep learning models. However, few professionals, such as doctors, professors, and researchers, have opportunities to learn programming languages and apply AI in their research. As a solution, graphic user interface (GUI) tools have been developed. Keras UI [4], a laboratory virtual instrument engineering workbench, the Waikato environment for knowledge analysis, and the Konstanz Information Miner (KNIME) analytics

platform are tools that can analyze data without coding.

Keras UI can run the training process by uploading dataset items (images) to the web application by pressing the train button. The results are provided in a log file. The tool's advantage is that personal computer (PC) performance is not an issue because the dataset is uploaded to a web server. Even a smartphone can serve this purpose. Furthermore, model verification is easy because it provides the model's weights as the result [4]. The laboratory virtual instrument engineering workbench enables ease of use with the application's GUI. It is also efficient in running and avoiding unnecessary operations, especially in repeated tasks, such as looping structures. This makes the application reliable and efficient. In addition, it minimizes the complexity by requiring fewer nodes [5]. The entry barrier is not high for beginners in learning and handling tools. However, customizing the code in detail is not an option in graphical format. The advantages of the Waikato environment for knowledge analysis are that it provides a wealth of state-of-the-art machine learning algorithms, and it is open-source and fully implemented in Java, which allows it to be run on almost any platform. The main disadvantage is that all data are supposed to be in the main memory. Not many algorithms may run data increasingly or in batches. In addition, C/C++ implementation is relatively faster than Java implementation [6].

In this paper, we introduce the KNIME analytics platform, which addresses the shortcomings of these existing tools and is capable of performing state-of-the-art deep learning algorithms [7], such as convolutional neural network (CNN) analysis. We performed a deep learning analysis to determine whether a patient was infected with coronavirus disease 2019 (COVID-19) using a simple CNN model [8]. Thus, we validated the performance of the model, which is meaningful during the COVID-19 pandemic. By applying the methods used in this study, professionals who are not familiar with coding will be able to begin research with machine learning or deep learning analysis.

II. Methods

1. Installation and Settings

In this study, a PC with an Intel I7 8700 K 3.70 GHz processor, 32 GB DDR4 RAM, NVidia GeForce GTX 1080 8 GB, Anaconda with Python, and TensorFlow was used. Once the installation of the KNIME analysis platform was complete, the latest Anaconda was downloaded from https://docs.knime.com/latest/python_installation_guide/#anaconda_setup. The procedures to install Anaconda and KNIME extensions were followed as shown in Figures 1 and 2, respectively.

After installing Anaconda and extensions, we went to the Preference menu located at "File→Preferences," to create the Conda and Deep Learning Conda environment as shown in Figure 3. If a PC has a graphic card, a graphics processing unit (GPU) environment can be created for deep learning.

2. Preprocessing of Dataset

In this study, we extracted 88 patients diagnosed positive to COVID-19 and 67 negatives. All patients were admitted to the intensive care unit. The dataset is collected from public sources, such as the Italian Society of Medical and Interventional Radiology, the Hannover Medical School, and other institutions [9,10]. Data consisting of two different formats were converted to Portable Network Graphics. To make sorting simple, we put "1_" in front of the file names of the positive dataset and "0_" of the negative dataset.

As shown in Figure 4, the "List Files" node imported the dataset. "Sorter" node sorted the data in ascending order, which put negative data first, followed by positives. The "Rule Engine" node classified the first 67 data as "negative" and the rest as "positive" and added the classification column on the datasheet. The "Row Sampling" node selected the data from all rows to utilize them. The "Load and Preprocess Images (local files)" node in KNIME handles the normalization and resizing of the images. Right-clicking the node and clicking "Component→Open" activated the editor window, which allowed us to normalize and resize the images to 150 × 150. The "Column Filter" node removed unnecessary variables. In the final preprocessing stage, a table for a deep learning model was prepared by executing the "Table Writer" node. Then, we trained a simple CNN model to conduct a computer-assisted diagnosis of COVID-19.

3. Training

The process of training and testing the model is depicted in Figure 5. The table from preprocessing needed to be imported. The "DL Python Network Creator" node was coded to generate a simple CNN. The user may check and modify the code if necessary in configuration, which can be found by right-clicking the node. First, the "Table Reader" node imported the table stored in preprocessing. In the "Partitioning" node, the dataset was divided into two sets randomly. Eighty percent of the dataset was selected to train the model, and the rest was used to test it. The "Rule Engine" node was used to set the rule of turning "negative" into 0 and "positive" into 1. The "DL Python Network Learner" node learned the

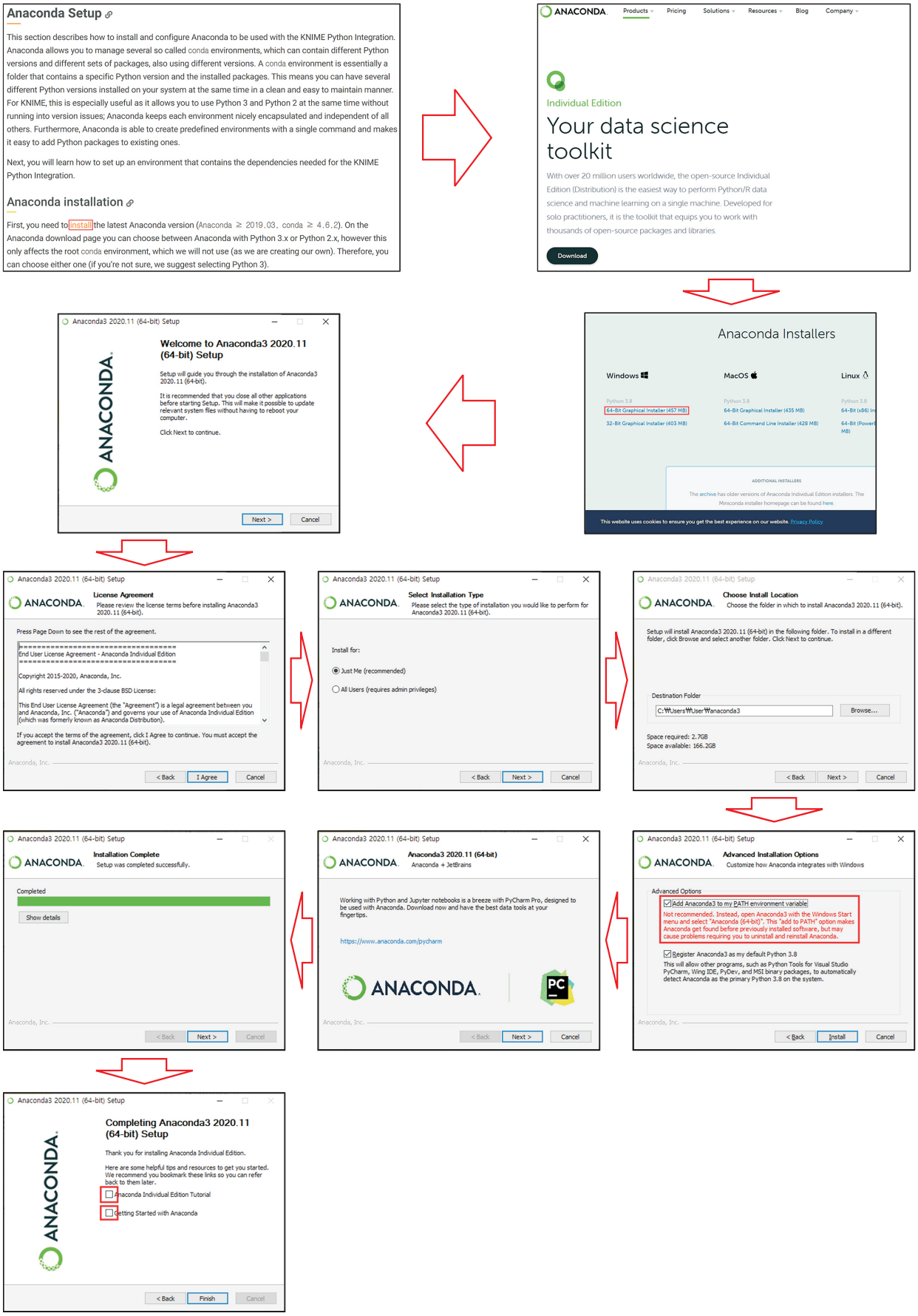
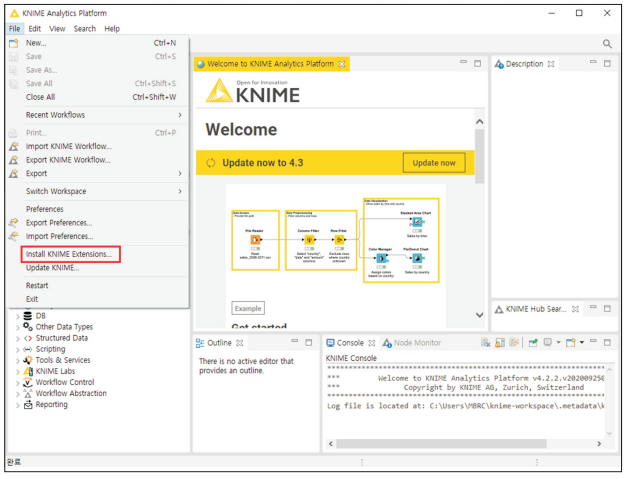
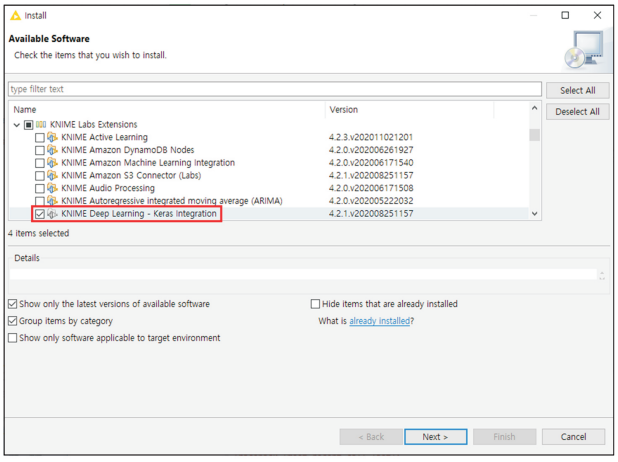


Figure 1. Installation procedures of Anaconda.

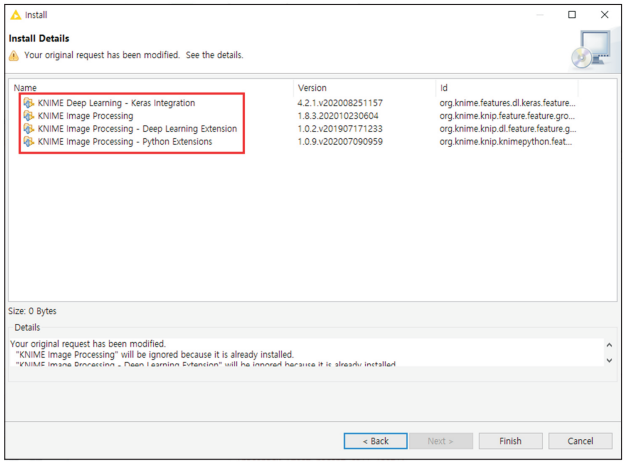
https://doi.org/10.4258/hir.2021.27.1.82



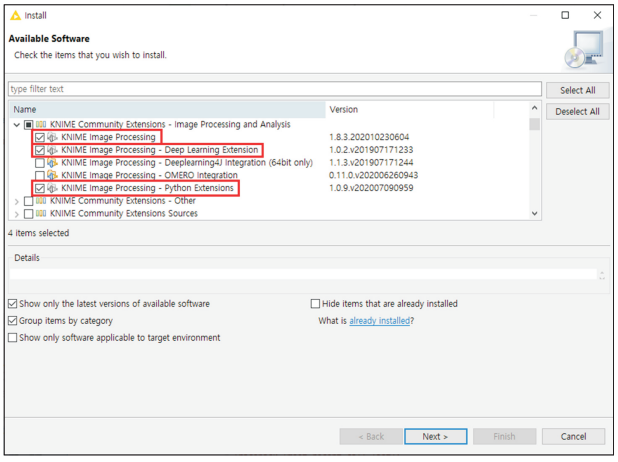
A Install KNIME extensions



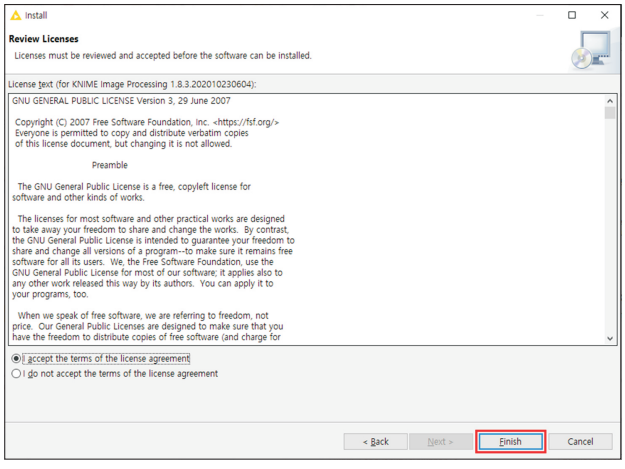
B Select extensions (1)



D Install detail extensions



C Select extensions (2)



E License agreement

F Restart

Figure 2. Installation procedures of extensions. (A) KNIME extensions should be installed to execute deep learning model. (B) Select "KNIME Deep Learning – Keras Integration." (C) Select "KNIME Image Processing," "KNIME Image Processing – Deep Learning Extension," and "KNIME Image Processing – Python Extensions." (D) Click "Finish." (E) Agree to "License agreements." (F) Restart.

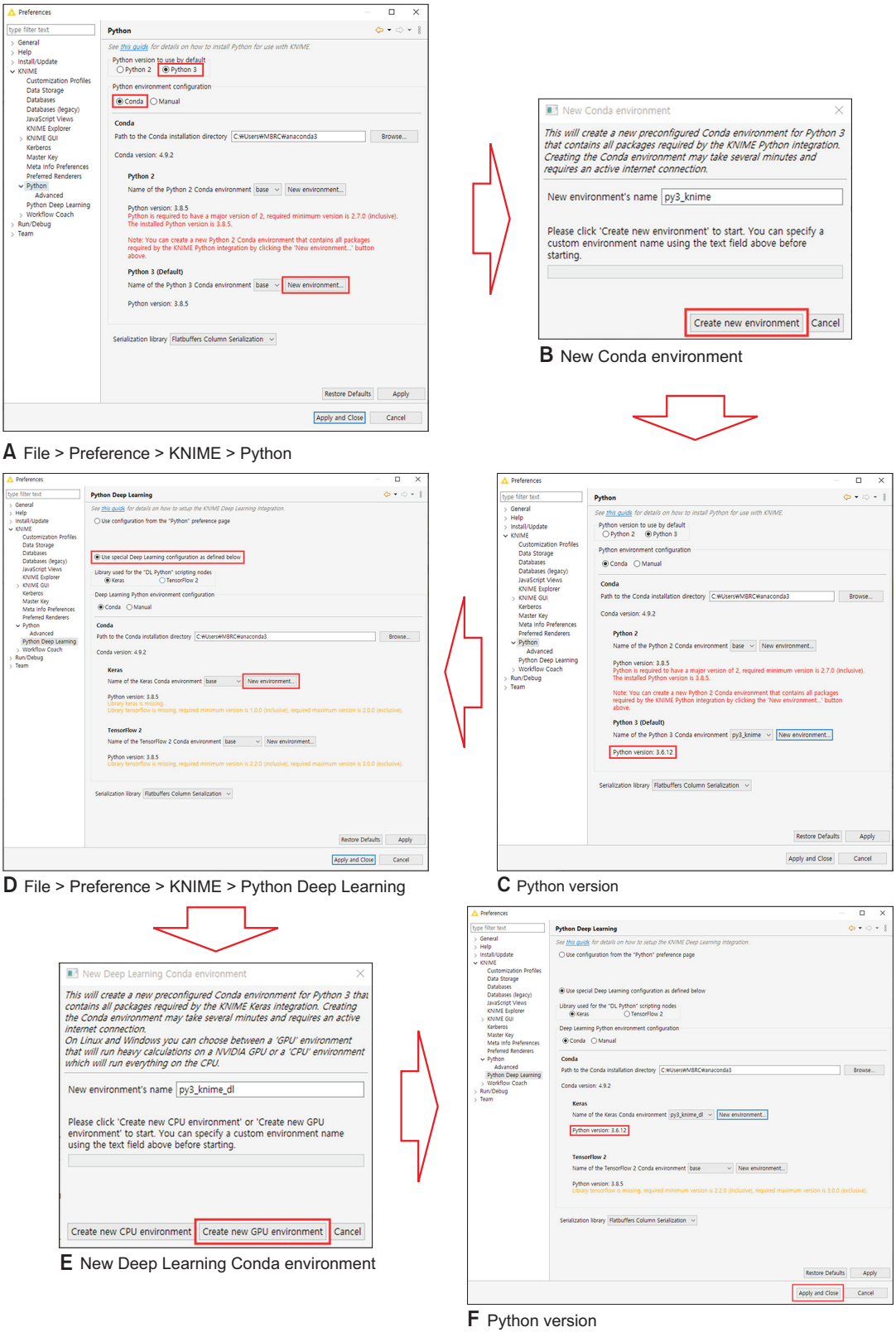


Figure 3. Conda and Deep Learning Conda environment setting. (A) Select "Python3." (B) Create "New Conda environment." (C) Create Python environment in which correct version is matched automatically. (D) Select "Use Special Deep Learning Configuration as Defined Below." (E) Create "New Deep Learning Conda environment." (F) Create deep learning environment in which correct version is matched automatically.

https://doi.org/10.4258/hir.2021.27.1.82

Figure 4. Example of overall preprocessing flow. (A) All nodes for preprocessing. (B) Import filenames of a dataset. (C) Sort by filenames. (D) Classify files as "positive" or "negative." (E) Perform stratified sampling and Open "Load and Preprocess Images" node by right-clicking the node→component→open. (F) Import components. (G) Add image column before filename column. (H) Normalize images. (I) Resize images by 150 × 150. (J) Filter out other columns except image and class columns.

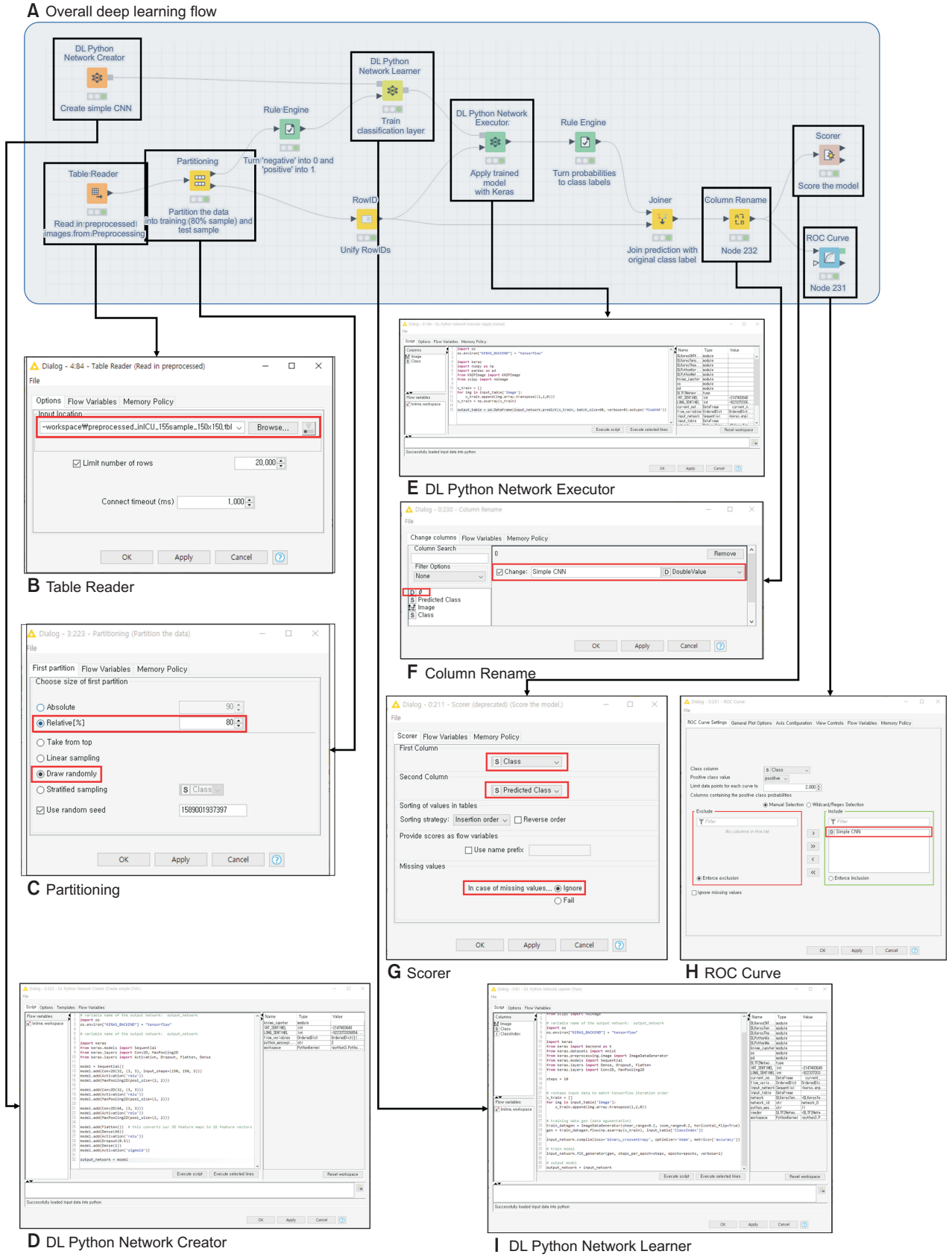


Figure 5. Example of overall deep learning flow. (A) All nodes for deep learning flow. (B) Import table of preprocessed dataset. (C) Divide the dataset into two, one for training and the other for testing. (D) Create a convolutional neural network. (E) Apply the trained model. (F) Rename the column of ROC Curve. (G) Scores the model. (H) Show area under curve. (I) Train the model.

https://doi.org/10.4258/hir.2021.27.1.82

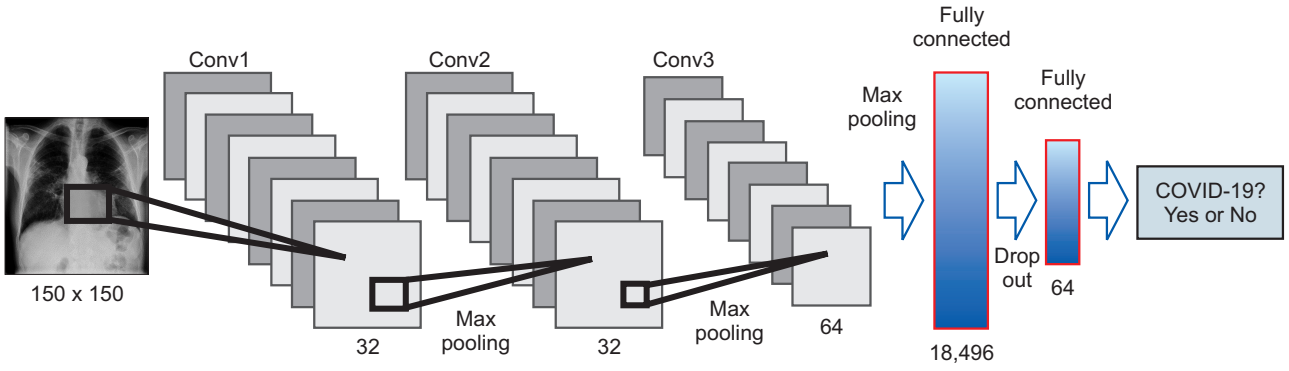


Figure 6. Architecture of the simple convolutional neural network algorithm used in the study.

A Accuracy statistics

Accuracy statistics - 3:211 - Scorer (Score the model.)

File

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Columns: 11	Column Type	Column I...	Color Han...	Size Han...	Shape Ha...	Filter Han...	Lower Bound	Upper Bound
TruePositives	Number (integer)	0					12	17
FalsePositives	Number (integer)	1					1	1
TrueNegatives	Number (integer)	2					12	17
FalseNegatives	Number (integer)	3					1	1
Recall	Number (double)	4					0,923	0,944
Precision	Number (double)	5					0,923	0,944
Sensitivity	Number (double)	6					0,923	0,944
Specifity	Number (double)	7					0,923	0,944
F-measure	Number (double)	8					0,923	0,944
Accuracy	Number (double)	9					0,935	0,935
Cohen's kappa	Number (double)	10					0,868	0,868

B Confusion matrix

Confusion Matrix - 3:211 - Scorer (Score the mod...

File Hilite

Class ₩ ...	negative	positive
negative	12	1
positive	1	17

Correct classified: 29 | Wrong classified: 2

Accuracy: 93,548 % | Error: 6,452 %

Cohen's kappa (κ) 0,868

C ROC Curve

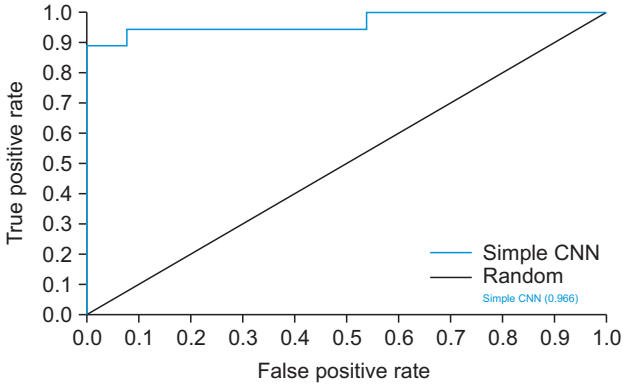


Figure 7. Performance of a simple convolutional neural network (CNN) model in detecting COVID-19: "Scorer" node shows (A) accuracy statistics (true positive, false positive, true negative, false negative, sensitivity, specificity, F-measure, accuracy and Cohen's kappa; (B) confusion matrix (accuracy, Cohen's kappa). (C) "ROC Curve" node shows area under the curve.

model, presented in Figure 6, using the training dataset from the "Partitioning" node. The code was checked and modified by right-clicking on the node and clicking on Setting. The "DL Python Network Executor" node validated the model that was learned using the test dataset. The result presents the probability values. After the Executor node, the "Rule Engine" node turned probabilities above 0.5 into "positive," and the rest into "negative." The "Joiner" node combined two different files and compared them. One was the class of the test dataset, and the other was the predicted value. The Scorer node scored the performance of the model.

4. Performance Evaluation

We obtained true positives, false positives, true negatives, false negatives, sensitivity (=recall), specificity, positive predictive value (=precision), accuracy, and area under curve.

III. Results

The extracted images were normalized and resized for equalization. After preprocessing of the dataset and training of a simple CNN model, 1,000 epochs were performed to test the model because the model's accuracy increases until 1,000 epochs [11-13]. The lower and upper bounds of positive predictive value (precision), sensitivity (recall), specificity, and F-measure are 92.3% and 94.4%. Both bounds of the accuracies of the model were equal to 93.5%. The area under curve of the receiver operating characteristic curve was 96.6%, as shown in Figure 7.

IV. Discussion

As the AI market in healthcare grows, research in the healthcare area has increased [1]. However, beginners have difficulty conducting research using machine learning or deep learning. This paper describes the overall deep learning process of applying a simple CNN model via the KNIME analytics platform. KNIME does not require users to code because they can follow the procedures with clicks, drags, and drops, thanks to the GUI. In this study, we trained a simple CNN model with data obtained from 155 intensive care unit patients; this work was expected to have implications for the COVID-19 pandemic. Our analysis showed that the model achieved 93.6% accuracy.

Prior studies also trained CNN models with X-ray images and obtained similar results. Gao [14] stated that, as part of the evidence used to distinguish COVID-19 from other diseases, X-ray images play an essential role. The researcher also mentioned that if hospitals have support in using X-ray equipment and diagnosing COVID-19, it would lead to cost savings. Ismael and Sengur [15] used pre-trained deep CNN models (ResNet18, ResNet50, ResNet101, VGG16, and VGG19) to obtain deep features with accuracies above 85% and below 93%. Maghdid et al. [16] used a simple CNN and modified the pre-trained AlexNet model, with accuracies of 94% and 98%, respectively. The researcher mentioned that a simple CNN model could be used to diagnose COVID-19 and stimulate research. Our study also used a simple CNN model to detect COVID-19 with a 93% accuracy without writing a single line of code through the KNIME analytics platform. This will contribute to lowering the threshold for research in AI.

The KNIME automatically discovered the versions of python, CUDA, and Keras. It is fast and easy to build a workflow by dragging the nodes installed in extensions. Since the example codes for the models are not built-in, some parts are necessary to search for and use the published nodes directly from the KNIME Hub website. A total of 155 data used in the analysis are difficult in view of generalization because the volume of the dataset is small. Furthermore, the data collection period is also not long, requiring larger amounts of data and broader periods of data in the future [17].

Codeless deep learning using the KNIME will reduce the time spent and lower the threshold for deep learning research applied to healthcare because the KNIME provides a user-friendly GUI and compatibility.

Conflicts of Interest

Hyoun-Joong Kong is an editorial member of Healthcare Informatics Research; however, he did not involve in the peer reviewer selection, evaluation, and decision process of this article. Otherwise, no potential conflict of interest relevant to this article was reported.

Acknowledgments

This work was supported by research fund of Chungnam National University.

ORCID

Jun Young An (http://orcid.org/0000-0003-0686-1955)
Hoseok Seo (http://orcid.org/0000-0001-7740-0887)
Young-Gon Kim (http://orcid.org/0000-0003-2148-1299)
Kyu Eun Lee (http://orcid.org/0000-0002-2354-3599)
Sungwan Kim (http://orcid.org/0000-0002-9318-849X)
Hyoun-Joong Kong (http://orcid.org/0000-0001-5456-4862)

References

1. Grand View Research. Artificial intelligence in healthcare market size, share & trends analysis report by component (hardware, software, services), by application, by region, competitive insights, and segment forecasts, 2019-2025 [Internet]. San Francisco (CA): Grand View Research; 2019 [cited at 2021 Jan 27]. Available from: https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-healthcare-market?utm_source=prnewswire&utm_medium=referral&utm_campaign=hc_16-dec-19&utm_term=ai-healthcare-market&utm_content=rd/toc.
2. Guo Y, Hao Z, Zhao S, Gong J, Yang F. Artificial intelligence in health care: bibliometric analysis. J Med Inter-

https://doi.org/10.4258/hir.2021.27.1.82

net Res 2020;22(7):e18228.
3. Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, et al. Artificial intelligence in healthcare: past, present and future. Stroke Vasc Neurol 2017;2(4):230-43.
4. Fontani D. Keras UI: a GUI to manage image classification [Internet]. [place unknown]: Code Project; 2019 [cited at 2021 Jan 27]. Available from: https://www.codeproject.com/Articles/4053651/Keras-UI-A-GUI-to-Manage-Image-Classification.
5. Blume PA. The LabVIEW style book. Boston (MA): Prentice-Hall; 2011.
6. Frank E, Hall M, Holmes G, Kirkby R, Pfahringer B, Witten IH, et al. Weka: a machine learning workbench for data mining. In: Maimon O, Rokach L, editors. Data mining and knowledge discovery handbook. Boston (MA): Springer; 2009. p. 1269-77.
7. KNIME Analytics Platform [Internet]. Zurich, Switzerland: KNIME AG; c2020 [cited at 2021 Jan 27]. Available from: https://www.knime.com/knime-analytics-platform.
8. Lin Q, Zhao S, Gao D, Lou Y, Yang S, Musa SS, et al. A conceptual model for the coronavirus disease 2019 (COVID-19) outbreak in Wuhan, China with individual reaction and governmental action. Int J Infect Dis 2020;93:211-6.
9. RSNA pneumonia detection challenge [Internet]. San Francisco (CA): Kaggle.com; 2018 [cited at 2021 Jan 27]. Available from: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge.
10. Cohen JP, Morrison P, Dao L, Roth K, Duong TQ, Ghassemi M. Covid-19 image data collection: prospective predictions are the future [Internet]. Ithaca (NY): arXiv.org; 2020 [cited at 2020 Jan 28]. Available from: https://arxiv.org/abs/2006.11988.
11. Shaheen F, Verma B, Asafuddoula M. Impact of automatic feature extraction in deep learning architecture. Proceedings of 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA); 2016 Nov 30-Dec 2; Gold Coast, Australia. p. 1-8.
12. Vinayakumar R, Soman KP, Poornachandran P. Applying convolutional neural network for network intrusion detection. Proceedings of 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI); 2017 Sep 13-16; Udupi, India. p. 1222-8.
13. Swapna G, Soman Kp, Vinayakumar R. Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals. Procedia computer science 2018;132:1253-62.
14. Gao T. Chest X-ray image analysis and classification for COVID-19 pneumonia detection using deep CNN. medRxiv 2020 [Epub]. https://doi.org/10.1101/2020.08.20.20178913.
15. Ismael AM, Sengur A. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl 2021;164:114054.
16. Maghdid HS, Asaad AT, Ghafoor KZ, Sadiq AS, Khan MK. Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms [Internet]. Ithaca (NY): arXiv.org; 2020 [cited at 2020 Jan 28]. Available from: https://arxiv.org/abs/2004.00038.
17. Phillips NA, Rajpurkar P, Sabini M, Krishnan R, Zhou S, Pareek A, et al. Chexphoto: 10,000+ smartphone photos and synthetic photographic transformations of chest x-rays for benchmarking deep learning robustness [Internet]. Ithaca (NY): arXiv.org; 2020 [cited at 2020 Jan 28]. Available from: https://arxiv.org/abs/2007.06199.