Application of GPU accelerated Deep Learning Neural Networks for COVID-19 recognition from X-Ray scans

Miroslav Nikolov, Georgi Tsenov, Ognyan Nakov, Milena Lazarova and Valeri Mladenov
Technical University of Sofia
8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria
miroslav.nikolov@novotika.com, {gogotzenov, nakov, milaz, valerim}@tu-sofia.bg

Abstract –**In this paper we are presenting the results of creation and training of a stand-alone expert system aimed at detection and diagnosis of COVID infection, that is based on automatic readings of X-ray imaging X Ray, which determines whether the patient has COVID pneumonia. The system is realized with deep learning neural networks and is accelerated with GPU utilized, instead of CPU.**

Keywords– **Deep learning Neural networks, pathogen recognition, GPU accelerated image recognition.**

I. INTRODUCTION

The corona virus pathogen became a real problem on a global scale in the recent years. For ensuring an effective diffusion control of corona virus illness (COVID-19), the supervision of many doubtful occasions for suitable patient isolation and healing actions is of a very high significance [1], [2]. At the present time the pathogenic analyses that are made by laboratories are the most widely distributed diagnostic standardized approaches, usually assumed with the application of antigen or PCR analyses and tests and it requires an extended time interval and a high amount of false positive or incorrect negative outcomes are raised [3]. In accordance to these facts quick and precise diagnostic techniques are now required for more efficient wrestling with this illness, yet in the future by introducing of novel and altered COVID-19 strain forms which influence even the people immunized to the present strains. Representations from radiologists are available, and they have stated effect on the lungs caused by COVID-19, observable on X-Ray images of the chest. A possibility is offered with this to generate and learn an expert system, thus ensuring a possibility to quickly detect and classify certain image properties and features. Founded on radiographic alterations of corona virus in the respective image scans, an in-depth training technique might be evolved that can obtain the visual features of COVID-19 appearance or influence for providing a medical diagnosis before the pathogen analysis, which ensures crucial controlling time saved. Owing to current developments of artificial intelligence in the present era and due to the massive application of accelerators to quicken the training computing it's feasible to generate a system constructed of artificial neural nets rapidly and by significantly reduced related costs [4], [5], [6], [7], [8]. That is conceivable to the application of Deep Convolutional Neural Networks (NN), that may be used as a strong and efficient instrument suitable for extraction of appropriate features and properties from a big-sized data set for training, and which may be trained quite quickly, using GPU accelerators [9],[10],[11],[12],[13], [14]. Applying such solution, the doctors or approved employees can detect COVID patients and prescribe proper therapy in the former illness stages, resulting in improved patient survival probabilities or avoiding the hardest disease form. The medics can derive precise reports in very short time since the trained model doesn't need great computing resources.

In this work we present a solution for COVID19 pathogen detection that is based on analysis of photos of chest x-ray imagery scans. This work is a continuation of a previous methodology where the Google AI package that is cloud based was used and where the computations were done in classical fashion with CPU, while in this work we present the results where an autonomous Graphical Processing Unit (GPU) is used. In the last decade the GPUs became much more sophisticated and powerful than CPUs and capable to be used as compute unit devices and not only as accelerators for graphics. With the GPUs from the last generations an option to use a compute device with thousands of tens of thousands of compute units is possible, which makes training and deploying deep neural networks in GPU hardware so much faster than CPU and allows even real time operation. In this work we have trained and deployed and used a deep neural network with 34 layer structure, which provides more than satisfactory results, with more than 90 percent successful recognition rate and in real time. This opens the opportunity for real-time COVID19 detection, which is not only fast, but also non-invasive. The trained system is applicable for implementation on a lower end embedded hardware with GPU silicone, paired with imaging device and provides also a significant cost reduction, because it can substitute the expensive clinical PCR and antigen tests, paired also with the case that even vaccinated patients can become sick again and serve as COVID19 carriers and with the applicable on time real-time virus detection and with decreased costs the number of screenings can be increased resulting in high tractability.

This work is prepared in the following way. In chapter 2 the authors describe the used methods and model architecture of Deep Convolutional Neural Network and the types of applied data sets for training and testing the neural network. The experimental results are presented in Chapter 3. The final Chapter 4 concludes the paper.

II. THE USED METHODOLOGY AND DATASETS

Many novel models for artificial intelligence implementations have their basis on Deep Learning (DL) that in succession have basis on artificial NNs, and especially convolutional NNs. They can also involve formulae for suggestions or hidden quantities arranged in layers in deep generative models, like nodes in deep NNs and deep Boltzmann machines. When using DL and training, each layer level has to be learned to convert its inputs to a little more abstract and complex depiction. For instance, in applications for image detection where an object under analysis is a face, the severe input data could be a set of image pixels, and the primary representative region can abstract the picture elements and code the boundaries; the second layer combines and encodes edge sequences; the third region encodes the eyes and nose; the final fourth region recognizes if the analyzed picture includes a face. Then the most significant point in the procedure is that DL training may make decisions that functions to use effectively at this level autonomously and where also the different number of regions and their dimensions can offer various degrees of abstraction. To some extent DL is related to the number of layers through which the data are converted and concretely these systems are related to an important depth of the credit division route that is the chain of conversions from the input to the output. This illustrates potential underlying relations between the output and the input. There is no generally approved depth value that splits shallow learning and DL, and many scientists confirm that deep learning includes a deepness larger than 2 and formerly a threshold of 2 has been presented as a generalized approximating tool meaning that it may represent all math functions [15].

Application of artificial intelligence in the medical sciences is related to the application of artificial intelligence methods, automatic procedures in analysis and therapy of patients which need of medical attention and care. Even though determination and therapy look as unpretentious stages, many different background processes for a patient to be cared correctly are required, for instance:

- Using data assembling applying patient talks and analyses
- Results analysis, processing and their investigation
- Providing an accurate diagnosis by the use of many data sources
- Applying a suitable method for therapy (frequently used possibilities)
- Preparing and administrating of the applied technique for therapy
- Observation and monitoring of patients

Follow-up concern, subsequent inspections, etc.

For modeling a system founded on chest scans based on X-ray images (roentgenogram), training Computed radiography and digital radiography analyses of chest imagery is obligatory. We formed an model that makes a decision if a human is positive for COVID founded on convolutional deep NN, which is trained for roentgenogram representation of lungs with lesions from image data sets, derived by data obtained from diseased people with COVID and from healthy people with no previous infections and disabilities. That method permits a generation of a system, applying hardware combined with a suitable software performing analysis and inspection of lungs roentgenograms, with data organized in training, test and validation groups.

To implement a model of a system for training and recognition of X-ray images in patients with COVID infection we used a specialized medical workstation with Windows, Intel Core i7 Processor, RAM 32 GB, GPU NVIDIA GTX 1080ti 11GB, medical diagnostic monitor WIDE, model CX20N - 2 MP. The used software environment was Anaconda, Web Jupiter browser access. We applied the Fastai deep learning library, Fastbook, programming language Python. We apply a dataset of COVID 19 X-rays readings and other ones verifying that those images are related to patients with diseases. The data are arranged on the local disk of a computer in the presented file structure.

Local_path/train/covid

Local_path /train/nocovid/

Local_path /validation/covid/

Local_path /validation/nocovid/

Local_path /test/

The clinical images from the data sets grouped for validation and training were separated into nocovid and covid sub-groups. The main purpose of this paper is COVID detection and storage of each member from the confirmed cases for this group in a covid class folder and collect each different pathogen as different Pneumocystis, ARDS, SARS types and others in nocovid group directory. No sub-directories exist in the testing folder and it is used for storing pictures that the considered system was not introduced to and we analyze them for establishing what the system investigates and reporting as resultant outcome. Hence negative and positive images of COVID exist in the described directory. The total dataset size is 7232 images. The applied dataset contains for the training totally 3616 positive of COVID 19 roentgenogram scans of the chest area, and the rest 3616 roentgenogram digital image scans of humans with no COVID 19, that is resulting in a final number of 7232 clinical images applied. The fastai, fastai.vision and fastai.metrics libraries are applied in the procedures. The applied prototype of the Deep Neural Net is founded on architecture Resnet34 Residual Network - models.resnet34 with a deepness of 34 regions [16].

We present and describe the technique of collecting data applied and prepare a check by visualizing the method if it is presented and properly available as exposed in Fig. 1.

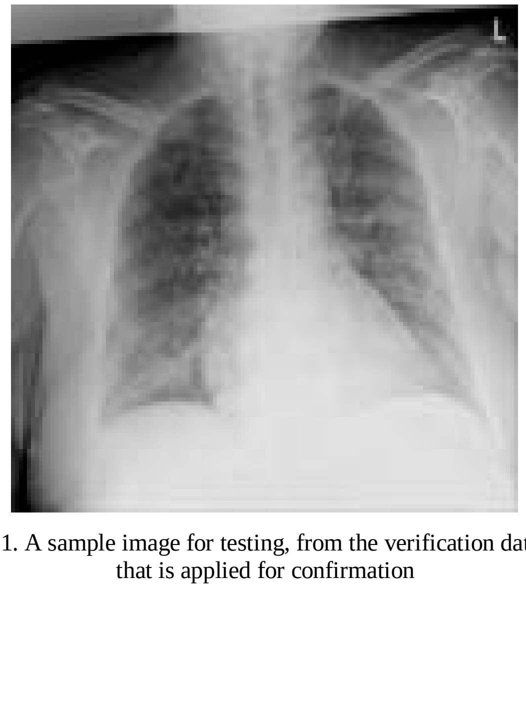

Fig. 1. A sample image for testing, from the verification data set that is applied for confirmation

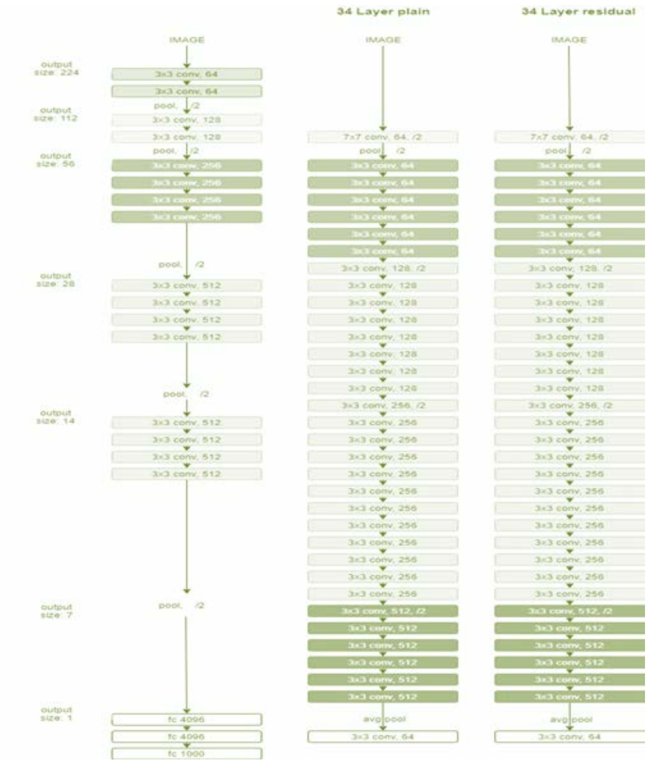

Fig. 2. Resnet34 structure of the applied and used Convolutional Deep Neural Network Architecture

III. Experimental results

The proposed considered Convolutional DL neural net is generated and tuned by applying the described parameters.

All of the roentgenogram image scans from the datasets are normalized at 224 by 224 picture elements resolution. Then the adjustment of the consignment parameter size bs is available. The batch dimension arranges the number of images that may be applied on the model at one iteration step on only one-time range. The batch is the number of images for each class that will be transmitted to the deep learning NN. It is to be noted, that when the training computing hardware does not include sufficient memory size, bs 16 or bs 32 can be applied and for that purpose primarily we are determining the dimension of the batch to be bs = 16 and that is one of the structure's hyper parameters.

Then follows a generation of an entity of type get_transforms, that permits the growth of the modified dataset created using the roentgenogram image scans of the training dataset. The accepted images are with a maximal spin of up to 25 degrees and horizontal spin. The procedure makes possible the decrease of overfitting and overtraining, since even though they are adapted from the original pictures, they are reasonably different one to another.

Then follows the loading of the pictures into an ImageDataBinch object. The Fastai permits direct loading of images from a directory, in order to determine the number of images in a batch, to define the training images, and these for validation and testing, to prepare previously defined changes and finally to make all images with given dimensions. Finally, by using the Show_batch function the number of columns and rows for representation and their determined dimension can be specified.

Then the functions for Loss are determined. Validation loss provides similar measure as the training loss, but it isn't applied for updating the weights and is computed in the same manner - by the use of the network forward over the input signals xi and by comparison of the net's outputs $Y_i^{\wedge}$ with the ground truth values Y_i applying a loss function:

$$J = \frac{1}{N}\sum_{i=1}^{N} L(Y_i^{\wedge}, Y_i), \quad (1)$$

Here, L is the specific function for loss founded on the variance amidst forecast values and the targets. The losses from validation are not applied in the weights update procedure. The basic goal of applying a dataset which is not applied in the train phase, is the establishing how the model is generalized to newer data as input signals. It is not reliably to apply the training data in the same manner for checking the generalization since the model may deliver wrong good rate of recognition due to overfitting. Applying validation datasets is providing a technique for observation and helps regulation for avoidance of overfits. NNs usually apply a validating dataset on every epochs, since a long-time training may lead to over-fitting, and the models do not improve, they just deteriorate their form. In that way a lot of wasted energy for observing validation can be reduced, and break training if it has not enhanced for a long-time intervals passed, or continues getting worse.

In the end is checked the generated data sets - data.c, len (data.train_ds), len (data.valid_ds), len (data.test_ds), len (data.classes) - (2, 154, 80, 83, 2). As a result 7232 pictures are available, with depicted in Fig. 3 some representatives.

The generation and adjustment of the classifying tool is completed as follows. When the data are organized, we generate and tune the classifier by the use of pre-trained models for deriving the respective features of the images. Various trained models are available, and the method used is noted as transfer training.

A learning item of type cnn_learner is generated (NN with convolutional layers), and it is provided with the dataset entity and is programmed for using the previously trained model of resnet 34 and the error_rate measure with the dataset for validation. Then the model is trained applying a technique known as fit_one_cycle - learn.fit_one_cycle, and follows the procedure for train as presented in Fig. 4. There are some parameters observed, as Epochs (number of the iteration), tuning and validation datasets loss function, the value of mean squared error and the passed time since beginning. That is the number of periods the data goes through the deep neural net in a straight forward and then in backward route, consequently using weights' update. In many cases, that may cause over-training, as we sweep the same pictures and simply identify them, and not training with newer data.

Finally, a reasonably well-trained model is derived, and the respective error rate is about 10%, with effective recognition rate of 90%. For using the trained model again, it is saved in a new learn.save folder.

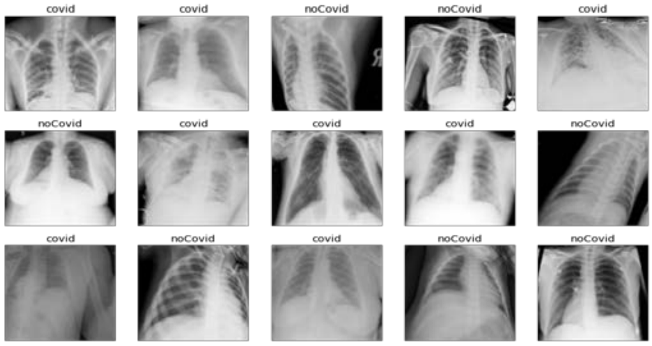

Fig. 3. Illustrative images obtained from the data set for training

epoch	train_loss	valid_loss	error_rate	time
0	0.288974	0.363103	0.175000	06:37
1	0.220784	0.272031	0.125000	06:39
2	0.169330	0.244056	0.087500	06:37
3	0.166766	0.275936	0.100000	06:38
4	0.147461	0.231211	0.100000	06:37
5	0.133777	0.282941	0.087500	06:38
6	0.131000	0.266385	0.112500	06:38
7	0.123028	0.332749	0.062500	06:38
8	0.113200	0.252950	0.075000	06:38
9	0.098253	0.199432	0.037500	06:38
10	0.107682	0.778493	0.187500	06:37
11	0.087630	0.433904	0.125000	06:37
12	0.113910	0.388979	0.100000	06:37
13	0.107630	0.759643	0.200000	06:37
14	0.096153	0.497093	0.175000	06:37
15	0.085709	0.378146	0.137500	06:36
16	0.073563	0.416728	0.125000	06:37
17	0.075314	0.295989	0.100000	06:38
18	0.055412	0.223292	0.037500	06:36
19	0.061284	0.610371	0.200000	06:39
20	0.052270	0.513481	0.137500	06:37
21	0.049090	0.465645	0.162500	06:37
22	0.046305	0.420856	0.112500	06:38
23	0.036703	0.358024	0.112500	06:39
24	0.051594	0.388113	0.137500	06:38
25	0.032964	0.275369	0.100000	06:38

Fig. 4. The learning training process of the Convolutional Deep Neural Network

For interpretation the outcomes, a ClassificationInterpretation object on the training entity is generated. We then apply plot_top_losses, it gives us a possibility to prove how and whether the forecast is correctly realized.

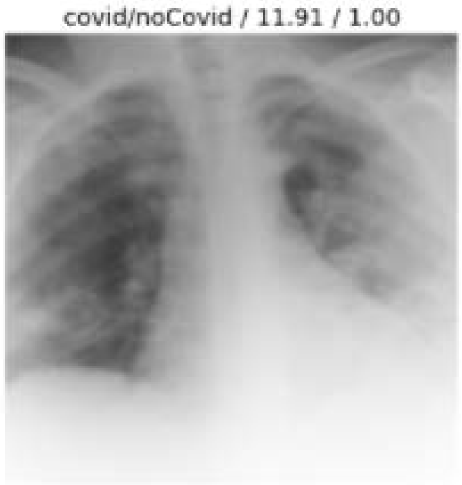

Fig. 5. Generation of an item for explanation of the derived results

The results of four values of prediction / current / loss / likelihood for the present group are obtained. This approach allows us to verify where the tuning fails. The correspondent confusion matrix is represented. The diagonal illustrates the extent of accuracy of one of the cases and is presented in Figure 6.

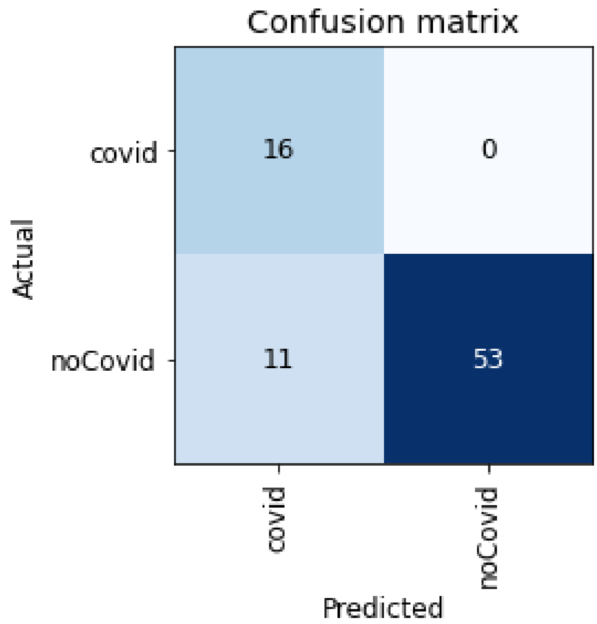

Fig. 6. The confusion matrix representing the correctness of our examples

Applying the finally trained model visual representation of the outcomes of the learned Deep Convolutional Neural Network may be completed.

For testing if our model of image detection system with or without COVID 19 operates with images it has never used, we have generated function buttons. Using these buttons, we can load an X-ray image unfamiliar to the system and a button to diagnose and check the same. We apply a Modality DX imaging analysis to check if the human is infected with COVID virus. And the result from the diagnostic imaging is that the correspondent human has COVID pneumonia as presented in Figure 7.

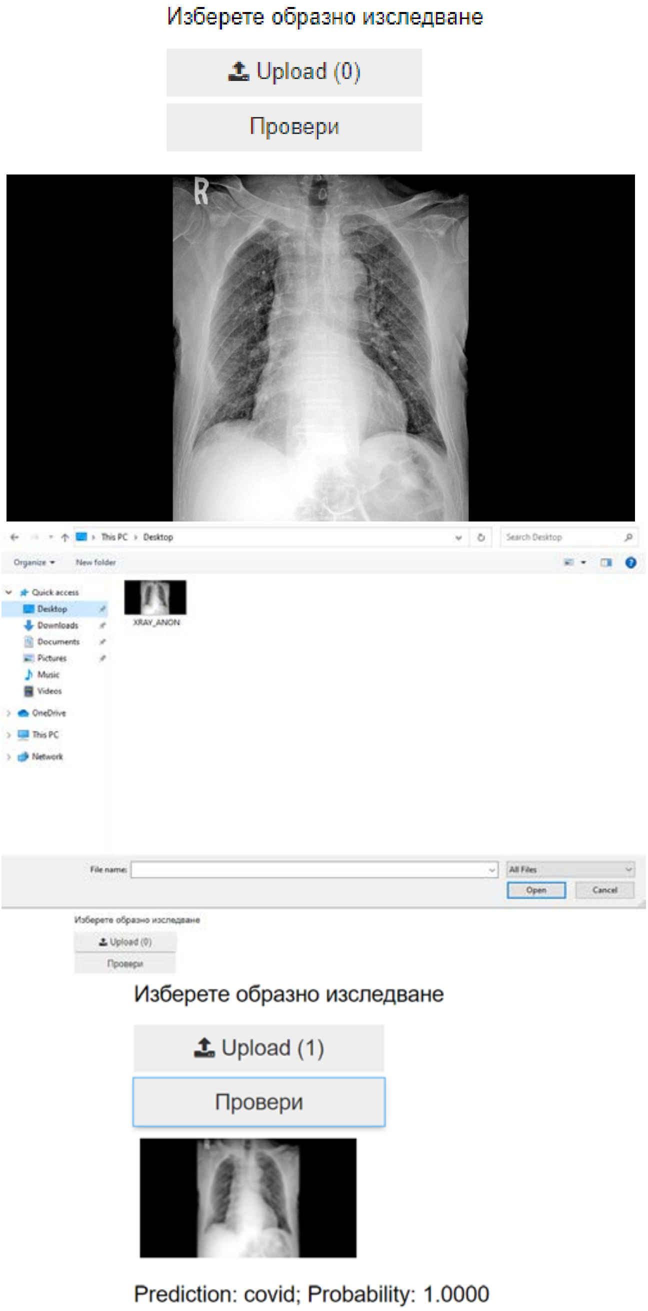

Fig. 7. Last model outcomes interpretation when testing with image from test data set

CONCLUSION

In this paper we present the outcome and recognition success rates of an expert system, which is making a decision if a human is COVID-19 positive, by applying GPU acceleration. This solution basis is on a Convolutional Deep Learning Neural Networks learned from roentgenogram lungs images with lesions and using data sets derived by infected with COVID-19 or healthy people without infections and disabilities. The described approach ensures the opportunity for generation of a system for fast COVID recognition.

ACKNOWLEDGMENT

The research presented in the paper is funded by the European Regional Development Fund, Operational Program “Science and Education for Smart Growth” under project UNITe BG05M2OP001-1.001-0004/28.02.2018 (2018–2023).

REFERENCES

[1] Singh, Jaspreet and Singh, Jagandeep, COVID-19 and Its Impact on Society (April 3, 2020). Electronic Research Journal of Social Sciences and Humanities, Vol 2: Issue I,

[2] Wu He, Zuopeng (Justin) Zhang, Wenzhuo Li, Information technology solutions, challenges, and suggestions for tackling the COVID-19 pandemic, International Journal of Information Management, Vol. 57, 2021, ISSN 0268-4012

[3] Manoucher Teymouri, Samaneh Mollazadeh, Hamed Mortazavi, Zari Naderi Ghale-noie, Vahideh Keyvani, Farzaneh Aghababaei, Michael R Hamblin, Ghasem Abbaszadeh-Goudarzi, Hossein Pourghadamyari, Seyed Mohammad Reza Hashemian, Hamed Mirzaei, Recent advances and challenges of RT-PCR tests for the diagnosis of COVID-19, Pathology - Research and Practice, Vol. 221, 2021, ISSN 0344-0338

[4] K, Gurney, "An Introduction to Neural Networks", 1st Edition, CRC Press, 2018, ISBN-13: 978-1857286731

[5] C.C.Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, 2018, ISBN-13: 978-3319944623

[6] A. Glassner, Deep Learning: A Visual Approach, No Starch Press, 2021, ISBN-13: 978-1718500723

[7] Talebi H. and Milanfar P., Learning to Resize Images for Computer Vision Tasks., 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 487-496.

[8] Yadav, S.S., Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis, Journal of Big Data, Vol. 6, pp. 1-18, 2019, DOI: 10.1186/s40537-019-0276-2

[9] Quaiaa E., Baratellab E., Crimia F., Cancianc L., Crivellid P., Vianello A., High-resolution CT features in patients with COVID-19 pneumonia and negative nasopharyngeal and oropharyngeal swabs, Pulmonology Journal, Vol. 27. Issue 4, pp. 351-353, Elsevier open access, 2021.

[10] Zhao, Jinyu & Zhang, Yichen & He, Xuehai & Xie, Pengtao, COVID-CT-Dataset: A CT Scan Dataset about COVID-19, 2020, arXiv:2003.13865v3

[11] Chang-Eun Park, Diagnostic Methods of Respiratory Virus Infections and Infection Control, Korean Journal of Clinical Laboratory Science, Vol. 53, Issue 1, pp. 11-18, 2021, ISSN: 2288-1662

[12] Jeff Heaton, Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks, CreateSpace Independent Publishing, ISBN-13:978-1505714340, 2015

[13] Mohamed Elgendy, Deep Learning for Vision Systems, Manning Publications, ISBN13 9781617296192, 2021

[14] X. Jiang, A. Hadid , Y. Pang , E. Granger, X. Feng, Deep Learning in Object Detection and Recognition, Springer, ISBN-13: 978-9811051517, 2019

[15] Daniel Roberts , Sho Yaida and Boris Hanin, The Principles of Deep Learning Theory: An Effective Theory Approach to Understanding Neural Networks, Cambridge University Press, ISBN13 9781316519332, 2022

[16] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, Conference on Computer Vision and Pattern Recognition IEEE 2016, pp. 770-778, DOI: 10.1109/CVPR.2016.90.