Expert System for Analyzing CT Images using CNN with a Self-Attention Module*

Artur Kostera[1,*,†]

[1] *Faculty of Applied Mathematics, Silesian University of Technology Kaszubska 23, 44100 Gliwice, POLAND*

Abstract

Expert systems in medicine support doctors in analyzing medical data. Such a system is based on artificial intelligence methods, where expert data is used to train the model. In this paper, we present the architecture of an expert system for the classification of computed tomography (CT) images in the detection of COVID-19. The architecture of a convolutional neural network was proposed, including the self-attention module to focus on specific features in CT images. The proposed methodology has been described and tested on a publicly available database to demonstrate its effectiveness. The model reached 94% of accuracy according to the validation set.

Keywords

CNN, self-attention, classification, covid-19, expert system

1. Introduction

Artificial intelligence (AI) is an innovative technology that has become extremely popular in recent times. One of its features may be its adaptation to various tasks and the possibility of obtaining very good results compared to other, classic methods. AI has been used for a long time as part of small aids such as image sharpening in a photo via mobile applications [1]. Over the last few years, thanks to ChatGPT [1] and generative or linguistic models [2], it turned out that artificial intelligence has enormous applications and potential for development and implementation in many areas of life. Attention should also be paid to the enormous technological progress that has resulted in the expansion of applications for various sensors and objects, i.e. the Internet of Things [3]. As a result, Smart Cities [4] are being built, where automation is a basic element in collecting data from the environment, immediate analysis and undertaking operations. The literature also shows that federated learning plays a large role [5]. The new way of training classifiers is important, especially when the data should remain private. Additionally, distributed learning and sharing only weights allow for increased security in the solutions created [6].

Thanks to such solutions, dealing with certain problems is no longer as time-consuming as it used to be. First of all, automation and minimizing the amount of work for humans contribute to the improvement of human life. Many machines used on construction sites or in factories do not require much human involvement. More autonomous vehicles [7], detection systems [8] and various analyses [9, 10, 11].

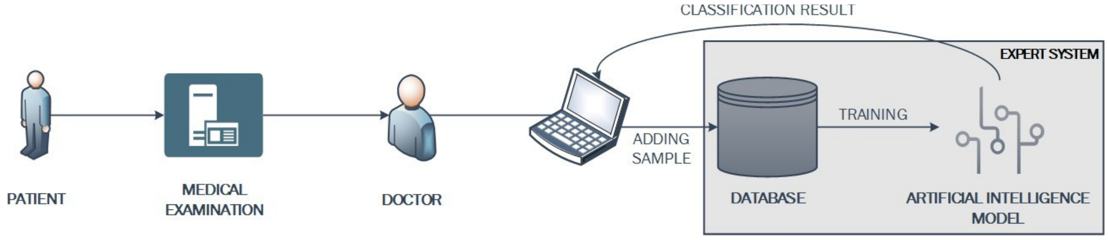


Figure 1: Visualization of the operation of the expert system to support medical doctor's decision-making

IVUS2024: Information Society and University Studies 2024, May 17, Kaunas, Lithuania

1,* Corresponding author

† These author contributed equally.

ak307879@student.polsl.pl (A. Kostera)

Medicine is an important and very delicate science where the slightest mistake can cost a human life, and AI is also starting to support people in various duties [12, 13]. Artificial intelligence analyzes patient data, prescribes appropriate amounts of drugs, can provide a second opinion to doctors, analyzes and processes data received from various devices such as tomographs and can even help doctors during operations. Based on the analyzed literature.

The development of artificial intelligence methods allows the creation of expert systems, i.e. systems based on expert knowledge and able to make decisions. Such a decision cannot replace the doctor, but it can support him. An example is a system using fuzzy logic and rule-based reasoning [14]. Another example is a system [15] based on a convolutional network, where knowledge transfer was used to detect disease lesions. The presented idea is presented as an engine in an expert system. Another example is a framework for an expert system that uses ensemble techniques for classification purposes [16]. Based on these observations, in this paper, we propose a medical expert system architecture. The system is designed to support the doctor in deciding whether the disease occurs. Convolutional networks were used for this purpose. The main contributions of this paper are: an expert system architecture and convolutional neural network with a self-attention module for CT analysis.

2. Methodology

In this section, we present the architecture of the expert system model. Then, the mathematical model of the classifier, which is a convolutional neural network, is described, taking into account the self-attention module.

2.1. Expert system

The application of artificial intelligence is mostly an expert system. Such a system is created to support an expert and is created based on his/her knowledge. In general, a classic expert system is defined as a rule-based model, but it can be also modeled with classifiers. It can be justified through non-numerical data, an example of which is graphic data. Moreover, attention should be paid to the issue of practical application, because the use of a neural network as the main engine of the expert system allows for returning the classification result, which is understood as an additional opinion. The task of this system is to diagnose the patient based on his brain scans taken by a CT scanner. Here, AI decides based on the scans whether the patient has COVID-19 or not. Correctly diagnosing the patient is crucial during the entire examination. An error at such an important stage of treatment can have disastrous consequences, such as prescribing inappropriate medications that may endanger the patient's life or lead to unnecessary surgeries. Hence, decisions regarding a patient's health are usually made by an experienced doctor who, thanks to years of study and accumulated knowledge, can truly diagnose the patient.

In the modeled system, the patient would be scanned by a doctor using a CT scanner. The doctor would then check the received data and describe it as belonging to a sick or healthy person. The database created in this way would be fed to the classifier in question, which would learn during the training process based on the knowledge provided by doctors. Of course, if the set is expanded with new medical data, the learning process can be repeated. Training the AI model would be repeated until the doctor concluded that the system had achieved sufficient effectiveness. After each training process, the doctor has access to the trained model through the application, which provides him with the opinion of a second expert. A visualization of such a model is shown in Fig. 1.

2.2. Neural network classifier

Neural networks are mathematical models inspired by the functioning of the human brain that receive various data and then process them [17, 18]. The main idea of a neural network can be described by defining the operation of a single neuron because the network is a set of such units arranged in columns called layers. Assume that a given neuron will process the value x_i from the i-th neuron through the connecting synapse. We describe such a synapse with a weight value of w_j. An analyzed neuron receives two values: signal x_i from the previous neuron and the weight value, which allows us to define the output as follows:

$$y = f(w_j \cdot x_i), \tag{1}$$

where $f(\cdot)$ is a function that scales the multiplication result to the appropriate range specified by the function, e.g. ReLu.If there are more neurons in the previous layer, the above formula takes the form:

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i\right), \tag{2}$$

where n is the number of all neurons in the previous layer.

Convolutional Neural Networks (CNN) is a newer network architecture that allows processing and analyzing images [19, 20, 21]. Architecture is also built on layers. The first type of layer is the convolution one, which is based on modifying the image to extract features. The layer has a set of weights called a kernel or filter, usually, such a kernel can cover only a small part of the entire image at a time. The filter, starting from the upper left corner of the image, moves one pixel at a time to eventually cover the entire image. Covering each pixel involves modifying its value, which is accomplished by analyzing its neighbors, i.e. the previously mentioned

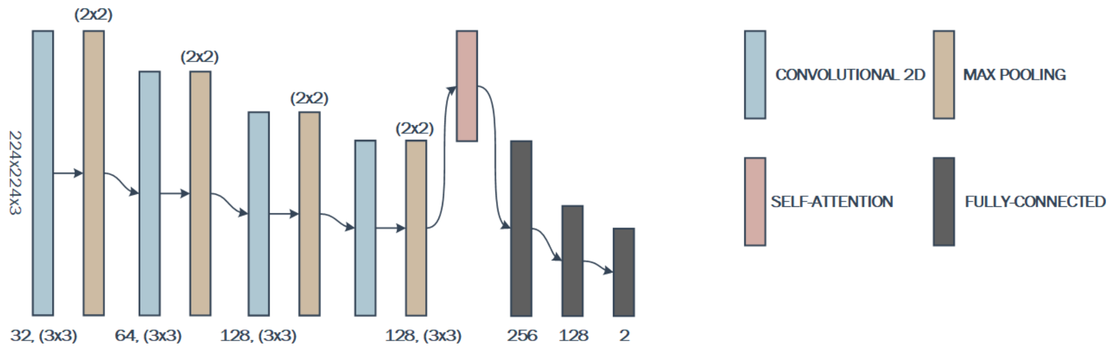


Figure 2: Visualization of the proposed convolutional neural network model for medical data classification

neighboring pixels. The modification using convolution will be described as follows:

$$(f * g)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(i, j) \cdot g(x - i, y - j), \tag{3}$$

where I is an image, x and y are a coordinates of a given pixel on image I, k is the size of neighborhoods, g is a kernel/filter.

The second type is the pooling layer, whose task is to minimize the image size while maintaining the most important features of objects in the images. The idea of action can be described as selecting a specific value at a given position relative to the neighborhood. As an example, we take the size of the analyzed neighborhood as 2×2, then the pooling layer selects the value that the pooling function indicates. The most commonly used values are maximum, minimum, or average. After selecting a given value, the layer analyzes the next position in the image. The third type of layer is the fully connected layer, which is composed of neurons described in Eq. (2).

For the expert system being built, a neural network model consisting of 14 layers was built. The input layer accepts images of dimension 224×224. Then the image from the input layer is passed to the convolutional layer, which has 32 filters of size 3×3 to focus on extracting relevant elements. This layer uses the ReLU function, i.e. rectified linear unit. The function used is based on leaving positive values and replacing negative values with 0. The next layer is Pooling with a maximum function, which focuses on removing data with the smallest values. Thanks to this process, the amount of calculations that the program still needs to perform is once again reduced. In the next steps, a set of layers is added three times: convolution and pooling. The difference is the number of filters in the convolutional layers, which are increased to 64, 128 and 256. The returned result from the fourth pooling layer is transferred to the attention module, whose task is to mark individual image elements with weights that will allow the identification of important features (the description of the module can be found in Sec. 2.3). The resulting tensor is flattened to pass it to dense layers, i.e. fully connected. The network model has three fully connected layers of 256, 128 and 2 neurons. The last layer has only two neurons due to binary classification, i.e. returning values 0 or 1, meaning "the patient is sick" or "the patient is healthy", respectively. The first two dense layers use the ReLU function as an activation function, and the last one uses the softmax function, which, based on the information received from the previous layer, determines probability values regarding the class of the processed image. The model is described also in Fig. 2.

Training involves modifying weights in the entire model, including filter values in convolutional layers. For this purpose, the ADAM algorithm is used [22], which is based on the modification of values by determining statistical values such as moments of a selected order. To check the performance of the classifier, the classification accuracy as well as the loss value are verified. This second metric is defined depending on the selected problem. For the binary problem, we use the cross-entropy function defined as:

$$L = -\left[y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right], \tag{4}$$

where y is true label and $\hat{y}$ is predicted label. However, in the context of analyzing a larger number of samples, here batches, then this function takes the following form:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y_i})\right], \tag{5}$$

where N is the number of samples in the batch. The entire process should be repeated until the network achieves satisfactory effectiveness. The learning process is performed for a specified number of epochs or until previously set loss function values or accuracy are obtained.

2.3. Self-attention module

The attention module is an element of the neural network that allows you to add weighting to all values in the processed sample. For this purpose, the weights will have values in the range $\langle 0, 1\rangle$ and will be interpreted as the significance of a given value. To describe this module, it should be assumed that the weight values will be marked with a matrix $W \in \mathrm{R}^{D\times 1}$, where D is the size of the processing matrix (sample) $X \in \mathrm{R}^{N\times D}$ (N is the number of samples in batch). The first step is to determine the attention score relative to the matrix, and the weights and rescaling by the softmax function:

$$A = \mathrm{softmax}(XW). \tag{6}$$

A weighted sum is then performed along the axis elements of the given feature as:

$$S = g(X \star A), \tag{7}$$

where $\star$ means multiplying values as the Hadamard product.

3. Experiments

The tests used a database of computed tomography images from COVID10 and healthy patients [23, 24]. The data set consisted of 1252 scans of patients with confirmed coronavirus and 1229

(a) Covid

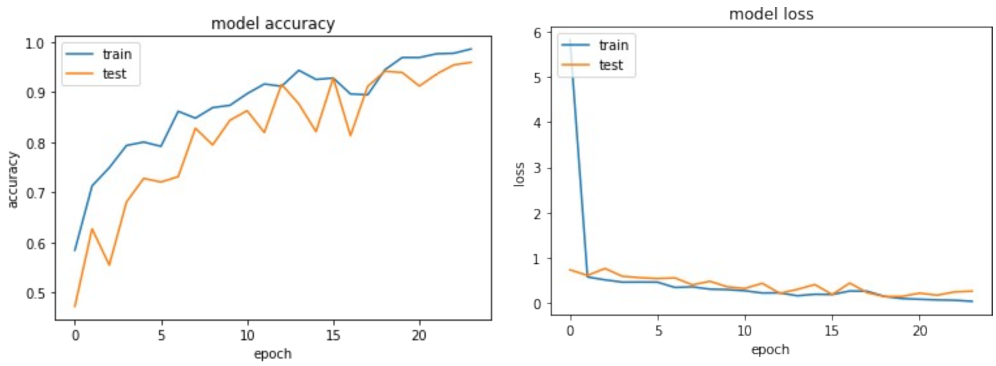
(b) Non covid

Figure 3: Selected samples representing the class of a COVID-19 patient and a healthy patient.

scans of healthy patients (see representative images in Fig. 3). The set was randomly divided in a 70:30 ratio, where 70% of the data was used for training purposes and the remaining 30% for classifier validation. As a result, 1736 samples were obtained in the training set (876 photos assigned to Covid) and 745 in the validation set (376 samples belonged to sick patients and 369 to healthy patients). All tests were performed on a computer with the following parameters: NVIDIA GeForce RTX 4060 Ti, AMD Rayzen 9 7900X 12-Core and 32GB RAM.

The training process was set to 100 epochs with callbacks in case of no changes in values. This caused the training process to stop after 27 iterations. In Fig. 4a the effectiveness of the system during learning and testing is presented. The blue line represents effectiveness during the learning process. It can be noticed here that within the first five epochs, the system quickly achieved an efficiency of approximately 80%. Then it rose smoothly with small jumps. Around epoch 10, the classifier reached a value close to 90%. Subsequent iterations brought small jumps in the values, which ultimately allowed us to achieve 98% accuracy for the training set. The orange line shows performance when testing on the validation set. During the first 5 epochs, he achieved a level of efficiency of 70%, then the effectiveness increased until epoch 10. After achieving effectiveness of over 80% at epoch 10, for the next 7 epochs, the effectiveness was measured in the range of 80% up to 90%. After the 17th epoch, the spikes in the values were reduced and the accuracy finally reached 94%. Analysis of the accuracy graph shows a very good adaptation of the classifier to the training set and, in later iterations, to the validation set.

In Fig. 4b, the loss value was shown. For the training set, we can see that the system started to decline rapidly within one epoch. After this one jump, the entire graph is smooth. Over the following eras, the number of losses continues to decrease slowly. Only after the 15th epoch, i.e. at the point where we saw a drop in effectiveness in the previous chart, losses increased slightly and then drop again, ultimately reaching a result close to zero. In the case of the validation set, the loss value decreases with subsequent iterations. The jumps are not as large as in the case of effectiveness. The largest jumps in this graph can be seen in epochs 10 to 17, precisely during these epochs the largest jumps in effectiveness took place in the previous graph. Eventually, the graph reaches values close to zero. Comparing the loss and efficiency graphs, we can see that in both cases, the model has a smoother graph during training. Despite the differences, the analysis of the classifier against the validation data allows us to note that ultimately the classifier was quite well adapted to these data as well.



(a) Accuracy to epochs. (b) Loss values to epochs.

Figure 4: Charts of the loss function and accuracy of the classifier during the training process.

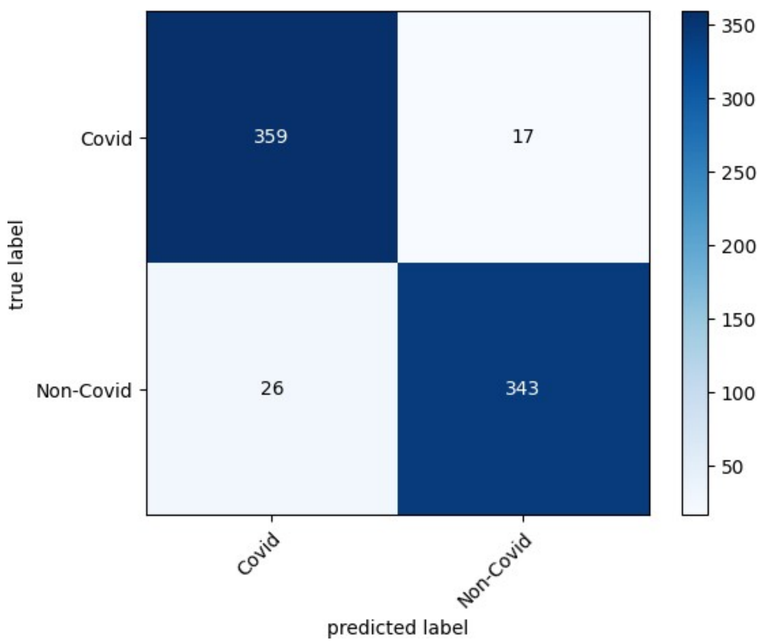


Figure 5: Confusion matrix for the validation set.

A confusion matrix was also generated for the trained model and the validation set. It is presented in Fig. 5. Lines marked as "true label" determine whether a given photo shows a scan of a person suffering from COVID-19 or not. In turn, columns marked as "predicted label" indicate how the system assessed a given photo. Combining these two pieces of information, we know that the number in the first field, i.e. the field that belongs to both the Covid row and the column with the same name, means cases in which the system received a photo of a sick person and diagnosed the patient that way. The same, only for photos of healthy patients, applies to the number contained in the field belonging to the Non-COVID row and the column of the same name. The remaining two places are the number of situations when the system made a mistake, i.e. it received a photo of a sick person and recognized it as healthy or vice versa. Additionally, we also see that the intensity of the blue color means how often the model made a given decision. Based on the numbers and color intensity, we see that in most cases the model responded correctly. By comparing the number of correct answers with incorrect ones, we obtain a level of accuracy of 94%. In terms of other statistical values, the classifier was evaluated and reached a sensitivity metric of 0.93, specificity of 0.955 and precision of 0.953. Such metrics indicate good classification results in terms of classification of COVID and non-COVID samples.

The proposed model in this paper is based on CNN with a self-attention module that can help in focusing on different features. As a consequence, the classifier can learn the features of a given class faster. In terms of comparison with the state of the art, 91% of accuracy was achieved by [25], which shows that the self-attention module can be an interesting tool in terms of CNN. It should be noted that the achieved values of evaluation metrics are high, although not ideal. To increase them, transfer learning or even feature fusion can be used.

4. Conclusion

The construction and analysis of new artificial intelligence models are important because we are looking for more and more accurate solutions. In the case of expert systems, knowledge and purpose may change, so adapting such architecture to newer solutions is crucial. This paper presents an expert system model using a neural classifier. The main architecture was based on creating knowledge in terms of medical image and its label. Such a system was composed of a convolutional neural network with a self-attention module. The addition of self-attention allowed the classifier to draw attention to important features of medical images. This is a kind of directing the classifier to weigh the pixels, which allows for achieving a very good classification result. In terms of accuracy metrics, it was over 94% on the validation set. This result allows the solution to be used as a system supporting the doctor's decision in the detection of COVID-19 disease.

In future works, we plan to extend expert system architecture to an end-to-end model that could be used in a more flexible way. Also, it seems important to extend the target of the system to other diseases, which will allow for the expansion of its operation.

5. Acknowledgments

This work was supported by the Rector's mentoring project "Spread your wings" at the Silesian University of Technology.

References

[1] Y. K. Dwivedi, N. Kshetri, L. Hughes, E. L. Slade, A. Jeyaraj, A. K. Kar, A. M. Baabdullah, A. Koohang, V. Raghavan, M. Ahuja, et al., "so what if chatgpt wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy, International Journal of Information Management 71 (2023) 102642.

[2] M. Petkevičiusa, D. Vitkutė-Adžgauskienėa, Fine-tuning language models for emotion recognition in lithuanian texts using neural machine translation of training datasets, Training 16 (2023).

[3] M. Saied, S. Guirguis, M. Madbouly, Review of artificial intelligence for enhancing intrusion detection in the internet of things, Engineering Applications of Artificial Intelligence 127 (2024) 107231.

[4] J. d. J. Camacho, B. Aguirre, P. Ponce, B. Anthony, A. Molina, Leveraging artificial intelligence to bolster the energy sector in smart cities: A literature review, Energies 17 (2024) 353.

[5] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, A. H. Celdrán, Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges, IEEE Communications Surveys & Tutorials (2023).

[6] D. Połap, G. Srivastava, A. Jaszcz, Energy consumption prediction model for smart homes via decentralized federated learning with lstm, IEEE Transactions on Consumer Electronics (2023).

[7] A. Zielonka, A. Sikora, M. Woźniak, Trust mechanism fuzzy rules intelligent car real-time diagnostic system, in: 2023 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2023, pp. 1–8.

[8] K. Prokop, D. Połap, G. Srivastava, Agvs quality of service throughput prediction via neural networks, in: 2023 IEEE International Conference on Big Data (Big Data), IEEE, 2023.

[9] V. Rudžionis, A. Lopata, S. Gudas, R. Butleris, I. Veitaitė, D. Dilijonas, E. Grišius, M. Zwitser- loot, K. Rudzioniene, Identifying irregular financial operations using accountant comments and natural language processing techniques, Applied sciences 12 (2022) 8558.

[10] D. Kuizinienė, P. Savickas, T. Krilavičius, Check for autoencoder as feature extraction technique for financial distress classification, in: Information and Software Technologies: 29th International Conference, ICIST 2023, Kaunas, Lithuania, October 12–14, 2023, Proceedings, Springer Nature, 2024, p. 71.

[11] K. Prokop, Grey wolf optimizer combined with k-nn algorithm for clustering problem, in: IVUS 2022: 27th International Conference on Information Technology, 2022.

[12] L. Singh, L. Kamra, Check for updates synergizing reinforcement learning for cognitive medical decision-making in sepsis detection, in: Information and Software Technologies: 29th International Conference, ICIST 2023, Kaunas, Lithuania, October 12–14, 2023, Proceedings, Springer Nature, 2024, p. 160.

[13] A. Jaszcz, The impact of entropy weighting technique on mcdm-based rankings on patients using ambiguous medical data, in: International Conference on Information and Software Technologies, Springer, 2023, pp. 329–340.

[14] R. H. Faisal, S. Debnath, M. M. U. Islam, S. Sifath, S. A. Kakon, M. S. Alam, N. Siddique, A modular fuzzy expert system for chemotherapy drug dose scheduling, Healthcare Analytics 3 (2023) 100139.
[15] A. Caroppo, P. Siciliano, A. Leone, An expert system for lesion detection in wireless capsule endoscopy using transfer learning, Procedia Computer Science 219 (2023) 1136–1144.
[16] S. Majumdar, P. Pramanik, R. Sarkar, Gamma function based ensemble of cnn models for breast cancer detection in histopathology images, Expert Systems with Applications 213 (2023) 119022.
[17] S.-C. Wang, S.-C. Wang, Artificial neural network, Interdisciplinary computing in java programming (2003) 81–100.
[18] Y.-c. Wu, J.-w. Feng, Development and application of artificial neural network, Wireless Personal Communications 102 (2018) 1645–1656.
[19] Z. Duan, X. Luo, T. Zhang, Combining transformers with cnn for multi-focus image fusion, Expert Systems with Applications 235 (2024) 121156.
[20] D. Połap, Hybrid image analysis model for hashtag recommendation through the use of deep learning methods, Expert Systems with Applications (2023) 120566.
[21] U. K. Lilhore, S. Simaiya, S. Dalal, R. Damaševičius, A smart waste classification model using hybrid cnn-lstm with transfer learning for sustainable environment, Multimedia Tools and Applications (2023) 1–25.
[22] D. Kingma, L. Ba, et al., Adam: A method for stochastic optimization (2015).
[23] E. Soares, P. Angelov, Sars-cov-2 ct-scan dataset, 2020. URL: https://www.kaggle.com/dsv/1199870. doi:10.34740/KAGGLE/DSV/1199870.
[24] P. Angelov, E. Soares, Explainable-by-design approach for covid-19 classification via ct-scan (2020).
[25] Z. Wang, Q. Liu, Q. Dou, Contrastive cross-site learning with redesigned net for covid-19 ct classification, IEEE Journal of Biomedical and Health Informatics 24 (2020) 2806–2813.