# Application Of MLOps Practices For Biomedical Image Classification

Oleh Berezsky[a)], Oleh Pitsun[a)], Grygory Melnyk [a)], Yuriy Batko [a)], Bohdan Derysh [a)], Petro Liashchynskyi [a)]

[a] *West Ukrainian National University, 11 Lvivska st., Ternopil, 46001, Ukraine*

**Abstract**
With active hardware development, the number of software machine learning-based systems has increased dramatically in all areas of human activity, in particular, in medicine. The use of machine learning elements in software systems requires the organization of a pipeline process of software development, testing, and support. The application of MLOps technologies will improve the quality and speed of system development, as well as simplify the process of adjusting the algorithm parameters to improve the system operation quality. The purpose of this work is to develop an MLOps pipeline that will consider all the necessary stages of software development based on machine learning algorithms for biomedical image processing.

**Keywords**
Machine learning, MLOps, biomedical images, programming.

## 1. Introduction

Many software development companies in various fields have begun to actively implement machine learning techniques. A large number of funds will be allocated for these needs. According to Deeplearning.ai reports [1], only 22 percent of all projects using artificial intelligence have successfully implemented the process of using machine learning models. The standard software development process uses only programming languages, frameworks, and libraries. The process of developing software using elements of machine learning requires the development of neural network architectures, tools for processing large volumes of data, and training and testing system modules. The software development industry has faced a number of challenges that have led to the development of the DevOps model. This model provides a pipelined development process that allows optimizing the code development process. Leite et al. in [2] presented the concepts and peculiarities of DevOps technology. In the work [3], the authors provided tools and techniques that are widely used in DevOps-based software development.

The MLOps model is aimed to organize the machine learning process. MLOps uses DevOps practices for machine learning and allows programmers to work collaboratively on a single project. This allows for increasing the speed of development and provides rapid data analysis by means of using monitoring tools. Thus, the use of this approach allows implementing machine learning in modern projects on an industrial scale, and not only in a test form. The peculiarity of this publication is that we analyze all the steps necessary for the high-quality implementation of machine learning elements in the process of development and maintenance of specialized software based on image processing. The novelty of the work is that the necessary additional steps inherent only in the stage of processing biomedical images are taken into account.

The life cycle of machine learning-based software development consists of the following components:

- obtaining data (biomedical images);
- data processing, bringing it to the required form, for example, image filtering, image segmentation, etc.;

IDDM-2022: 5th International Conference on Informatics & Data-Driven Medicine, November 18–20, 2022, Lyon, France
EMAIL: ob@wunu.edu.ua (A. 1); o.pitsun@wunu.edu.ua (A. 2); mgm@wunu.edu.ua (A. 3); bum@wunu.edu.ua (A. 4); dbb@wunu.edu.ua (A. 5); p.liashchynskyi@st.wunu.edu.ua (A. 6);
ORCID: 0000-0001-9931-4154 (A. 1); 0000-0003-0280-8786 (A. 2); 0000-0003-0646-7448 (A. 3) ; 0000-0002-6732-4865 (A. 4) ; 0000-0002-7215-9032 (A. 5) ; 0000-0002-3920-6239 (A. 6)

- development of neural network architecture, for example, convolutional neural network;
- architecture tuning;
- deployment;
- monitoring of work results.

One of the key approaches for project code deployment is the use of continuous integration and continuous delivery.

MLOps develops the software development pipeline by providing a closer collaboration between data groups. This accelerates the speed of project release and the ability to adapt the input parameters of machine learning algorithms depending on the indicators of the monitoring results. MLOps is an extension of the concept of DevOps and is designed to run machine learning models in production. The purpose of this work is to develop an MLOps pipeline that considers all the necessary stages of software development based on machine learning algorithms for biomedical image processing.

## 2. Literature review

Analysis of recent publications of scientists in this field is presented in the literature review.

In work [4], the authors investigated the "MLOps" concept and highlighted the advantages of its application for software development.

Yue Zhou in [5] reviewed such platforms as TensorFlow Extended, ModelOps, and Kubeflow. As a result of the analysis, the author highlighted the systems' imperfections from the ML pipelines' point of view. The author analyzed the speed of each stage in ML pipelines.

Kreuzberger et al. in [6] conducted a generalized analysis of MLOps approaches and modern architectures. The authors analyzed publications, software tools, and expert feedback in this area.

In the book "Practical MLOps" [7], the authors provided examples of using MLOps solutions in combination with AWS, Microsoft Azure, and Google Cloud services. The authors also provided the best solutions for applying MLOps-based practices at the stage of system monitoring.

Application of MLOps-practices using AWS SageMaker, Google Cloud, and Microsoft Azure services is considered in work [8]. In addition, the authors presented the results of using the PyTorch, Keras, and TensorFlow libraries.

Reddy et al. in [9] proposed a framework for the machine learning process (MLOps) for platform development. This platform optimizes data and integrates processes, as well as brings together all processes by automating the project deployment phase.

Currently, there is a problem with harmonizing software development standards in medicine with elements of artificial intelligence. The authors in [10] provided arguments for the need to implement software development standards at the international level.

Kaminwar et al. in the work "Structured Verification of Machine Learning Models in Industrial Settings" [11] showed 5 stages of the life cycle of developing software applications based on machine learning.

The DevOps methodology appeared much earlier than the concept of MLOps and involved approaches to software development without the use of machine learning elements. In a research study [12], Erich et al. provided ways to use the DevOps methodology in software development in organizations that operate in various industries. In research [13], the authors focused their attention on automation, software development culture, continuous integration, and continuous delivery approaches.

Therefore, in these publications, scientists paid considerable attention to data processing in general, and in most cases in text format. The main goal of implementing DevOps practices is to eliminate the barrier between software developers and operations [14]. In work [15], the authors emphasized the application of DevOps practices at the level of cloud computing and testing. It made it possible to provide software and services quickly, reliably, and with better quality. DevOps uses a variety of methodologies that unite developers and operations personnel [16]. Applying DevOps practices of continuous automation for machine learning is described in [17]. In work [18], Ebert et al. analyzed modern tools for DevOps specialists.

The application of machine learning technologies for biomedical image analysis has its own peculiarities. The task of automatic biomedical image segmentation using the U-Net architecture is considered in [19]. The specific features of immunohistochemical image-based breast cancer diagnosing were demonstrated in [20]. An adaptive method of immunohistochemical image processing was developed in [21]. The classification of cytological images was considered in the article [22]. The process of entire biomedical image processing requires the development of a specialized approach that includes computer vision algorithms, machine learning, and other typical software components.

Currently, there are other similar tools and prototypes that cannot implement the necessary functionality. However, they have a number of disadvantages:

- poor documentation;
- the platforms are under development, so some functionality is not fully implemented; resource limitation in the free version;
- experience with Amazon services is required to get started.

Therefore, research and development of a pipeline for biomedical image processing is an urgent task.

## 3. Problem statement

Development of the MLOps methodology for designing a software system for biomedical image processing is an important task.

The objectives of this work are as follows:

1. Analyze MLOps platform tools.
2. Develop the main components of the pipeline for image analysis.
3. Describe the ML-pipeline for biomedical image processing.

## 4. Analysis of MLOps tools and platforms

MLOps provides an entire software development lifecycle, from an idea to the project deployment. Comparison of MLOps tools is a complex process, as there are a large number of evaluation criteria and specificity of a subject area. Table 1 provides a comparative analysis of MLOps tools and highlights their advantages and disadvantages.

**Table 1**
Comparative analysis of MLOps tools.

| MLOps - tool | Advantages | Disadvantages |
|---|---|---|
| Iguazio | Availability of a large number of ready-made features.<br>A convenient interface for implementing the model in real life.<br>Availability of a free trial period.<br>API availability. | Poor documentation |
| Kubeflow | Availability of pyTorch, Jupyter, TensorFlow, and scikit-learn.<br>Availability of integration with Kubernetes.<br>Ability to scale the architecture. | The need for knowledge and experience in containerization |
| Superannotate | Emphasis on image and video processing.<br>Conveniently organized conveyor.<br>Clear documentation, in particular, for forming a set of images. | The platform is under development, so some functionality is not fully implemented. |

| | | |
|---|---|---|
| | Ability to import/export annotations from third-party services, such as AWS.<br>Integrations with AWS, Azure, and GCP. | |
| Amazon SageMaker | Having Amazon SageMaker Pipelines.<br>Ability to apply CI/CD approach.<br>Presence of logging of training data processes, platform configurations. | Complex documentation<br>Experience with Amazon services is required to get started |
| Valohai | Open API.<br>Availability of means for A/B testing.<br>The availability of means to ensure the security of the Sign-On system (SSO), (2FA). | No free version. |
| H2O MLOps | Support for integration by leading cloud providers.<br>Real-time dash boarding.<br>Availability of A/B testing.<br>Convenient deployment environments. | Not detailed documentation |
| Neptune.ai | Model metadata logging.<br>Convenient visualization.<br>Data comparisons.<br>Integration with cloud systems. | Resource limitation in the free version. |

So, machine learning-based software systems are currently actively developing. Available services provide an opportunity to develop systems that use artificial intelligence. Most of the MLOps tools have a convenient graphical interface that allows for monitoring all stages of program development. Also, a key characteristic of such tools is a typical workflow and components for integration with cloud services.

## 5. MLOps workflow for biomedical image.

Unlike the DevOps concept, MLOps involves more experiments and tests. MLOps is a set of approaches for communication between data scientists, developers, and operation engineers.

MLOps workflow consists of 3 main components:

1. Build.
2. Deploy.
3. Monitor.

MLOps-workflow for processing biomedical images is shown in Figure 1.

The main difference between DevOps and MLOps is the availability of data. Data can be in structured or unstructured forms. After the formation of the data set, it is necessary to divide it into a test sample and a training sample. There are two main approaches to dividing the sample:

1. Creation of two directories "test" and "training".
2. Storage of all images in one directory and software division into test or training samples.

The developed directory structure for processing immunohistochemical images looks like this:

- ***Research_title***

- **Er** (estrogen)
- **Her2neu**
- **Pr** (progesterone)
- **Ki-67**
- **Histology**

In these directories, there are files of researched images in RGB format. Image labeling is also a component of the MLOps workflow.

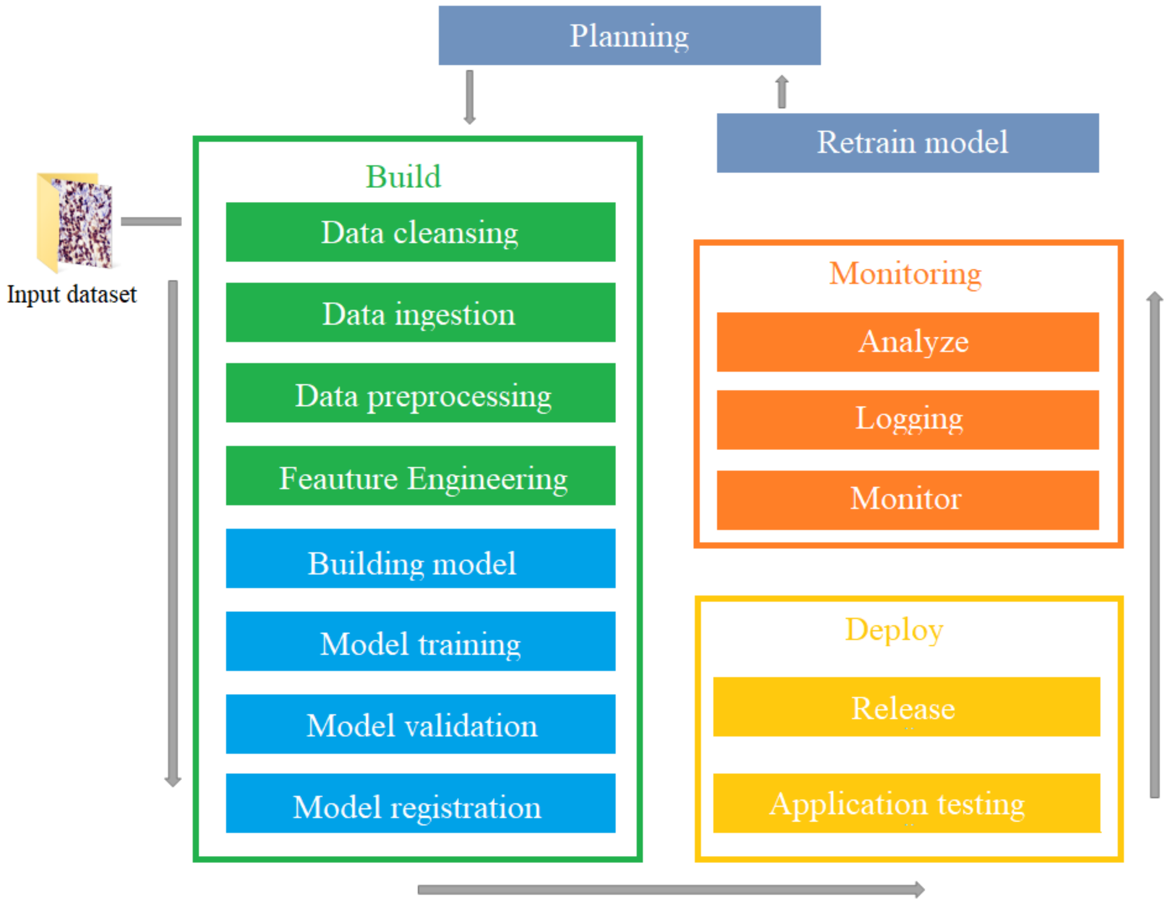


**Figure 1:** MLOps workflow for biomedical images.

In the first stage, input data is loaded and further processed. In most cases, the image database is created manually. Pre-processing takes place automatically based on computer vision algorithms. Algorithm parameters are formed on the basis of training results. An example of the preprocessing code is:

```
Mat newImageMat = new Mat();
this.normalSegmentedImgMat.copyTo(newImageMat);

/** IMAGE AFTER PREPROCESSING*/
ImageManagerModule imageManagerModule = new ImageManagerModule();
newImageMat =
imageManagerModule.autoImageCorrection(newImageMat,lowTreshValue.get(i));
```

Examples of immunohistochemical images are shown in Figure 2.

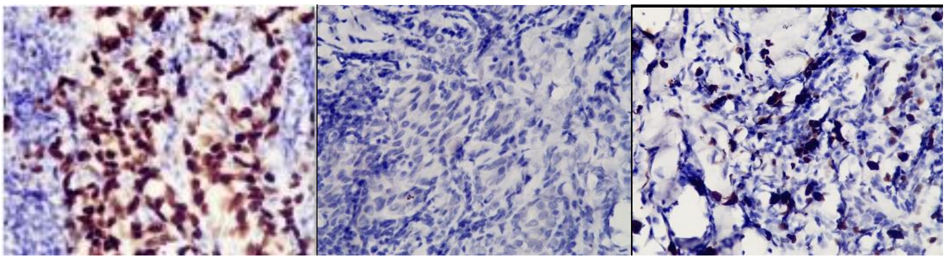

**Figure 2:** Examples of immunohistochemical images

After the data is prepared and processed, the build model is developed. A machine learning model is a file that consists of the results of training data based on certain algorithms. To build a model, besides data training, it is necessary to develop neural network architecture for classification. The developed CNN architecture is shown in Figure 3.

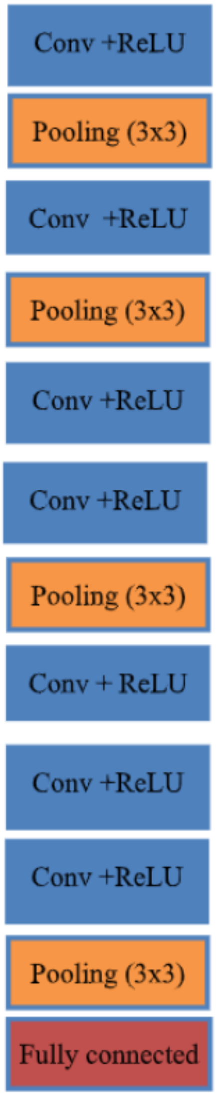


**Figure 3:** CNN Architecture

The MLOps pipeline for biomedical images processing is characterized by the fact that it is necessary to provide steps related to pre-processing of images, taking into account filtering elements, brightness/contrast level adjustment based on computer vision algorithms.

For image segmentation, the architecture of the U-net network was developed, which is shown in Figure 4:

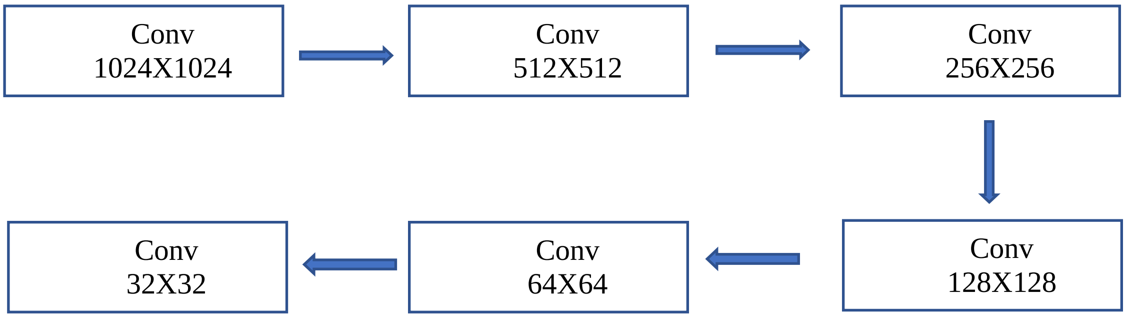


**Figure 4:** Architecture of the U-net encoder

After building and training the neural network model, the performance of the model is evaluated based on the test sample. If the results of the model are as expected, then the model is saved.

The "Model registration" stage involves containerization of the project together with the developed model using Docker. Docker is a software tool that combines operating system code and additional libraries. Containerization allows the creation of a configuration file that includes all the necessary dependencies for project execution.

The Deploy stage serves to deploy the project in the required environment using such tools as container instances, Kubernetes clusters, or a virtual machine. In this stage, testing is a key procedure. Successful execution of all automatic tests allows deploying and fully engaging the project in the required environment, for example, in the cloud.

After the implementation of the developed project and machine learning model, it is important to monitor the performance of the developed program. Therefore, monitoring stages are used in the MLOps pipeline. With the help of special tools, it is possible to monitor various work parameters, including system load. Logging in is also an important step. This stage helps to monitor the state of the system not only in real-time, but also during some specific period (e.g., week, day, or hours). Usually, log files are located in the server where the software system is running. There are also tools that allow more convenient record analysis in log files.

Specially trained engineers are engaged in the analysis of system indicators. Feedback from engineers on the system indicators allows for adjusting the developed project: to improve the architecture of the neural network, use more training data at the training stage, and increase the characteristics of the environment in which the project is deployed. If it is necessary to change any characteristics of the project, then data scientists, developers, or system administrators start working and the project deployment process takes place according to the previous workflow.

## 6. Conclusions

1. A comparative analysis of MLOps tools was carried out, which made it possible to highlight their advantages and disadvantages. In particular, most of the tools include API for working with the system and integrating with known cloud services.

2. There was developed MLOps workflow for biomedical images. This workflow takes into account the peculiarities of image processing.

3. Architectures of convolutional neural networks and U-net networks, which are components of the model code built during the workflow execution, were developed.

## 7. Related Works and Discussion

In future research, it is planned to improve the existing pipeline by adding functionality to use not only convolutional neural networks, but also other machine learning tools, such as logistic regression, SVM, etc. It is also planned to combine the developed pipeline with its own software for the analysis of biomedical images.

## 8. References


[1] Deeplearning.ai. The batch companies slipping on ai goals self training for better vision muppets and models china vs us only the best examples proliferating patents. (2019) URL: https://info.deeplearning.ai/the-batch-companies-slipping-on-ai-goals-self-training-for-better-vision-muppets-and-models-china-vs-us-only-the-best-examples-proliferating-patents

[2] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. 2019. A Survey of DevOps Concepts and Challenges. ACM Comput. Surv. 52, 6, Article 127 (November 2019), 35 pages. https://doi.org/10.1145/3359981

[3] L. Zhu, L. Bass and G. Champlin-Scharff, "DevOps and Its Practices," in IEEE Software, vol. 33, no. 3, pp. 32-34, May-June 2016, https://ieeexplore.ieee.org/abstract/document/7458765

[4] Alla, S., Adari, S.K. (2021). What Is MLOps?. In: Beginning MLOps with MLFlow. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6549-9_3

[5] Y. Zhou, Y. Yu and B. Ding, "Towards MLOps: A Case Study of ML Pipeline Platform," 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2020, pp. 494-500, https://ieeexplore.ieee.org/abstract/document/9361315

[6] D. Kreuzberger, N. Kühl, S. Hirschl "Machine Learning Operations (MLOps): Overview, Definition, and Architecture". Arxiv (2022) https://doi.org/10.48550/arxiv.2205.02302

[7] Noah Gift, Alfredo Deza (2021) Practical MLOps. O'Reilly Media, Inc.

[8] Sridhar Alla, Suman Kalyan Adari. (2021) Beginning MLOps with MLFlow. Apress Berkeley, CA. XIV, 330. https://doi.org/10.1007/978-1-4842-6549-9

[9] M. Reddy, B. Dattaprakash, S. Kammath, S. Kn. Application of MLOps in Prediction of Lifestyle Diseases. 2022 ECS Trans. 107 1191 https://doi.org/10.1149/10701.1191ecst

[10] Maximo J Marin, Xander M R Van Wijk, Thomas J S Durant, Machine Learning in Healthcare: Mapping a Path to Title 21, Clinical Chemistry, Volume 68, Issue 4, April 2022, Pages 609–610, https://doi.org/10.1093/clinchem/hvab285

[11] Sai Rahul Kaminwar, Jann Goschenhofer, Janek Thomas, Ingo Thon, and Bernd Bischl. (2021) «Structured Verification of Machine Learning Models in Industrial Settings». Big Data.ahead of print. http://doi.org/10.1089/big.2021.0112

[12] F. M. A. Erich. (2017) A qualitative study of DevOps usage in practice. Special Issue: Recent Advances in Agile Software Product Development – volume 26, Issue 9 https://doi.org/10.1002/smr.1885

[13] Alok Mishra, Ziadoon Otaiwi, (2020) DevOps and software quality: A systematic mapping. Computer Science Review. Volume 38 https://doi.org/10.1016/j.cosrev.2020.100308

[14] Welder Pinheiro Luz, Gustavo Pinto, Rodrigo Bonifácio, (2019) Adopting DevOps in the real world: A theory, a model, and a case study. Journal of Systems and Software. Volume 157 https://doi.org/10.1016/j.jss.2019.07.083

[15] Battina, Dhaya Sindhu, Devops, A New Approach To Cloud Development & Testing (2020). International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.7, Issue 8, page no.982-985, Available :http://www.jetir.org/papers/JETIR2008432.pdf

[16] Battina, Dhaya Sindhu, The Challenges and Mitigation Strategies of Using DevOps during Software Development (2021). International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.9, Issue 1, pp.4760-4765, January 2021, Available at :http://www.ijcrt.org/papers/IJCRT2101583.pdf

[17] Karamitsos, Ioannis, Saeed Albarhami, and Charalampos Apostolopoulos. 2020. "Applying DevOps Practices of Continuous Automation for Machine Learning" Information 11, no. 7: 363. https://doi.org/10.3390/info11070363

[18] Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. DevOps. IEEE Software 33, 3 (2016), 94--100; https://ieeexplore.ieee.org/document/7458761

[19] O. Berezsky, O. Pitsun, B.Derysh, T.Datsko, K. Berezka, N. Savka Automatic segmentation of immunohistochemical images based on U-NET architectures. IDDM-2021: 4th International Conference on Informatics & Data-Driven Medicine, November 19–21, 2021 Valencia, Spain. pp. 22-33, http://ceur-ws.org/Vol-3038/paper3.pdf
[20] O. Berezsky, O.Pitsun, T.Datsko, B.Derysh, I.Tsmots, V. Tesluk Specified diagnosis of breast cancer on the basis of immunogistochemical images analysis, IDDM’2020: 3rd International Conference on Informatics & Data-Driven Medicine, November 19–21, 2020, Växjö, Sweden. pp. 129-135. http://ceur-ws.org/Vol-2753/short5.pdf
[21] O. Berezsky, O. Pitsun, B. Derish, K. Berezska, G. Melnyk and Y. Batko, Adaptive Immunohistochemical Image Pre-processing Method, 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 2020, pp. 820-823, doi: 10.1109/ACIT49673.2020.9208920.
[22] O. Berezsky, S. Verbovyy, O.Pitsun, Hybrid Intelligent information techology for biomedical image processing. Proceedings of the IEEE International Conference “Computer Science and Information Technologies” CSIT’2018. Lviv: Ukraine. 11-14 September, 2018. pp. 420–423. https://doi.org/10.1109/STC-CSIT.2018.8526711.