Machine Learning Technology for Neoplasm Segmentation on Brain MRI Scans

Ivan Sova[0000-0002-8972-4069], Ievgen Sidenko[0000-0001-6496-2469],
Yuriy Kondratenko[0000-0001-7736-883X]

Intelligent Information Systems Department, Petro Mohyla Black Sea National University,
68th Desantnykiv Str., 10, Mykolaiv, 54003, Ukraine
owlvano@gmail.com, ievgen.sidenko@chmnu.edu.ua,
yuriy.kondratenko@chmnu.edu.ua

Abstract. In this paper, machine learning technology for neoplasm segmentation on brain MRI scans is analyzed. This analysis allows to choose the most appropriate machine learning architecture and various preprocessing techniques to increase the precision of tumor instance segmentation. Understanding the image and extracting information from it to accomplish some result is an important area of application in digital image technology. Image segmentation has quickly found its use in medicine and specifically oncology. Precise segmentation masks may not be critical in other cases, but marginal segmentation errors in medical images may render the results unreliable for clinical use. Therefore, biomedical problems require much higher boundary detection precision to improve further analysis. Comparison of different machine learning algorithms, neural network architectures will achieve the highest accuracy of recognition and segmentation. During the comparison, a system of the U-Net architecture with additional processing methods was selected as the final model. Its accuracy reached 94%, which is a significant result compared to manual image segmentation.

Keywords: Neural Network, Instance Segmentation, Machine Learning, Deep Learning, Biomedicine, Neoplasm, Magnetic Resonance Imaging.

1 Introduction

In recent years, the field of machine learning (ML) [1] has undergone significant development, and advances in this field have influenced the development of other areas of our lives [2]. Machine learning algorithms are used everywhere: to analyze financial markets, search engines, personalize online advertising, recognize speech or handwriting, and detect online fraud - these are only a small part of what they are capable of [3, 4]. One of the important areas of ML is computer vision (CV), the technology of machine development that can detect, track, and classify objects. Segmentation in CV is the process of dividing a digital image into several segments [5]. The purpose of segmentation is to simplify and / or change the presentation of an image so that it is easier to analyze. Segmentation is used in many areas, for example, in production for indicating defects in the assembly of parts, in medicine for primary processing of images, as well as for compiling maps of the area from satellite images, etc [6-8].

2 Related Works and Problem Statement

Machine learning has found its way into medicine: for example, regression systems based on patient’s data can classify them as potentially ill, and pattern recognition systems can analyze medical images and extract certain features for further analysis [9, 10]. One of the most important applications of segmentation is the field of biomedicine. In most cases, video data is obtained by microscopy, tomography, and the like. An example of the information obtained is the measurement of organ size or even blood circulation. This application area also promotes research and provides new information such as the structure of the brain or the effects of certain medical devices.

To detect tumors in the tomographic images, the image is segmented. Computer vision segmentation is the process of dividing an image into several segments. The result of segmenting an image is a set of pixels that together form segments and cover the entire image. All pixels in a segment are similar regarding one or more properties, such as color. Neighboring segments differ significantly in such factors.

The segmentation problem is being considered on a set containing data on 110 patients from The Cancer Genome Atlas, created by National Cancer Institute and National Human Genome Research Institute, USA. The purpose of the project is to systematize data on genetic mutations that contribute to the development of cancer. This project started in 2005 and has been designed for only 3 years. As of 2017, the TCGA project was finalized, but the investigated data has been made open. The TCGA includes samples from more than 11,000 patients for 33 types of cancer. It is by far the largest collection of cancer data [3, 5].

A systematic set of brain MRI images of only one type of cancer, glioma, was selected for the task [6, 7]. Glioma is a very common type of brain tumor: they are diagnosed in 60% of cases. Glioma malignancies are classified according to the World Health Organization classification. In this case, low-grade gliomas (or LGGs) are considered - these tumors usually have signs of benignity, but can sometimes rise in order, so they are classified as malignant. The set of data with such neoplasms is very relevant because low-grade gliomas are treatable, so the accurate detection and delineation of such a disease are of the utmost importance.

3 Machine Learning Methods for Brain MRI Neoplasm Segmentation

The artificial intelligence industry has been developing since the 1960s and has many different algorithms and approaches for solving computer vision. Quite often, developers have relied on elementary representations of human brain processes (for example, the process of transmitting nerve impulses has become the basis of the perceptron), combining them with approaches of statistics [11]. Consider some methods for solving segmentation problems.

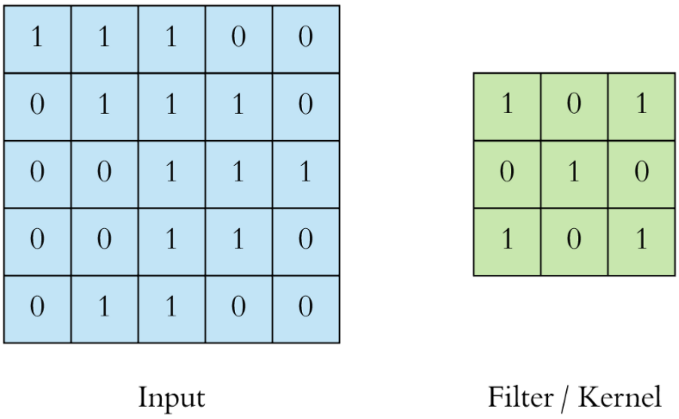
Deep learning is a field of artificial intelligence of machine learning based on a list of methods that seek to create high-level abstractions in visual data [12]. Research in this area is trying to make better representations and create models for learning these representations from large-scale data without professional markup [13-15].

Deep learning algorithms are based on certain principles. In this case, the observed data are generated by the interaction of certain factors that have formed at a deep level in the networks. Deep learning assumes that these levels of factors correspond to different levels of abstraction. Advances in deep learning have contributed to significant progress in optimizing optical image recognition tasks. Nowadays, architectures have become standard to solve computer vision problems, including segmentation. Usually, greedy optimization algorithms are quite common for deep learning problems [16].

Convolutional neural network (CNN) is a class of deep artificial neural networks that have been successfully applied to the analysis of visual images. CNNs use a variety of multilayer perceptrons designed to require minimal preprocessing [17, 18]. To explain the operation of this type of model, the main building blocks of CNN should be explained in detail.

Convolutional layer. The convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution simulates the response of an individual neuron to a visual stimulus [3].

Consider this example in Fig. 1. The Input is an array of input data, such as an image (in this case, a 5x5 pixel binary image). The convolution process itself is performed using a filter (in this image it is a Filter / Kernel array). Filter sizes and values are usually determined by an expert who designs the CNN model, and they are smaller than the input data because they are used as some sort of “sliding windows” [2].



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Fig. 1. An example of the input during the convolution process.

During the convolution process, the filter is "overlaid" on the input image. The example of such overlay is shown in Fig. 2. During the overlay, the corresponding values from both arrays are multiplied and the results are summed. For a cell with a value of 4 in the figure, the corresponding area on the input array is called its receptive field, because values in this area affect the values in the resulting cell [3, 4].

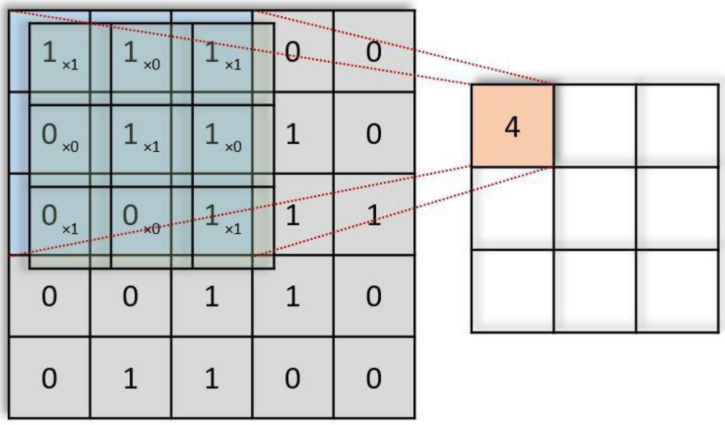
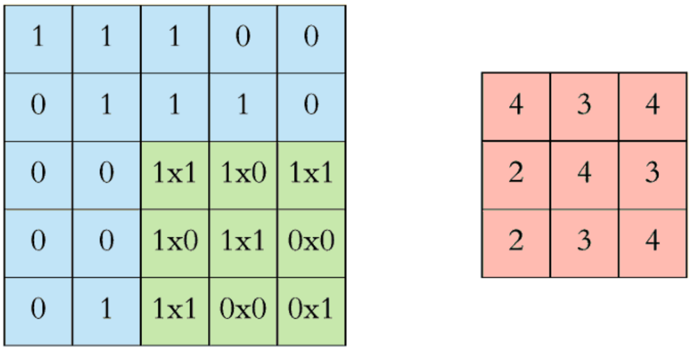


Fig. 2. The first iteration of the convolution process.

After that, the filter starts to move in the image with a certain step (in this example step - 1 pixel) and forms an output data layer whose dimension is smaller than the dimension of the input data. For a 5x5 image, a 3x3 filter, and a 1-pixel step, a 3x3 output array is obtained when minimized. In Fig. 3, the result is defined in red - the output array, which is an intermediate layer in the overall structure of the CNN [18].



1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

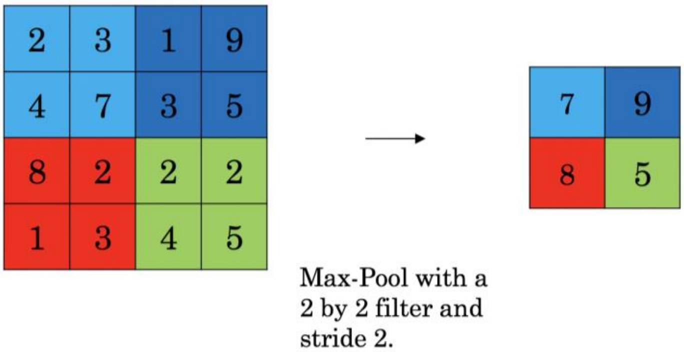
Fig. 3. The last iteration of the convolution process.

Pooling. After the convolution process, the resulting feature map undergoes a pooling process. During aggregation, data is "compacted" and shrinks in size, highlighting the most important features from a model perspective. Aggregation is also one way to prevent model retraining [3, 18].

During aggregation, the size and pitch of the filter are chosen, which is similar to the convolution filter, but when aggregating, the filter only separates the receptive field regions for the values of the resulting array [4].

There are different ways to aggregate a feature map. The most common is max pooling. An example of his work is shown in Fig. 4.

In this case, a filter of size 2x2 and step 2 was selected. The operation of max-pooling selects the maximum value of the receptive field for each of the cells in the output matrix. This approach preserves important information in the image and eliminates noise information when the model is processing large images [2-4, 18].



$$\begin{matrix} 2 & 3 & 1 & 9 \\ 4 & 7 & 3 & 5 \\ 8 & 2 & 2 & 2 \\ 1 & 3 & 4 & 5 \end{matrix} \longrightarrow \begin{matrix} 7 & 9 \\ 8 & 5 \end{matrix}$$

Max-Pool with a 2 by 2 filter and stride 2.

Fig. 4. Max-pooling operation.

Now let's take a look at some popular CNN architectures.

ResNet refers to one modification of convolutional neural networks [19]. When the deeper network begins to collapse, there is a problem: as the depth of the network increases, accuracy first increases and then decreases rapidly. Reduced learning precision shows that not all networks are easy to optimize. To overcome this problem, a “residual” learning structure was introduced. It uses shortcuts.

UNet is also one of the implementations of CNN [20, 21]. It was designed to segment biomedical images at the Faculty of Computer Science, University of Freiburg, Germany. The network is based on a fully convolutional network whose architecture has been modified and expanded to work with fewer images for training and to provide more accurate segmentation results. The basic idea is to complement the conventional network of sequential layers where pooling operations are reduced (dimension reduction) the objects are replaced by the operators of transposed convolutions or deconvolutions.

The architecture of the SegNet model was published in the journal by IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI 2017) [22]. This post has over 2600 references. This architecture is reminiscent of the U-Net form: the basic structures are the collapsing and expanding layers, they will process the input images in parallel. But, unlike U-Net, SegNet migrates aggregation layer indexes to maintain dimension instead of fully copying and pasting data to source layers. Thus, SegNet requires less computing power than U-Net.

4 Training and Selecting Brain MRI Segmentation Model

A certain technology stack was selected to solve this problem.

The low order glossary dataset was reviewed on Kaggle, a platform for machine learning and analytics competitions. Kaggle proposes to use Kaggle Kernels, a cloud-based technology that allows code to be run in a deployed Linux mini-environment, to process competition data. This is what Docker technology can do. Such notepads include the following technology points and make it possible to interact with the code in real-time. Such technologies are based on Jupyter systems.

Jupyter allows you to deploy multiple *Python* interactive viral environments and work with data analysis or neural network training. The *IPython* (Interactive Python) system simplifies the process of analyzing and visualizing data.

Other technologies are used for visualization and data processing. The *pandas* library is used for statistical data analysis - it makes it easy to analyze a large and documented array of data. *Matplotlib* and seaborn libraries are usually used to display graphs. Finally, the *scikit-image* library will be used for image processing.

A set of technologies to work with different machine learning methods should also be identified. For more classic algorithms, the *scikit-learn* library is used. However, a second technology stack was selected for deep learning. The popular *Pytorch* library has been selected for its work and has found its purpose in solving computer vision problems.

Modeling was done in the *Python* library using *Pytorch* building blocks. Because all other mechanisms and algorithms (such as data processing or result analysis) are unified, the program requires manual model selection, as all Pytorch library models inherit the *nn.Module* class.

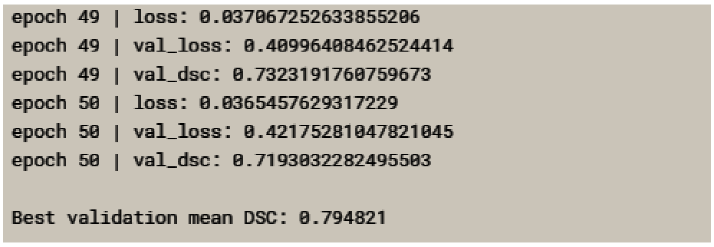
Most of the operation is performed at Kaggle. The main reason for this is the ability to use the GPU for free to train models in the cloud (providing 30 hours of GPU per week).

The following models use the same settings: one of the variations of the stochastic gradient descent *Adam* is used to write off.

This section the creation of U-Net, SegNet, and ResNet models and the selection of a shared accuracy metric. In this case, the Dice coefficient, or DSC, was used - the statistical coefficient used to compare two statistical samples, in this case, the sample is the pixels of the image assigned to the class. It allows you to compare the manual mask around the tumor with the created mask.

To create the ResNet architecture, certain classes have been developed that implement the elementary building blocks of a residual network: the classes *ResidualBlock*, *ResNetEncoder*, *ResNetDecode*, and *ResNet* itself. Also, according to research materials, several types of ResNet were created that differ only in settings: from *resnet18* to *resnet152*.

The ResNet model was then trained. This model architecture shows a good result in 79%. The ResNet training process is shown in Fig. 5.

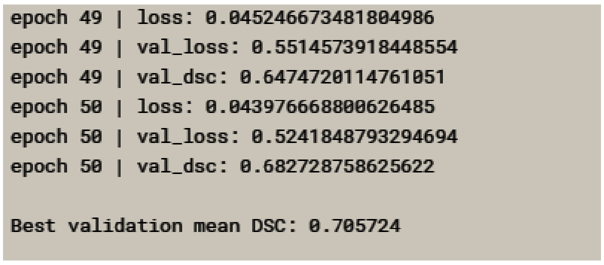


```
epoch 49 | loss: 0.037067252633855206
epoch 49 | val_loss: 0.40996408462524414
epoch 49 | val_dsc: 0.7323191760759673
epoch 50 | loss: 0.0365457629317229
epoch 50 | val_loss: 0.42175281047821045
epoch 50 | val_dsc: 0.7193032282495503

Best validation mean DSC: 0.794821
```

Fig. 5. The ResNet training process.

After this training, the SegNet model passed. This model architecture shows a result of 70.57%. At the moment, the ResNet model prevails over SegNet accuracy. The SegNet training process is shown in Fig. 6.

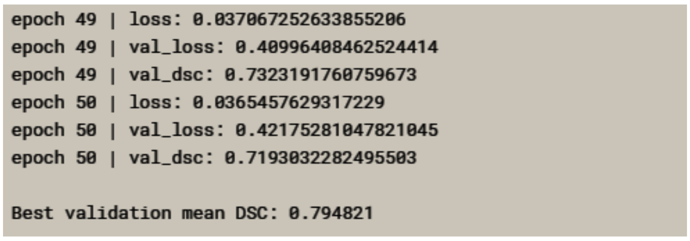


```
epoch 49 | loss: 0.045246673481804986
epoch 49 | val_loss: 0.5514573918448554
epoch 49 | val_dsc: 0.6474720114761051
epoch 50 | loss: 0.043976668800626485
epoch 50 | val_loss: 0.5241848793294694
epoch 50 | val_dsc: 0.682728758625622

Best validation mean DSC: 0.705724
```

Fig. 6. The SegNet training process.

Now we compare these models with U-Net [23]. A UNet class was created to follow the nn.Module class. The standard U-Net building block is a set of convolution, normalization, and activation. Such blocks are combined into a double structure, where data is first encoded and collapsed, and then decoded and expanded. The architecture of the U-Net model shows a result of 79%. The model shows better results than the previous two models. The training process of U-Net is shown in Fig. 7.



```
epoch 49 | loss: 0.037067252633855206
epoch 49 | val_loss: 0.40996408462524414
epoch 49 | val_dsc: 0.7323191760759673
epoch 50 | loss: 0.0365457629317229
epoch 50 | val_loss: 0.42175281047821045
epoch 50 | val_dsc: 0.7193032282495503

Best validation mean DSC: 0.794821
```

Fig. 7. The U-Net training process.

In the future, we will apply pre-processing methods to improve the accuracy of the models. To simplify the work and improve the results, we apply the statistical technique of "ceiling analysis". During operation, the model will consistently train without processing methods and then train on these methods, which will be added sequentially. This process is repeated for each architecture. This approach allows you to check the progress of model training and the impact of each part of the system on accuracy: taking into account changes in accuracy in different parts will allow us to highlight priority areas for further work. The results of such a ceiling analysis are shown in Table 1.

As you can see from the table, the best result is to use the U-Net model with additional data processing to unify the data, and then to increase the localization accuracy. The result of 94% is a significant improvement and satisfies the needs of the task, so the problem can be considered solved.

In Fig. 8 the predictions of the final optimized model on the validation data are indicated. Model predictions are indicated using red color, while ground truths (input segmentations from a radiologist) are colored green.

Table 1. Ceiling analysis of CNN models used for the segmentation of brain MRI scans on neoplasms.

	No processing	**Cropping**	**Resizing**	**Rotation**	**Thresholding**
ResNet	0.7948	0.8011	0.8114	0.7915	0.8674
SegNet	0.7057	0.7544	0.7915	0.8114	0.8456
U-Net	0.7948	0.8023	0.9012	0.9115	0.9498

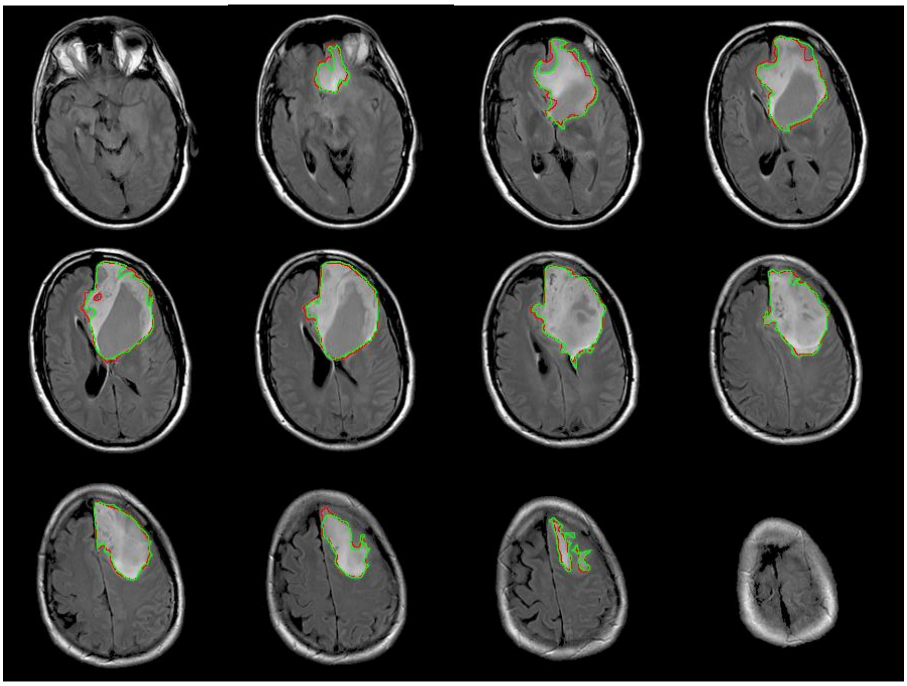


Fig. 8. Validation Predictions - U-Net (Final Version).

However, let us point out the advantages of such a representation approach. The table has a key place for the U-Net architecture: with the use of resizing techniques the accuracy of the U-Net model has increased by 10%. This shows that image resizing technology has significantly affected the accuracy of the model as a whole.

5 Conclusions

This paper focuses on the study of machine learning algorithms for the automated segmentation of neoplasms in brain MRI images.

The first section provides a general description of the field of computer vision, its application in biomedicine, and the importance of pre-diagnosis of neoplasms.

The second section analyzes the selected task and the selected dataset from The Cancer Genome Atlas project. The steps required for the study were also formulated, namely the study and selection of available libraries for machine learning, testing of various machine learning and depth learning algorithms, optimizing tumor segmentation while improving the performance of existing architectures.

In the third section, the selected machine learning algorithms were used and developed to solve the problem of segmentation of tumors on brain MRI images. After all, during the comparison, a system of the U-Net architecture with additional processing methods was selected as the final model. Its accuracy reached 94%, which is a significant result compared to manual image segmentation.

References

1. Mitchell, T.: Machine Learning. McGraw Hill. p. 2 (1997). ISBN 978-0-07-042807-2.
2. Zeng, Z., Wang, J.: Advances in Neural Network Research and Applications. SpringerVerlag, Berlin, Heidelberg (2010). DOI: 10.1007/978-3-642-12990-2.
3. Jordan, M.I., Mitchell, T.M.: Machine learning: Trends, perspectives, and prospects. Sci, NY (2015). DOI: 10.1126/science.aaa8415.
4. Yadav, N., Yadav, A., Kumar, M.: An Introduction to Neural Network Methods for Differential Equations. Springer, Netherlands (2015). DOI: 10.1007/978-94-017-9816-7.
5. Chouhan, S.S., Kaul, A., Singh, U.P.: Image Segmentation Using Computational Intelligence Techniques: Review. Arch. Computat. Methods Eng. 26, 533–596 (2019). DOI: 10.1007/s11831-018-9257-4.
6. Coupé, P. et al.: AssemblyNet: A Novel Deep Decision-Making Process for Whole Brain MRI Segmentation. In: Shen D. et al. (eds.) Medical Image Computing and Computer Assisted Intervention, MICCAI 2019, LNCS, vol. 11766, pp. 466–474. Springer, Cham (2019). DOI: 10.1007/978-3-030-32248-9_52.
7. Yu, R., Deng, M., Yap, P., Wei, Z., Wang, L., Shen, D.: Learning-Based 3T Brain MRI Segmentation with Guidance from 7T MRI Labeling. In: Wang, L., Adeli, E., Wang, Q., Shi, Y., Suk, HI. (eds.) Machine Learning in Medical Imaging, MLMI 2016. LNCS, vol. 10019, pp. 213–220. Springer, Cham (2016). DOI: 10.1007/978-3-319-47157-0_26.
8. Khortiuk, I., Kondratenko, G., Sidenko, I., Kondratenko, Y.: Scoring System Based on Neural Networks for Identification of Factors in Image Perception. In: 4th International

Conference on Computational Linguistics and Intelligent Systems (COLINS’2020), CEUR-WS, vol. 2604, pp. 993–1003 (2020). http://ceur-ws.org/Vol-2604/paper66.pdf.

9. Esteva, A., Kuprel, B., Novoa, R. et al.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118 (2017).
10. Gomolka, Z., Dudek-Dyduch, E., Kondratenko, Y.P.: From homogeneous network to neural nets with fractional derivative mechanism. In: Rutkowski, L. et al. (eds.) International Conference on Artificial Intelligence and Soft Computing, ICAISC-2017, Part I, LNAI, vol. 10245, pp. 52–63. Springer, Cham (2017).
11. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25. pp 1090–1098 (2012).
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521, 436–444 (2015). DOI: 10.1038/nature14539.
13. Glorot, X., Bordes, A., Bengio. Y.: Deep sparse rectifier neural networks. In: 14th International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011).
14. Kushneryk, P., Kondratenko, Y., Sidenko, I.: Intelligent dialogue system based on deep learning technology. In: 15th International Conference on ICT in Education, Research, and Industrial Applications: PhD Symposium (ICTERI 2019: PhD Symposium), CEUR-WS, vol. 2403, pp. 53–62 (2019). http://icteri.org/icteri-2020/PhDS/11110053.pdf.
15. Siriak, R., Skarga-Bandurova, I., Boltov, Y.: Deep Convolutional Network with Long Short-Term Memory Layers for Dynamic Gesture Recognition. In: 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 158–162 (2019). DOI: 10.1109/IDAACS.2019.8924381.
16. Bengio, Y., Lamblin, P., Popovici, D.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, pp. 153–160 (2006).
17. Lytvyn, V., Gozhyj, A., Kalinina, I., Vysotska V., Shatskykh V., Chyrun, L., Borzov, Y.: An intelligent system of the content relevance at the example of films according to user needs. In: CEUR Workshop Proceedings, vol. 2516, pp. 1–23 (2019).
18. Ketkar, N.: Convolutional Neural Networks. In: Deep Learning with Python, pp. 63–78. Apress, Berkeley, CA (2017). DOI: 10.1007/978-1-4842-2766-4_5.
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015).
20. Ronneberger, O., Fischer P., Brox T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015).
21. Kondratenko, Y., Gordienko, E.: Neural Networks for Adaptive Control System of Caterpillar Turn. In: Annals of DAAAM for 2011 & Proceeding of the 22th Int. DAAAM Symp. "Intelligent Manufacturing and Automation", pp. 0305–0306 (2011).
22. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation (2016).
23. Buda, M., Saha, A., Mazurowski, M.: Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm (2019).