

Министерство просвещения Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования «Московский педагогический государственный университет»

Институт физики, технологии и информационных систем

Кафедра теоретической физики им. Э.В. Шпольского

Кузнецов Никита Викторович

**Методика аттестации трековых мембран с использованием
компьютерного зрения**

03.03.02. Физика

Фундаментальная физика (на английском языке)

Выпускная квалификационная работа

(бакалаврская работа)

Заведующий кафедрой теоретической
физики им. Э.В. Шпольского
Д.ф-м.н., профессор
Наумов А.В.

Научный руководитель
Доцент, к.ф-м.н.
Бедин С.А.

Проверка на объем заимствований:
_____ % авторского текста

Москва, 2021 год

Аннотация

В данной работе представлена реализация подхода аттестации трековых мембран с помощью изображений, полученных на электронном микроскопе. Предложенный метод состоит из двух этапов: использование модели машинного обучения (свёрточная нейронная сеть) для получения бинарных масок изображений, на которых белые области отвечают за поры мембраны, а черные области – за фон. Последующая обработка полученных масок с помощью классических алгоритмов компьютерного зрения, реализованных в библиотеке OpenCV, позволяет рассчитать желаемые характеристики поверхности трековой мембраны. Примером таких характеристик служат пористость, плотность пор или их количество. Использование бинарных масок, полученных с помощью сегментации изображения, позволяет существенно упростить алгоритмический подход к подсчёту данных характеристик.

В работе также произведён сравнительный анализ качества масок, восстанавливаемых моделью, в зависимости от количества эпох обучения, функции потерь и непосредственно архитектуры модели. По результатам этого сравнения была выбрана лучшая конфигурация для дальнейшего использования. Маски, полученные именно с использованием лучшей конфигурации, были в дальнейшем использованы для расчёта характеристик мембраны. Для лучшей конфигурации также были получены кривые обучения и метрики в зависимости от эпохи. Было произведено сравнение ручного подсчёта характеристик трековой мембраны с подходом, в котором используется обученная модель. Из этого сравнения были выявлены ограничения использования предложенного метода.

Ministry of Education of the Russian Federation
Federal State Budgetary Educational Institution of Higher Education
«Moscow Pedagogical State University»

Institute of Physics, Technology and Informational Systems

Department of theoretical physics

Kuznetsov Nikita Viktorovich

Evaluation method of track membranes' images using computer vision

03.03.02 Physics

Fundamental Physics (in English)

Graduate qualifying work

(Bachelor thesis)

Head of department of theoretical physics
D.Sc in Physics, professor
Naumov A.V.

Supervisor
Associate Professor, PhD in Physics,
Bedin S.A.

Verification on the level of borrowing:
_____ % of author's text.

Moscow, 2021 year

Abstract

This work presents the implementation of the track membrane evaluation approach using images made by electron-beam microscopy. The proposed method consists of two stages: using a machine learning model (convolutional neural network) to obtain binary image masks, in which white areas stand for the pores of the membrane, and black areas stand for the background. Subsequent processing of the obtained masks using classical computer vision algorithms implemented in the OpenCV library allows calculating the desired characteristics of the track membrane surface. Examples of such characteristics are porosity, pore density, or several pores. The use of binary masks obtained by image segmentation makes it possible to significantly simplify the algorithmic approach to calculating these characteristics.

The work also provides a comparative analysis of the masks' quality restored by the model, depending on the number of training epochs, the loss function, and the model architecture itself. Based on the results of this comparison, the best configuration was selected for further use. The masks obtained using the best configuration were later used to calculate the characteristics of the membrane. For the best configuration, learning curves and metrics values depending on the epoch was also obtained. The manual calculation of the track membrane characteristics is compared with the approach that uses a trained model. From this comparison, the limitations of using the proposed method were revealed.

Content

Introduction	6
Structure and main goals	7
Acknowledgements	8
Literature review	9
Track membranes	9
History	9
Formation of an etched track	10
Track etching geometry	11
Industrial technology of track membranes	12
Application of track membranes	13
Machine learning	14
Theoretical background	15
Loss functions	18
Metrics	18
Convolutional neural networks	20
Application of machine learning	22
Experimental part	23
Training model	23
Loss function and metrics	23
Comparison of configurations	25
Training and metrics curves	27
Classical computer vision approach	31
Porosity calculation	31
Pore localization	31
Pore calculation	32
Results and discussion	33
Image segmentation	33
Mask analysis	34
Characteristics calculations	36
Conclusion	38
References	39

Introduction

Track membranes play an important role in both technological and scientific processes. They have found applications as highly efficient filters for water purification [1], blood plasmapheresis [2], and other practical applications. In scientific work, they are mainly used as a template for the synthesis of nanomaterials [3, 4], namely ensembles of metal nanowires.

The key point is the ability to calculate the characteristics of the track membrane since its parameters affect the quality of the result obtained during operation. For example, the pore distribution on the surface of the track membrane is important, as it directly affects the strength of the membrane and its performance characteristics [5]. Speaking about the production of nanomaterials, in this case, it is necessary to know exactly the characteristics of the membranes used (pore density, diameter, porosity), which largely determine the properties of the resulting nanomaterials. And if the values of the pore diameters can be found out when etching tracks with an error of a couple of percents, then the remaining characteristics must be calculated.

Also, knowing about the characteristics of the template on which a certain nanomaterial was made, it is possible to produce high-quality mathematical modeling of the physical processes occurring. Speaking about nanowires, it is important to take into account their relative position, which is of great importance when using ensembles of such silver nanowires as active substrates for surface-enhanced Raman scattering [6], because in this case, the different relative positions of the nanowires can affect the mechanisms of amplification of the Raman signal. It is equally important to know the exact pore density to calculate such values as local fields and signal amplification coefficients from biological and chemical substances in nanoconcentrations.

As a solution to the problem of obtaining the characteristics of track membranes from images taken with an electron microscope, and automating this process, it is proposed to use machine learning. The main motivation for using machine learning is the lack of building a deterministic algorithm that will process

each image. An important factor is that the trained model extracts the features of the necessary objects, so it is relatively invariant to the distribution of colors and noise of images. Another equally important factor is the lack of manual configuration, which allows you to get masks of more images in less time, which allows you to automate the process.

Structure and main goals

This work is devoted to the implementation of the method of certification of track membranes using machine learning methods and classical computer vision algorithms. Structurally, the work consists of:

1. Overview of track membrane technology;
2. Overview of machine learning technology;
3. Description of model training and used functions, metrics;
4. Discussion and comparison of the results obtained.

The main goals of the work are:

1. Implementation of the proposed method, as well as a demonstration of the quality of the algorithm in comparison with the currently existing manual approach;
2. Performing the calculation of the characteristics of track membranes by the proposed method.

Acknowledgments

I would like to express my deepest gratitude to my scientific supervisor – Bedin Sergey Alexandrovich – for big support with writing my diploma thesis and for providing data for model training. Also, I would like to sincerely thank Kozhina Elizaveta for supporting me during my work and for her great ideas. Thank you!

Literature review

Track membranes

History

In 1962 P. B. Price and R. M. Walker received a patent for “molecular sieve and methods of its preparation”. They studied defects that were left in thin films of mica by fragments of natural uranium. After that, they came up with idea that these defects could be increased by chemical etching. A method for producing membranes with homogeneous pores in size and shape was developed quite quickly [7, 8].

But they were not pioneering in discovering this effect. David Young suggested that the method of manifestation of dislocations in alkali-halide crystals can also be extended to structural defects arising in them during the passage of a heavy charged particle. This defect is called tracks [9].

The etching method has been used for a long time in mineralogy to determine the structure of a sample. Track etching was first observed in 1894. Then a guide to the crystallographic study of minerals was published, which included micrographs of apatite preparations, on which etched tracks of fission fragments were identified a century later [10]. Of course, in those years it was impossible to give an adequate interpretation of the observed etching patterns because at that time not only the phenomenon of spontaneous fission of heavy nuclei was not known, but the very concept of an atomic nucleus did not exist.

The next significant step towards the creation of membrane technology was made by G. N. Flerov. Starting from 1960, experiments on the synthesis of new elements using beams of accelerated heavy ions have been carried out at the Dubna Laboratory of Nuclear Reactions. The method of etching tracks of fission fragments of heavy nuclei to register the latter was quickly mastered and successfully applied by the laboratory specialists. The natural continuation of these works was the “nuclear filters” [11, 12].

The idea was based on the following factors. Firstly, the accelerated ion beams are characterized by high intensity, which makes it possible to increase the efficiency

of the irradiation process at the first stage of the technological process. Secondly, the nuclei of accelerated ions, unlike fission residues, are not radioactive, which makes the technology safer. Thirdly, the accelerated ions can have higher energy and, consequently, a greater range in the material, which allows you to vary the thickness of the membranes over a wider range. The result was the establishment of the production of track membranes using heavy ions.

Formation of an etched track

The formation of an etched track in the dielectric is a complex process. The first stage of the process is a physical part: while passing through medium heavy-ion interact with medium's electrons and give them part of its kinetic energy. Because of that part of atoms become excited and another part gets ionized losing electrons. The core of a track is such a zone where the concentration of excited and ionized atoms is high, and the diameter of the core is usually about 10 nm. Electrons that left atoms stay in the surrounding area of the core. Delta-electron have the largest energy and form radiational defects in track shell. Processes that are responsible for structure destruction of material in the core are coulomb explosion and thermal peak [13]. The result in track formation is that the medium in the core of the track is dramatically different from the initial medium in physical and chemical properties. For example, in polymers, many covalent bands would be destroyed (main chain and side groups) and the density of polymer will be decreased by 10% in the core [4]. By analyzing experimental data, it is seen that the main reason under track formation is an increase of free volume [14]. The free volume which is formed in a narrow cylinder along the trajectory of ion makes it an easier way for chemical reagent and speeds up the chemical destruction of macromolecules. In the case of polymers, there is a possibility of new bond formation. This factor is used to find a more sensible place for track etching.

Speaking about track shell there is no such dramatic change of medium as it was described for the core. For polymers, there is a special feature that stands for the speed of etching. The shell of the track is etched much slower than non-damaged

material [15]. This effect can be explained by crosslinking of macromolecules at a distance of about 10-50 nm. Etching speed itself depends on the degree of material destruction in the core of the track, which depends on the energy of the irradiated particles. The most convenient energy of such particles for track formation is called Bragg's peak [13].

Track etching geometry

In the simplest case process of track-pore transition is described by a geometrical model which includes only two parameters – speed of track etching (V_T) and speed of non-damaged material etching (V_B) [13]. This case is presented in Fig. 1a [16].

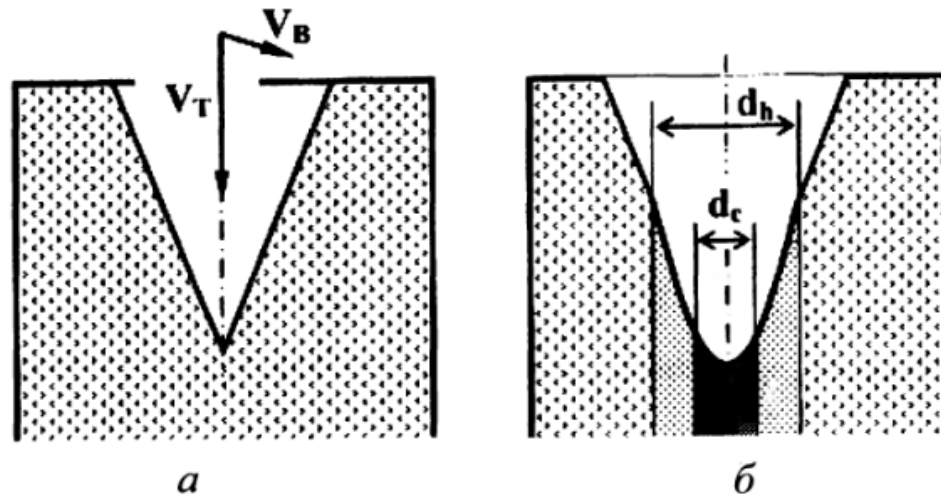


Fig. 1. The geometry of track etching. a - isotropic homogeneous medium, b – nanometer scale.

Neglecting the small size of radiational damage in the surrounding area of ion trajectory, it is taken that form of etching is a cone. In case when the speed of track etching is much larger than the speed of non-damaged material etching the angle at the top of the cone is small.

Another key quantity can be introduced - selectivity of etching. It is namely ratio of the speed of track etching and speed of non-damaged material etching. This parameter is important in the production and technology of track membranes. The form of the track after etching is characterized by selectivity (Fig. 1). For cylinder

pore, production selectivity must equal 10^3 . In nanoscale sizes form of the edge is dependent on the sizes of track core (d_c) and track shell (d_h) [17].

Industrial technology of track membranes

The SEM image of the track membrane is presented in Fig. 2 [18].

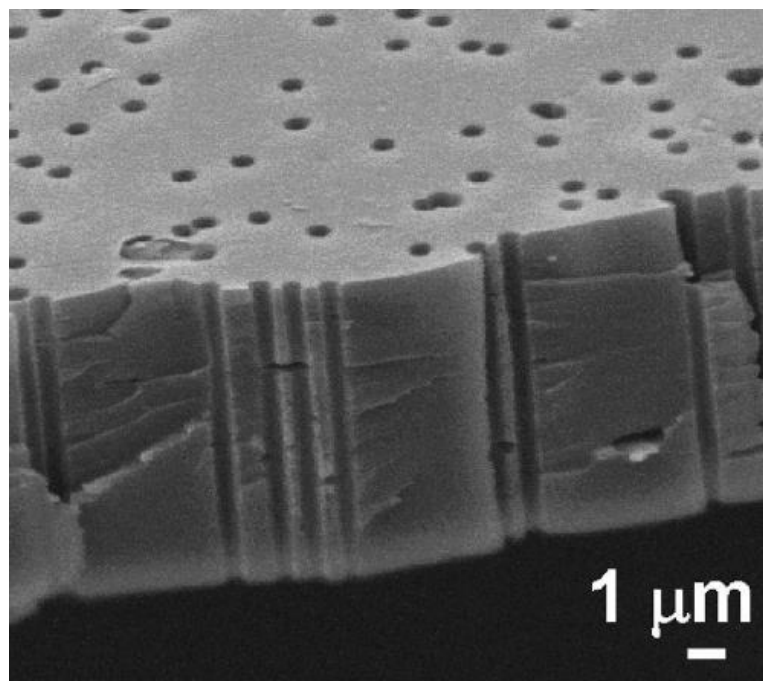


Fig. 2. SEM image of track membrane

The technology of track membrane production was developed in the Laboratory of Nuclear Reactions in Dubna by G. N. Flerov [11]. The whole process can be depicted in Fig. 3 [16].

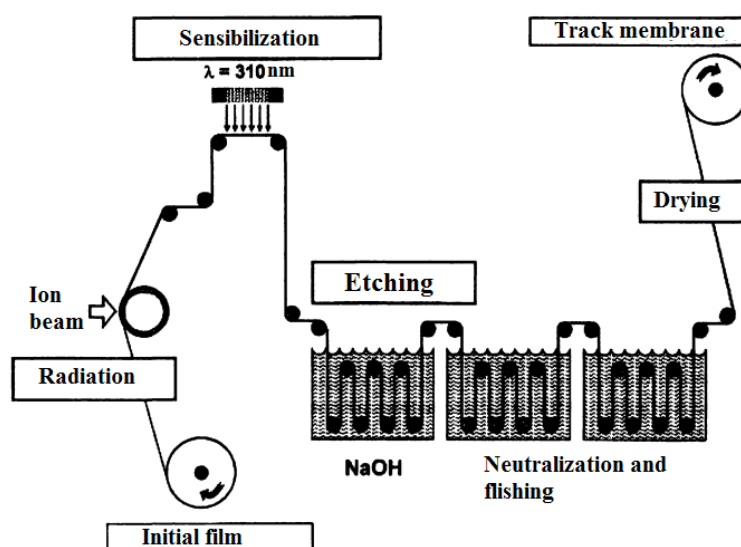


Fig. 3. Scheme of track membrane production

Polymer film goes through three main stages: ion beam radiation, ultraviolet radiation processing, and chemical etching. Firstly, the roll-shaped film goes to the radiation stage. Ion beam bombards the film in a vacuum chamber. The speed of the film transportation is fixed and can be changed to achieve different pore densities. For industrial irradiation of membranes U-400 and IC-100 cyclotrons are used. During radiation, polymer film goes through the cylindrical shaft which helps to achieve axis of tracks distribution in a specific interval of angles [19]. With the varying radius of the shaft, it is possible to fix the width of angle distribution in the membrane. By using this method it is possible to decrease the number of multiple overlapping pores [20]. The key role in track membrane production play the stability of ion beam in time.

The next step in the technological process is sensibilization by ultraviolet radiation. The purpose of the process is to increase the selectivity track etching and increase the uniformity of the membrane properties over the width and length.

At the stage of chemical etching formation of the track structure is started. The etching process is much faster than previous processes. Control of parameters of pore structure is done by periodically taking a sample and providing expert evaluation. The final control stage is realized with scanning electron microscopy.

Application of track membranes

Due to the transparency of the membranes, their small thickness, and smooth surface, they are an ideal material for the collection and analysis of micro-contaminants, colloidal particles, tissue cell preparations, etc. Because of these properties, track membranes are used for ecological research in case of water analysis and atmosphere, analysis with the use of microcopy, cell research, DNA research [13, 21, 22], and other researches as analysis of microparticles on deformability of red blood cells [13, 23]. The ability to accurately set the pore size and porosity provided track membranes with a decisive advantage when used for cell cultures. For example, in oncology, membranes are used for the Pap test. With the advent of track membranes, they were immediately used to model the processes of capillary-porous bodies [13, 14].

Machine learning

Looking up in the Oxford dictionary, one can define machine learning as a kind of artificial intelligence, in which computers, using large amounts of data, learn how to perform a certain task, rather than being programmed to perform it.

But besides machine learning, there are other terms that at the first sign can mean the same thing but getting deeper into the subject allows one to see the difference. These terms are artificial intelligence, deep learning, and neural networks. The difference can be depicted as shown in Fig. 4.

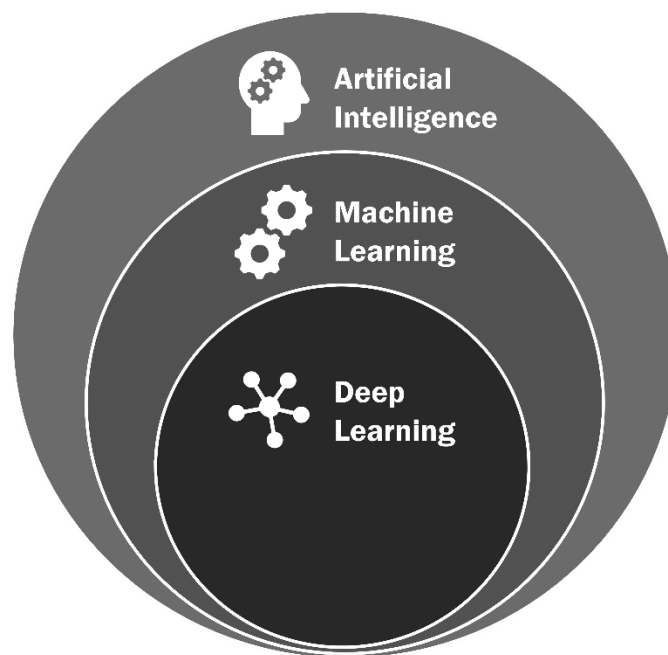


Fig. 4. Term diagram

In the classical case, programs have a completely deterministic and pre-thought-out algorithm, which someone has carefully developed and implemented in one of the many programming languages. Using machine learning techniques, one can avoid this part of the work. The trained model will try to independently extract the necessary features and build a certain logic to learn how to get the correct result on the input data supplied to it. Thus, it is possible to facilitate the solution of many problems in which it is difficult to build a certain algorithm of actions due to the complexity of the data, or because of the complexity of the implementation object.

Theoretical background

One of the first important steps towards a formal description of machine learning was taken in 1957 by Frank Rosenblatt [24]. He developed the early ideas of Warren McCulloch and Walter Pitt by proposing a scheme for a device that modeled the human process of perception and called this device a perceptron [25].

The described perceptron became one of the first models of neural networks that were widely used in machine learning. The mechanism has a direct analogy with a neuron in the human brain and has the following stages of operation: the signals coming from the sensors are transmitted to the associative elements, and then to the reacting elements. The key point in this approach is the training of the resulting neural networks, for which an iterative learning method was developed, inspired by how the learning process in humans occurs. This approach has been called the error correction method. After that, attempts were made to train the perceptron to classify letter images, as well as to test their generalizing abilities. In his work, Rosenblatt describes various types of perceptron: single-layer (Fig. 2), as well as multilayer perceptron with cross-and feedback connections.

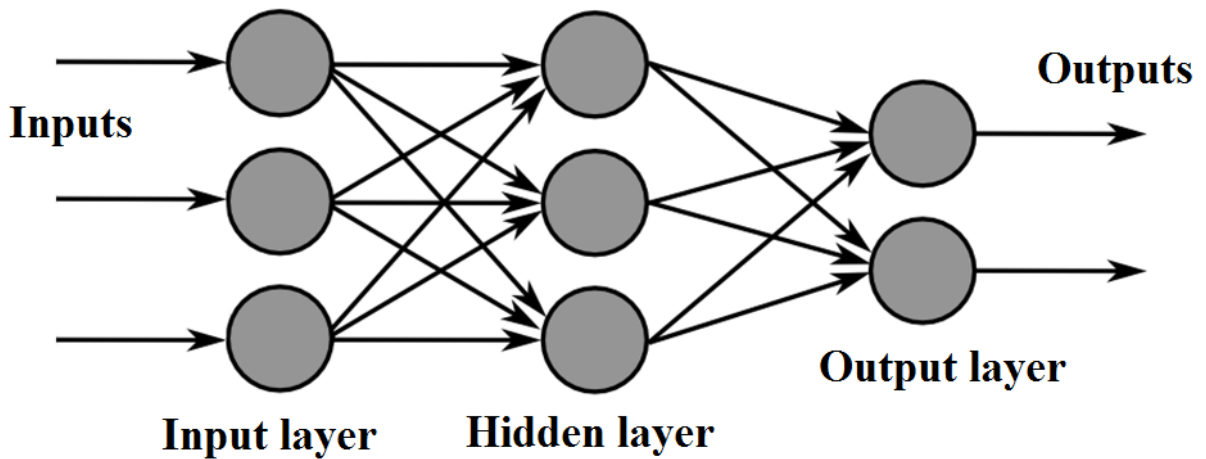


Fig. 5. Single-layer perceptron

In the context of this work, an important theorem on the convergence of the perceptron was also proved, which is that an elementary perceptron, which is trained by the error correction method, regardless of the initial state of the weight coefficients and the sequence of the appearance of stimuli, will always lead to the

achievement of a solution in a finite time. Rosenblatt himself believed that perceptron is primarily a step towards the research and use of artificial neural networks, and they do not represent the final version of a thinking machine. The main disadvantages of this approach are that perceptron has limitations in working with an invariant representation of images, as well as the fact that they do not have a significant advantage over analytical methods for solving problems and forecasting.

But progress did not stop at the approaches to the implementation of artificial neural networks proposed in the middle of the last century. Significant improvements have been made with both the theoretical and the hardware sides of the question. For example, a neural network of the perceptron is linear and cannot be a universal function approximator. To eliminate this drawback, you can add a nonlinear function at the output of the neuron, which will be called the activation function.

In [26] it was shown that in the case of a nonlinear activation function, a two-level neural network will universally approximate the function. In addition to non-linearity, such functions are also subject to other constraints, such as continuous differentiability, which allows the use of gradient descent methods for optimization, and monotonicity for convex optimization.

Tasks of machine learning can be divided into three categories depending on input data to the learning system: supervised, unsupervised, and reinforcement learnings.

In the case of supervised learning mathematical model is built in a such way that data contains inputs and corresponding outputs [27]. The given data is a training dataset using which model must set the correct way for solving the task. The training data is usually represented by a matrix and desired data – features – are vectors. Using algorithms of iterative optimization (as gradient descent) supervised model learns a function that can predict output associated with the input data. An optimal function will help the model to determine features for values that were not present in the training dataset [28]. An algorithm that improves the accuracy of its outputs

or predictions over time is said to have learned to perform that task [29]. The most popular supervised learning algorithms solve classification and regression problems. Classification is used in case the outputs are a given set of possible values and regression is used when the outputs are represented as continuous values within some numerical range [30].

Unsupervised learning differs from supervised in the sense that in this case model has only inputs with no corresponding outputs. In this situation, learning takes place with data that has not been labeled before learning. The main goal of unsupervised algorithms is to find similarities in data and give an answer based on the presence or absence of those similarities in data. Such an approach is used in finding the probability density functions [31] and cluster analysis of the data.

Just in between these two approaches stands semi-supervised learning. In this case, dataset some of the examples are missed. Closely related to this type of learning another branch of machine learning – weakly supervised learning. In this case, training labels are noisy and limited, but there are some advantages: these labels easier to obtain and this enlarge effectively the training sets [32].

The last type of learning approach is reinforcement learning. This branch of machine learning is concentrated on the actions of intelligent agents to maximize profit from the actions. This approach differs from the previously discussed type in a manner that there is no labeled data and no sub-optimal actions needed to be correct. The focus is on finding a balance between exploitation of uncharted territory and exploitation of current knowledge [33].

Despite all the existing approaches to learning, machine learning does not always allow you to get the desired result and there are several reasons for that: lack of data, incapability to access data, data bias, lack of resources, and evaluation problems. Machine learning approaches have not been developed well enough to reduce the workload burden without limiting necessary sensitivity [34].

Loss functions

Machine learning is tightly connected with mathematical optimization problems because learning problems are usually formulated as minimization of some loss function on a training dataset. The loss function itself represents the inconsistency between predicted value and value which was used to train a model. But the difference between machine learning and optimization appears when generalization arises [35]. Optimization tends to minimize the loss function on the training dataset, but the machine learning purpose is minimizing the function on new data.

Since neural networks can approximate functions, it is necessary to formalize this process. For this purpose, a loss function is defined, which varies depending on the machine learning task. In the future, it is by optimizing this function that it is possible to obtain a high-quality approximation using a neural network.

The most common optimization method in machine learning is gradient descent or stochastic gradient descent. This method of finding the local optimum of the function is based on the movement along the gradient. Since when working with neural networks, the main interest is the minima of the function, the movement occurs in the direction of the antigradient. This algorithm is the simplest to implement of all optimization algorithms, but it has weak convergence conditions. In the form of a formula, the method can be expressed as formula (1):

$$x_{j+1} = x_j - \lambda \nabla F(x_j) \quad (1)$$

λ in this formula is the rate of gradient descent, which can be either constant or decreasing in the process of gradient descent.

Metrics

Like any other tasks, machine learning solutions must be rated somehow to know the level of model performance. Evaluation of the model is an essential part of any project. Analyzing different metrics, it is possible to find weak places of the model and improve the task solution.

For different types of machine learning problems, there are different types of metrics. For example, for classification usually used such metrics as:

- Accuracy – the number of correct predictions divided by the total number of predictions;
- Logarithmic loss can be represented by formula (2):

$$-\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \cdot \log(p_{ij}) \quad (2)$$

where y is whether sample i belong to class j or not, and p is the probability of such an event. And it is truly not all possible examples of loss functions.

Often in machine learning concept of a confusion matrix is used (Fig. 6). It is also called an error matrix [36].

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Fig. 6. Confusion matrix

Using this specific table, it is possible to visualize the performance of a machine learning model. This approach is usually used in supervised learning. Rows of the matrix represent examples of actual class, and columns represent predicted class. But it also could be vice versa [37].

All tasks have specific metrics. Computer vision metrics [38], metrics for unsupervised learning [39], and even specific metrics for bank and retail branches [40]. From these leads, that researcher needs to find proper metrics from a variety of

existing ones to provide a correct evaluation of the model. Choice of correct metrics is important because it can mislead and show not valuable results.

Convolutional neural networks

In the case of the tasks closely related to visual imagery analyze there is a special class of machine learning architectures which is called convolution neural networks. These networks are based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps [41, 42]. Besides image processing, convolutional networks found applications in recommender systems [43], natural language processing [44], and time series analysis [45].

Convolution networks are closely connected to multilayer perceptrons which are fully connected. That means neuron from one layer is connected to the all neurons from the next layer. Convolution networks are usually called regularized versions of perceptrons because such regularization approaches are used as weight decay, skipped connections, and dropout. There exist some researches that show the biological analogy of this type of neural network [46-48].

Any convolutional network is formed by an input layer, hidden layers, and an output layer which is a full analogy to feed-forward networks. The hidden layer of the convolutional network consists of layers that perform convolutions, which is typically a dot product of convolution kernel with input matrix, and after that ReLU activation function follows. Moving along the matrix, convolution operation builds a feature map, which goes to the next layer of the network. After passing through the convolutional layer, a feature map of the image is formed which is also can be called an activation map. Convolutional neural networks also consist of pooling layers that reduce the dimensions of the data. There are global and local pooling which differ in the sense that local pooling small clusters and global acts on all neurons of the feature map [49, 50]. There are also different types of pooling such as maximum pooling [51, 52] – uses a maximum value of a local cluster to reduce dimension – and average pooling – uses the average value of a local cluster.

Pioneering work in the development of convolution neural networks was done by LeCun in 1998 when he developed LeNet-5 architecture [53]. Its purpose was to classify the hand-written digits on images of size 32x32 pixels. The scheme of architecture is presented in Fig. 7 [54].

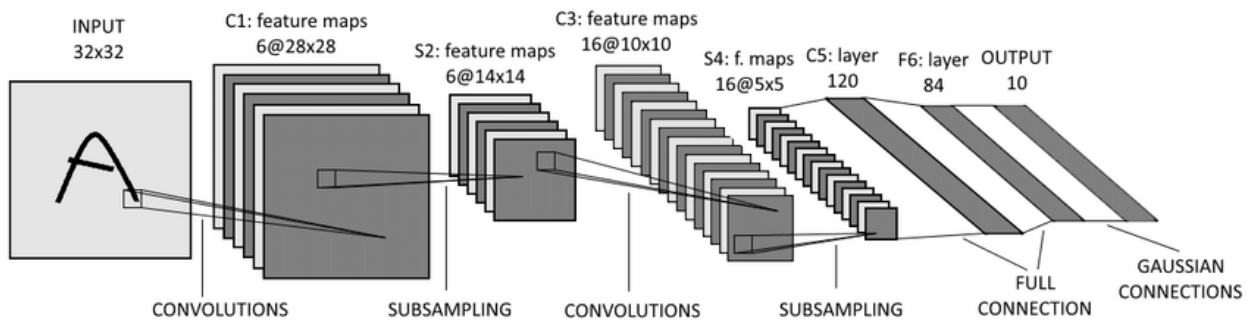


Fig. 7. LeNet-5

Even though the usage approach of convolutional neural networks was established in the 1980s, their breakthrough years started in the 2000s with the implementation of training using graphics processing units (GPU). It was shown that standard neural networks can be accelerated almost 20 times on GPUs [55-57]. Using this approach Alex Krizhevsky won the ImageNet Large Scale Visual Recognition Challenge 2012 [50]. There is also an architecture named after that – AlexNet. The scheme is presented in Fig. 8 [58].

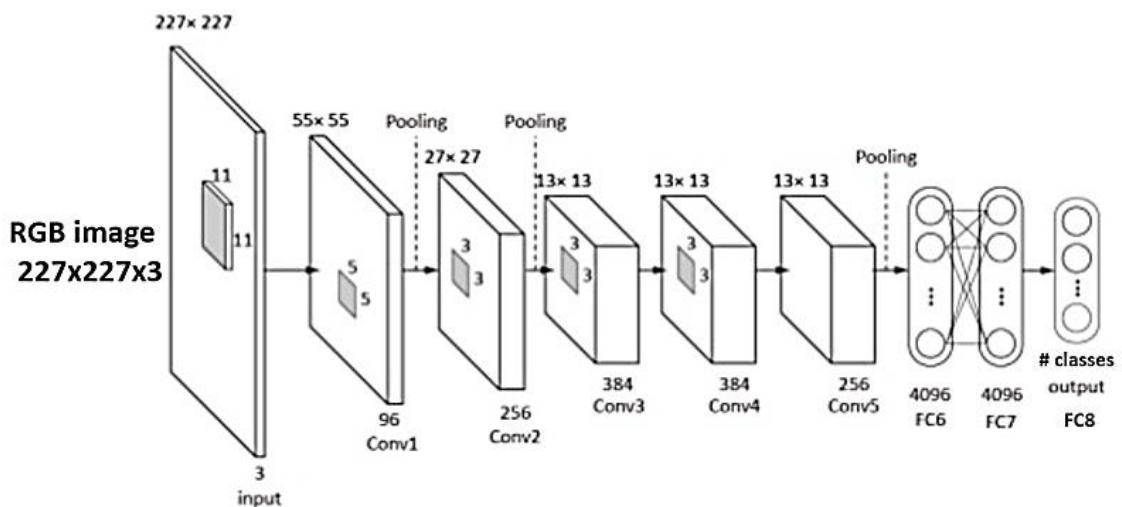


Fig. 8. AlexNet

After that models started to become more and more giant, the number of layers was dramatically increasing. For example, the 2015 ImageNet contest was won by the Microsoft model with over 100 layers [59].

Application of machine learning

Machine learning found applications in a variety of different fields. There are practical and strictly scientific fields.

Speaking about science it is possible to mention the application of such techniques in the field of atomic force microscopy [60, 61]. Results which were shown in this area helped to increase the accuracy and quality of measurements independently from a human. Another area of great results is material science, where machine learning helped to predict the properties of porous materials [62, 63]. Also, a great application was found in the field of microscopy to detect single particles [64, 65] and in work with femtosecond lasers [66]. But besides physics there are a lot of useful applications in biology, especially working with cells [67, 68].

Stepping aside from scientific fields, there are a lot of practical implementations of machine learning. For example, broad application machine learning is found in the banking area [69-71]. Helping feature machine learning played at linguistics area improving computational and statistical approaches [72-74]. And, machine learning was found useful in robotics researches [75-77].

Such a broad variety of application areas shows that machine learning has proven itself as a stable and well-resulted approach to problem solutions. There are still areas which are waiting for state of art solution of their problems, but the majority of the areas have successfully implemented machine learning solutions.

Experimental part

Training model

All code implementations and model training were provided in the Google Colaboratory cloud environment. The platform allows providing calculations using GPUs which significantly increase training speed.

Loss function and metrics

In the presented work a specific computer vision task was solved. The name of the task is image segmentation. There are classical and machine learning approaches for solving such a task. In this work machine learning approach is suggested.

Image segmentation itself is a process of separating an image into two or more segments. The main goal of this process is to simplify the image or change its representation for easier further analysis [78].

As it was said earlier every machine learning task has its specific metrics and loss functions. Image segmentation is not an exception. In this work was used three types of loss functions: binary-cross entropy loss (BCELoss), dice loss (DiceLoss), and focal loss (FocalLoss). These are three different functions and by optimizing them there are different possible results in quality that can be reached.

BCELoss can be represented using formula (3):

$$l_n = -w_n[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)] \quad (3)$$

This function is used for binary classification tasks and binary image segmentation is one of this kind because it is a problem of pixel classification – which pixel is needed object, and which is background.

Another possible loss function – DiceLoss – can be expressed as formula (4):

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \quad (4)$$

This loss function was firstly used in a computer vision contest in 2016 [79]. The motivation of using this loss function, but on BCELoss is that statistical

distribution of labels strongly influences training accuracy. If labels are imbalanced, then it will be hard to provide training of the model. And from formula (3) it is easy to see that cross-entropy loss considers the only micro sense of image rather than considering a global image, and it is not enough in the case of image-level prediction. In formula (4) “p” stands for prediction value and “g” stands for true value.

The last used loss function was FocalLoss, which takes for as formula (5):

$$FL = -(1 - p_t)^\gamma \log(p_t) \quad (5)$$

This function is usually used in case of imbalanced classes during training. Focal loss focusing during training in hard negative examples. In other words, it is dynamically scaled cross-entropy loss and the scaling factor can be zero in case of increasing confidence of correct class [80]. The focusing parameter in the formula (5) is γ and p_t is a probability of true class.

As it was said previously for the evaluation of machine learning results there should be metrics to calculate. In this work, IoU metrics is used. The metrics itself is presented in Fig. 9 [81].

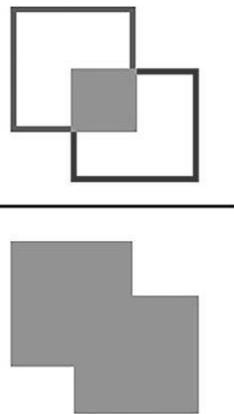
$$IoU = \frac{\text{area of intersection}}{\text{are of union}} = \frac{\text{area of intersection}}{\text{area of union}}$$


Fig. 9. IoU metrics

It is also known as the Jaccard index [82]. The calculation is provided in the following manner. For every image, there is a ground truth mask that was manually annotated and there is also a predicted mask from the model output. Using these two images and the calculating ratio presented in Fig. 9 it is possible to calculate how close these images are. For example, in an ideal case intersection will ideal, so it will be equal to the union of images and as the result, the IoU ratio will be equal to unity.

Comparison of configurations

In this work convolution, neural networks were used since the subject of the work is track membrane images. There are specific convolutional network architectures that are built to solve image segmentation tasks. One of the first breakthroughs in this branch of machine learning was U-Net architecture, which is presented in Fig. 10 [83].

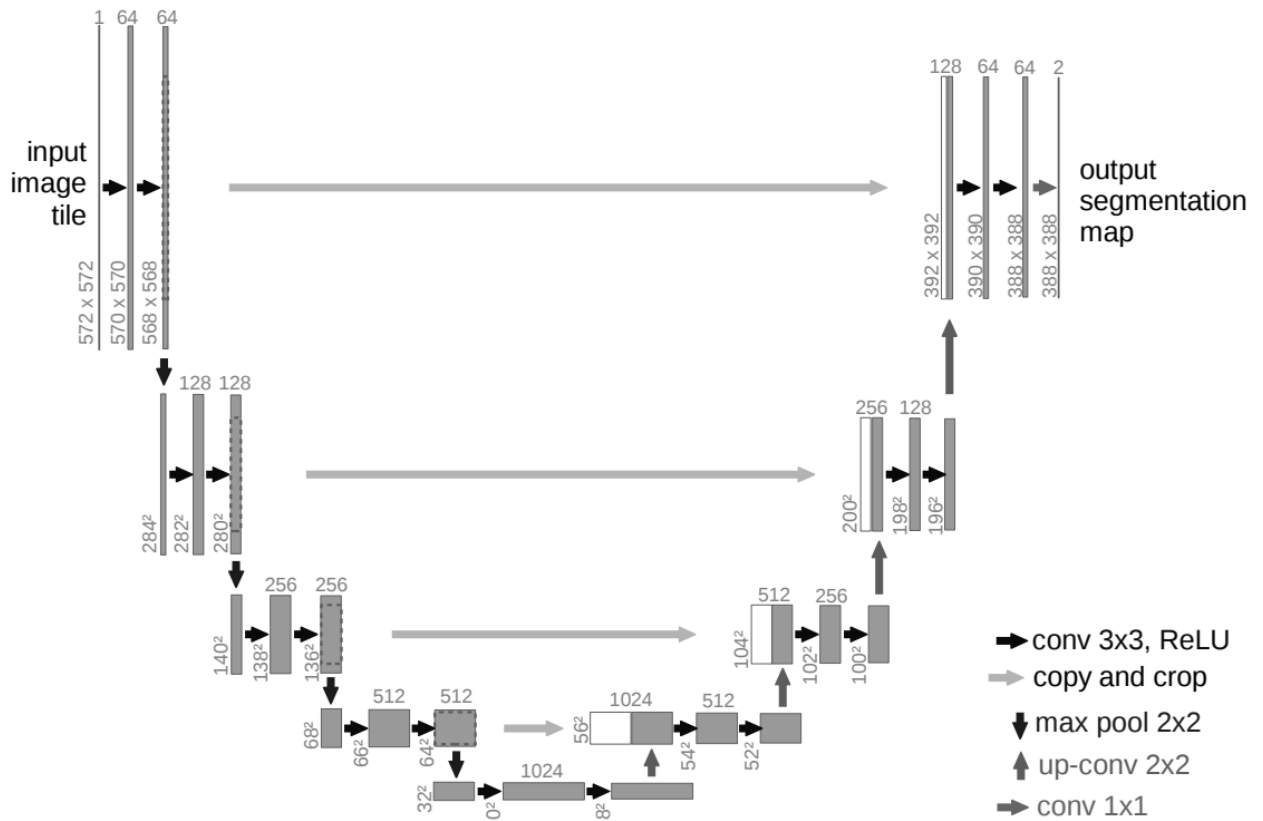


Fig. 10. U-Net architecture

The architecture itself is built from a contracting path and an expansive path, that is why the network is named U-Net, because of such structure it has a “u”-shape form. The first part of the network is a typical convolution network that iteratively uses convolution operations to reduce the size of the image. During this process extending of feature, information is happening. The next part of the network works upsampling operations instead of pooling operations. And replaces convolutions with up-convolutions. Also, an important part of the architecture is skip connections. They connect outputs of contracting paths with inputs of expansive paths and

concatenate. Using this approach, it is possible to expand the feature map of the model and get a more detailed map which is helpful in the case of the small size of the dataset.

But this architecture is not ideal and there is a modified version of it – Unet++, which is shown in Fig. 11 [84].

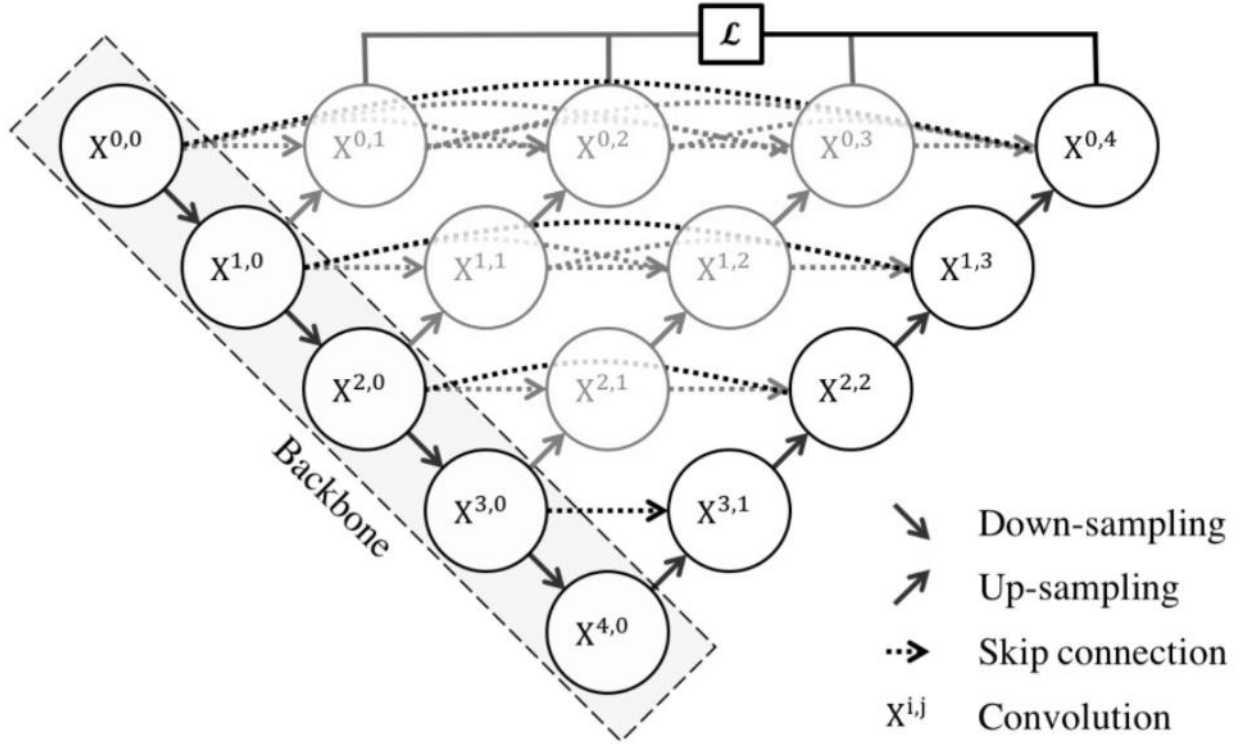


Fig. 11. UNet++ architecture

The UNet++ model has two principal differences from the U-Net model. Firstly, the presence of convolution layers in place of skip connections, which avoids a semantic gap between feature maps at the beginning and the end of the model. Secondly, the presence of fully connected layers in place of skip connections, which improves gradient optimization. And another innovative feature of this model is that it has not one, but four output and loss calculation can be provided using any of them and this allows to watch how the quality of the model changes with its depth.

The initial dataset consisted of 32 images of track membranes and all of them were manually annotated. After that using the augmentation method dataset was artificially enlarge to 608 images. The training was provided during 25 and 50 epochs which took about 10 minutes and 20 minutes respectively for the UNet model

and about 20 minutes and 40 minutes for the U-Net++ model. Training both models and changing loss functions was obtained comprising Table 1. The bold text corresponds to the best results of the model.

Model	Loss function	Epochs	Metrics value	
			Test score	Val score
Unet	BCELoss	25	0.86836	0.84576
		50	0.91750	0.91018
	DiceLoss	25	0.92085	0.91542
		50	0.97154	0.95578
	FocalLoss	25	0.77734	0.78690
		50	0.93594	0.90207
Unet++	BCELoss	25	0.95526	0.93887
		50	0.99024	0.98484
	DiceLoss	25	0.94966	0.94163
		50	0.98713	0.98378
	FocalLoss	25	0.94630	0.93150
		50	0.97177	0.96680

Table 1. Comparison of models

Training and metrics curves

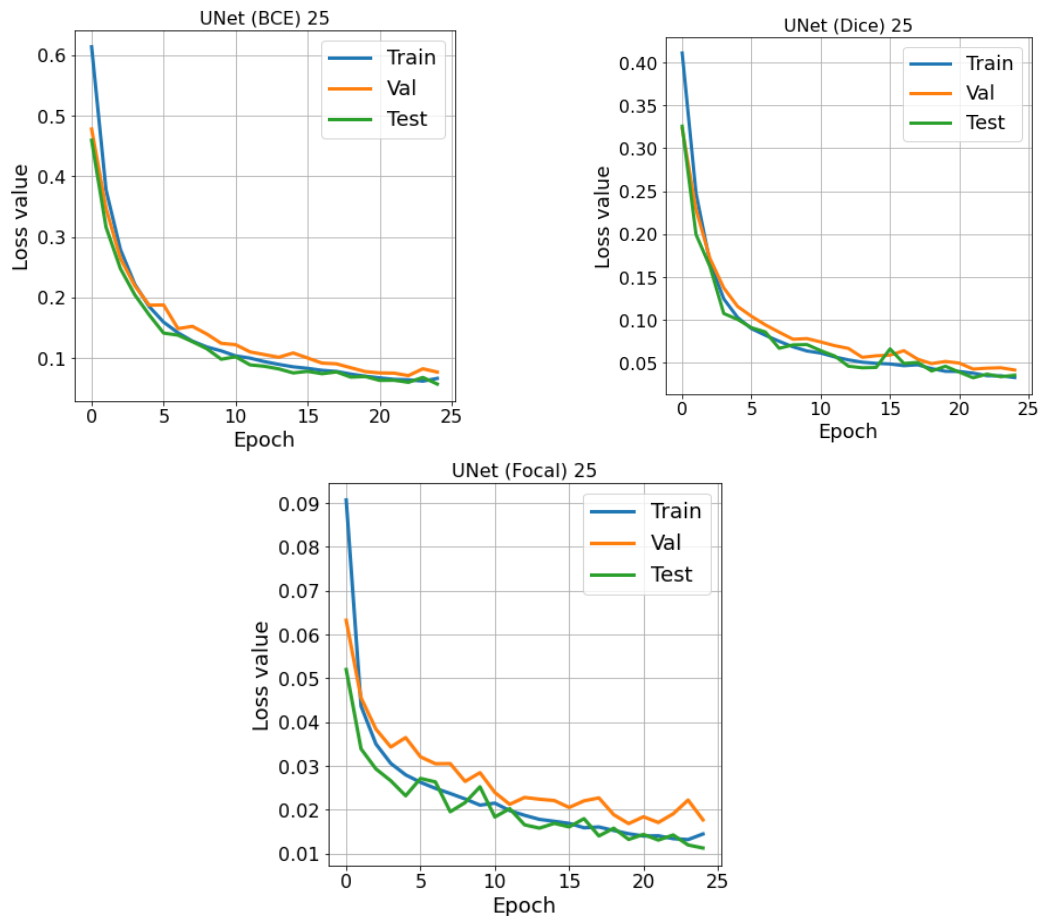


Fig. 12. Training curves during 25 epochs for U-Net model

For each training configuration, it is possible to provide training curves. Since the task is in optimizing loss function there must be decreasing character of the training curve. For the U-Net model trained during 25 epochs, there are the following curves (Fig. 12).

Fig. 13 presents training curves for the same configurations, but training was provided during 50 epochs.

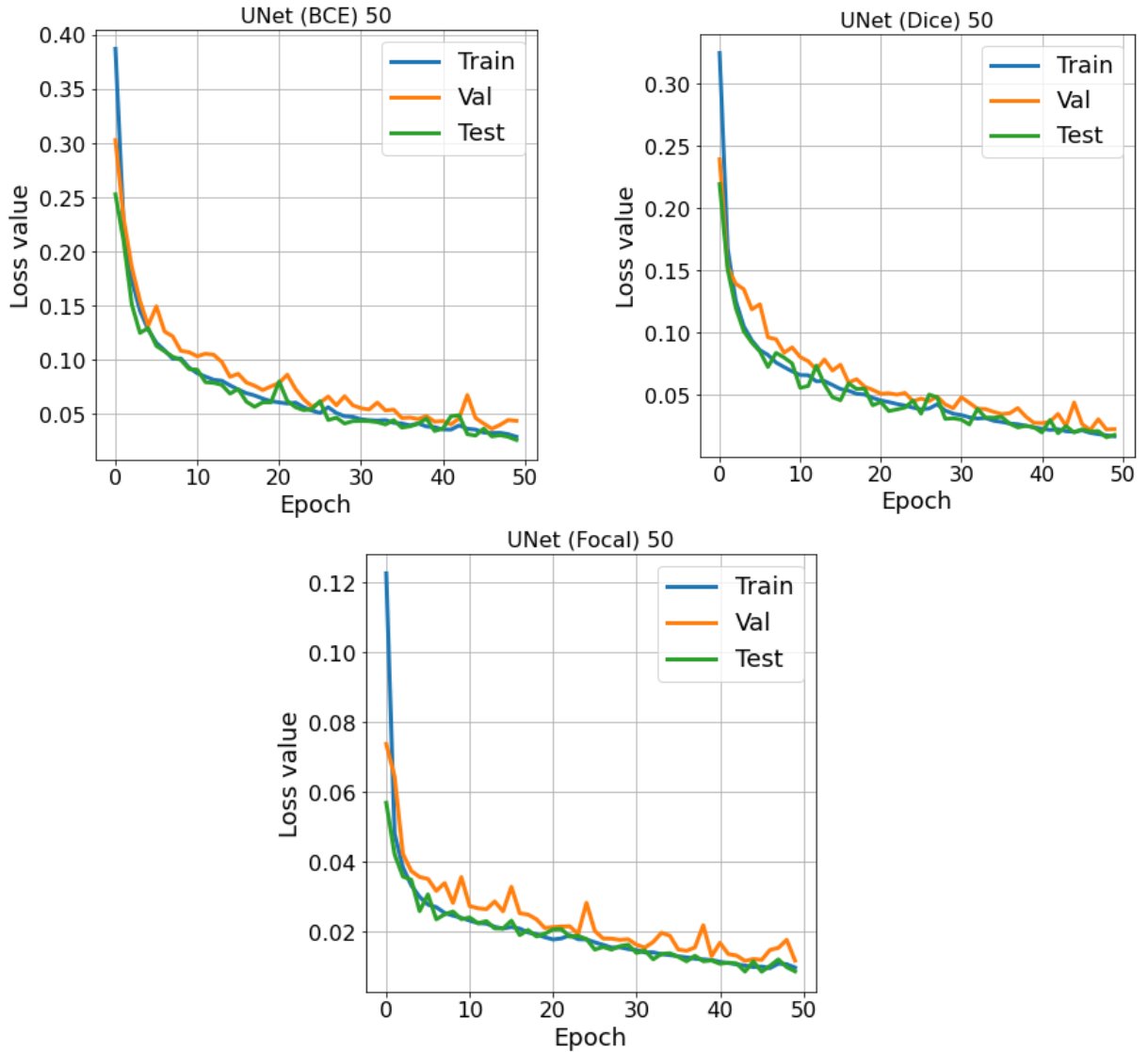


Fig. 13. Training curves during 50 epochs for U-Net model

It is easy to see that all parts of the dataset (train, validation, and test) decreasing as was expected. And, there is no overfitting since the training curve is below the validation curve.

Fig. 14 presents training curves for the UNet++ model which was trained for 25 epochs. Fig. 16 presented training curves for the UNet++ model which was trained for 50 epochs.

The behavior of the training curve of the UNet++ model is also decreasing and as expected. From curves, it is also obvious that no overfitting is presented.

After all training metrics curves were obtained for the two best configurations of U-Net and UNet++ models. Metrics was calculated on each epoch and then plotted. The result is depicted in Fig. 15.

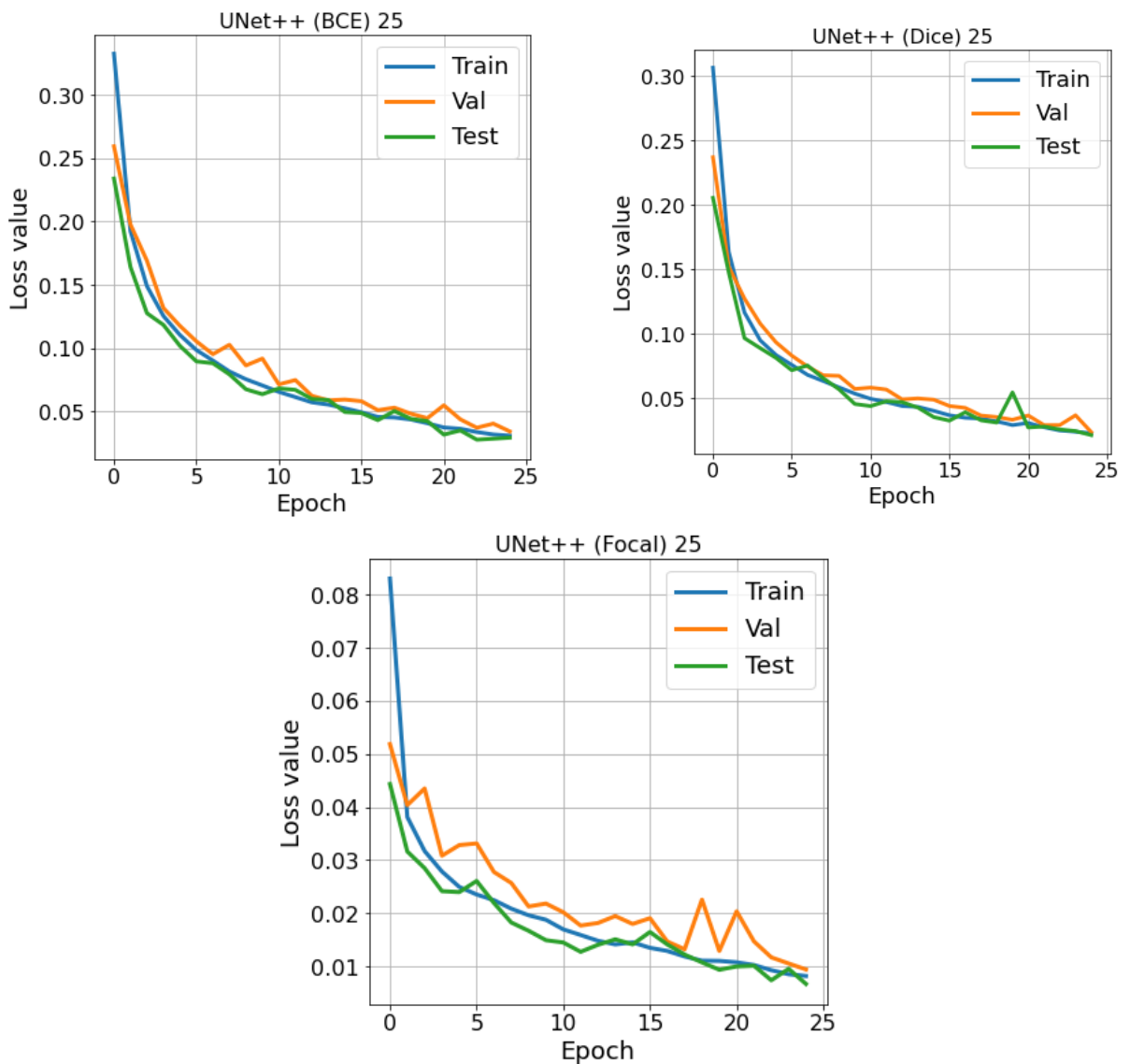


Fig. 14. Training curves during 25 epochs for UNet++ model

From curves, it is seen that good enough quality (about 80%) can be reached by the training model for about 10 epochs. But for usage in real-life tasks, it is not

significant, and the metrics required is about 95% in the ideal case. From these curves, it is seen that 50 epochs are enough to train the model with high quality and avoid overfitting the training dataset. It is important to note that there is no training curve in Fig. 16.

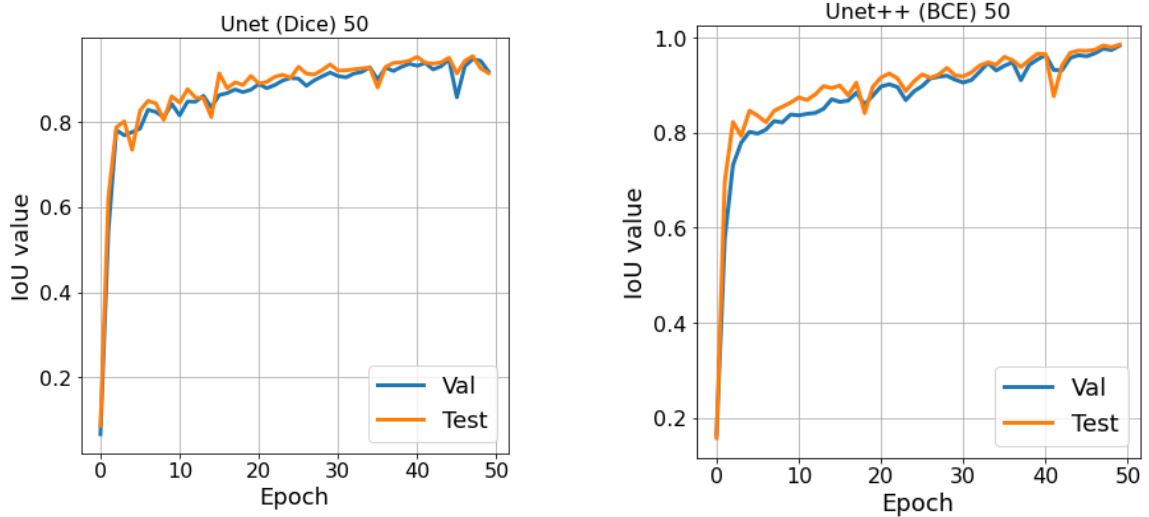


Fig. 15. Metric curves. Left – U-Net model, right – Unet++ model

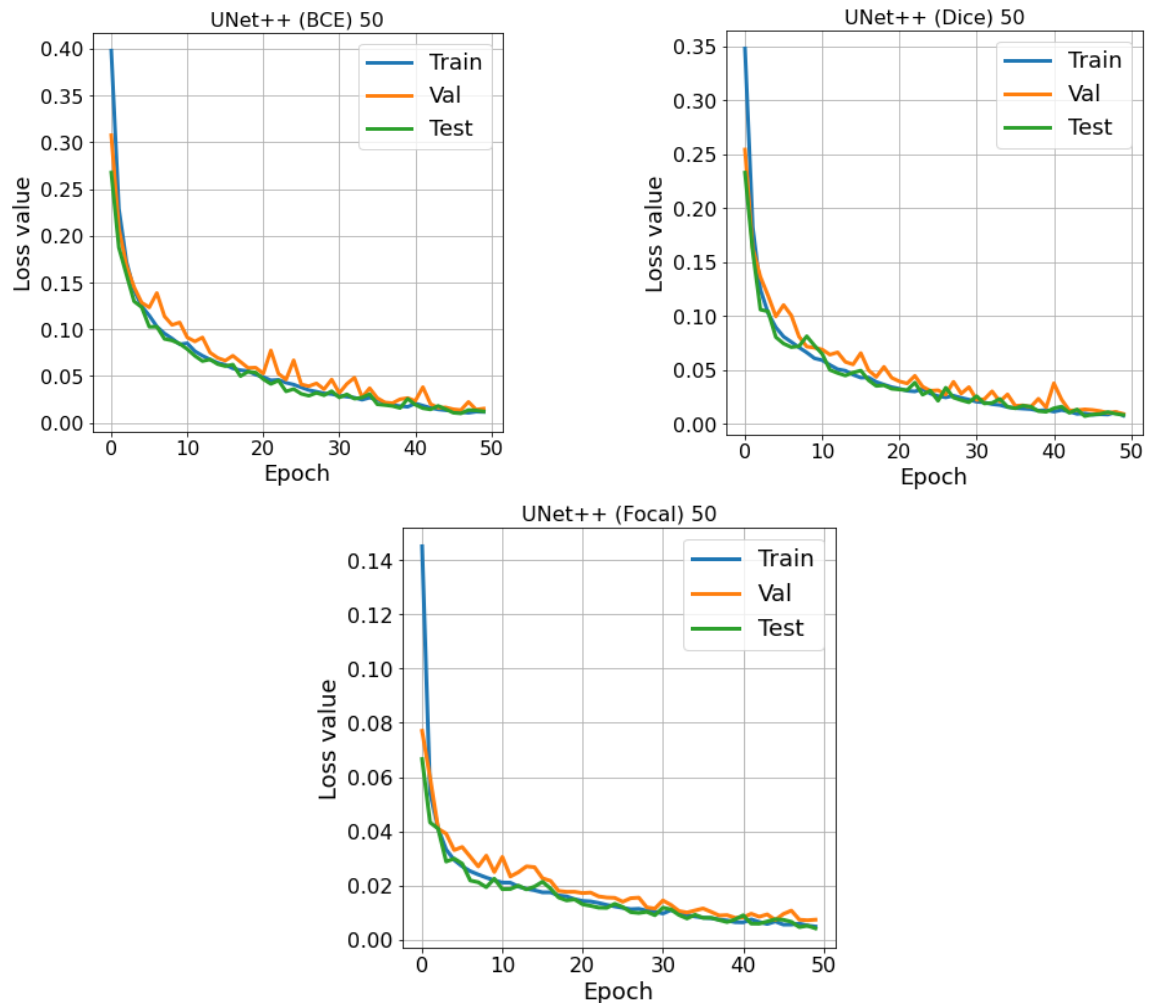


Fig. 16. Training curves during 50 epochs for UNet++ model

That is because there is no need to know how a good model works with a training dataset, it is important to know generalization error which can be seen on new data which is presented in test and validation datasets.

Classical computer vision approach

Porosity calculation

Porosity is a measure of empty spaces in the material. Speaking in terms of track membranes it is an area that is taken by pores of the membrane. Since using machine learning it is possible to convert an image of track membrane into the binary mask where white areas will stand for pores and black areas will stand for background. After that porosity is simply the total area of a white region divided by the total area of the image. Porosity is a dimensionless unit, but it can be expressed in percents.

Any image in computer representation is just a matrix with numbers. And in the case of a binary mask, these numbers are 0 and 1. It is possible to directly convert an image to an array of numbers and calculate how many ones in it. And after that divide this number by the total number of elements in the matrix and get a porosity.

Pore localization

For further analysis of the masks, the OpenCV library was used. The first step of the provided analysis is the localization of white regions which are pores of the membrane. Using implemented in OpenCV function named “findContuor”, it is possible to get all contours of white areas. The next step is to plot area distribution and find a median area for that distribution. To calculate areas of white regions “contourArea” function is used. Heuristically stated that median area will be equal to the area of one pore and any pore with an area smaller than half of the median area is not a pore. Using such an approach, it is possible to filter some artifact segmentations.

To localize received regions it is possible to use the “moments” function implemented in OpenCV to find the center of masses of all white regions. If it is a

single pore, then this function will show its center, but if pores are overlapped, then the function will show the center of masses of all overlapped pores.

Pore calculation

To calculate the number of pores on the image, a defined earlier median area is necessary. After that, another heuristic rule stated – if the area of the white region is greater than one and a half median area, then this region contains overlapping pores. Such a condition is taken from the assumption that membrane irradiation occurs under some angle and ions will not leave perfectly round pores, they more likely to leave elliptical pores, which can vary in the area comparing with circles. If the region satisfies that rule, then its area divides by the median area and round to the top. And using this method it is possible by iterating over all white regions to calculate the number of pores in the image of the track membrane.

Results and discussion

Image segmentation

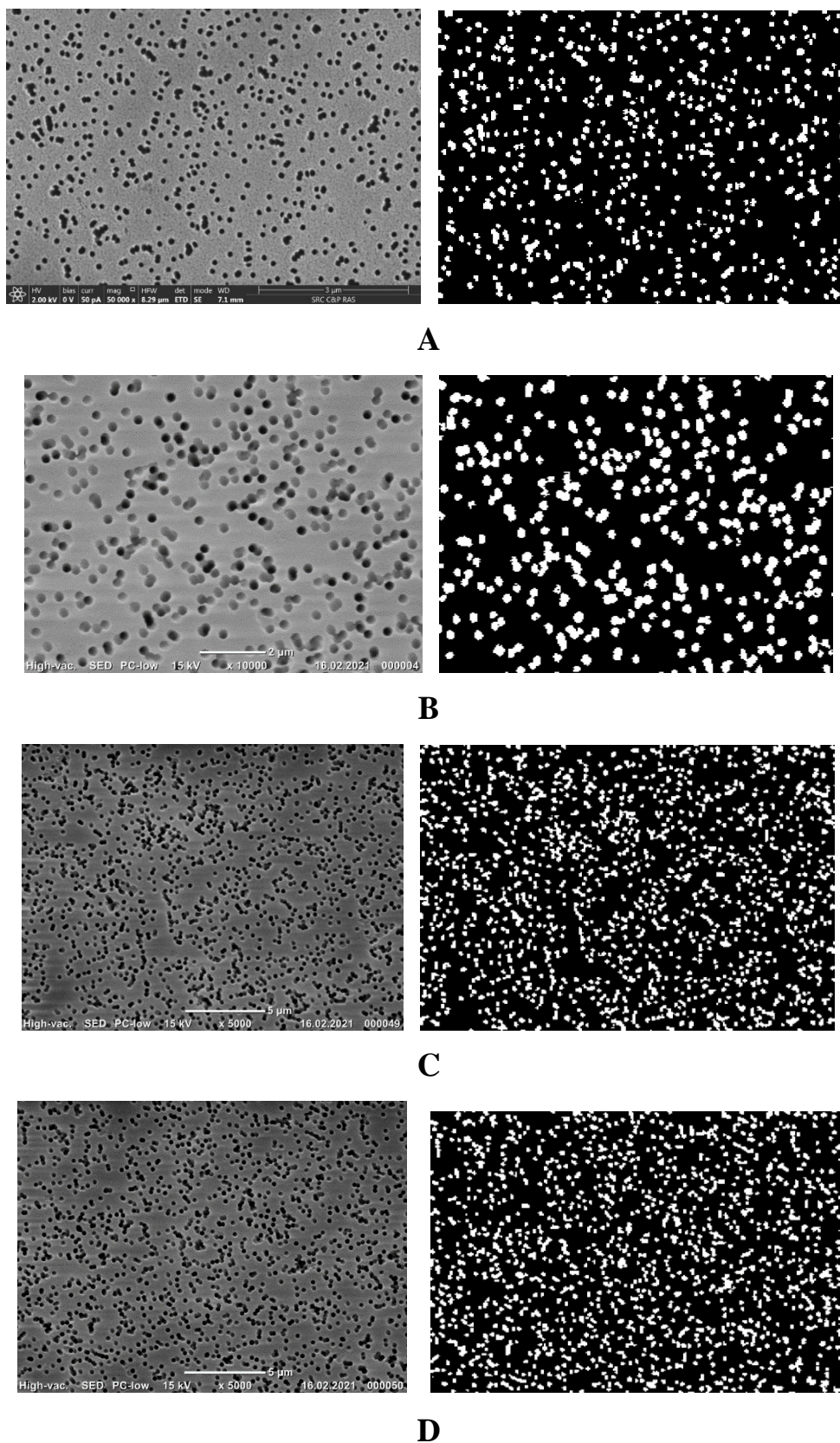


Fig. 17. Track membrane images and corresponding masks

For training chosen models described above loss functions were programmed and models' architectures were imported from the segmentation-models-pytorch library for Python programming language as well as IoU metrics [85].

Resultant masks can be seen in Fig. 17. The IoU metrics evaluated during training in maximum equals 0.99 for test dataset score and 0.98 for validation dataset score in case of UNet++ model. Mask can be generated by model on GPU in several seconds, which is fast comparing the time-quality ratio of other methods. Import to note that presented four images (Fig. 17) varies in color distribution and other image characteristics, but still, masks are reconstructed qualitatively which shows the perks of using the machine learning approach.

Mask analysis

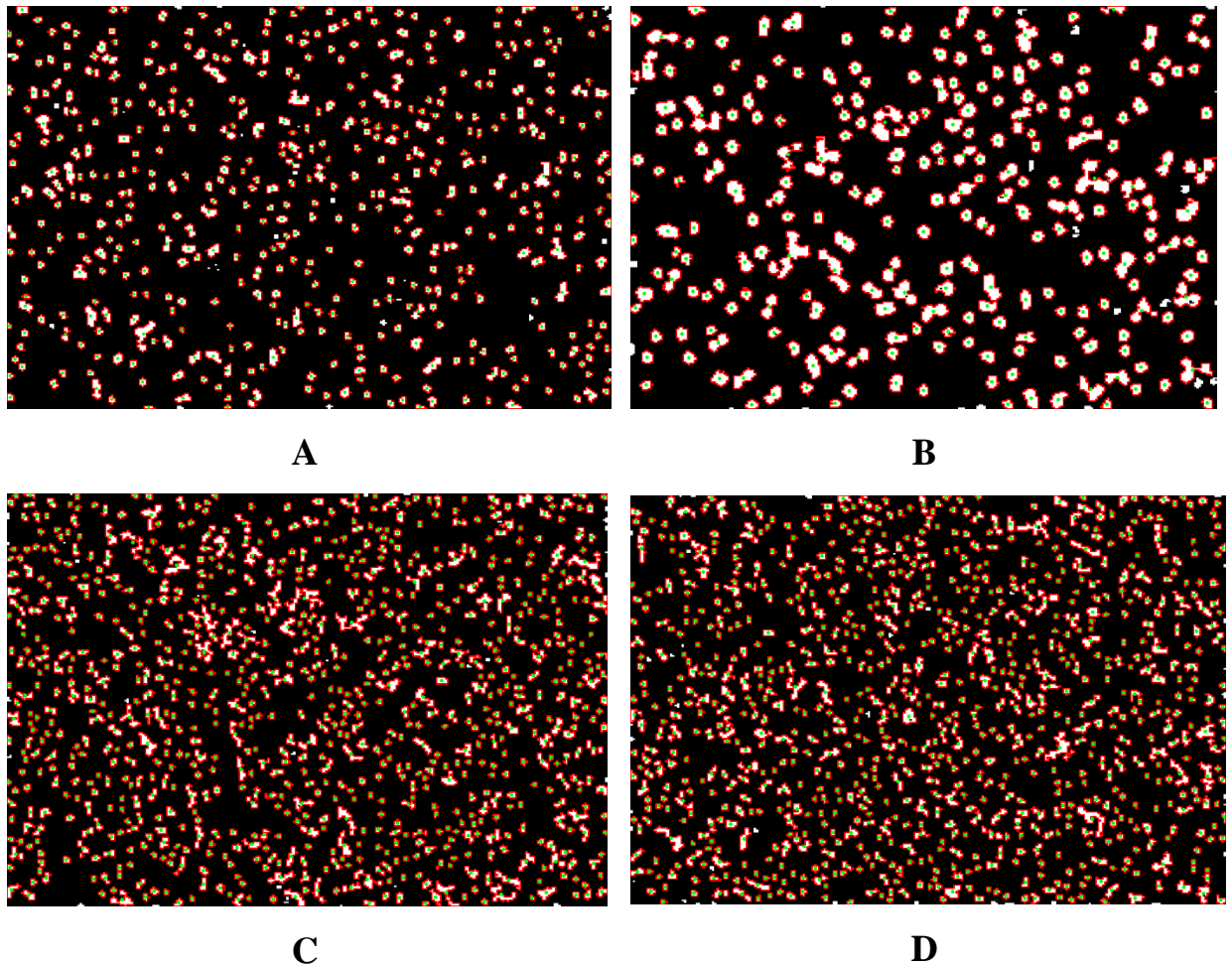


Fig. 18. Pores localization

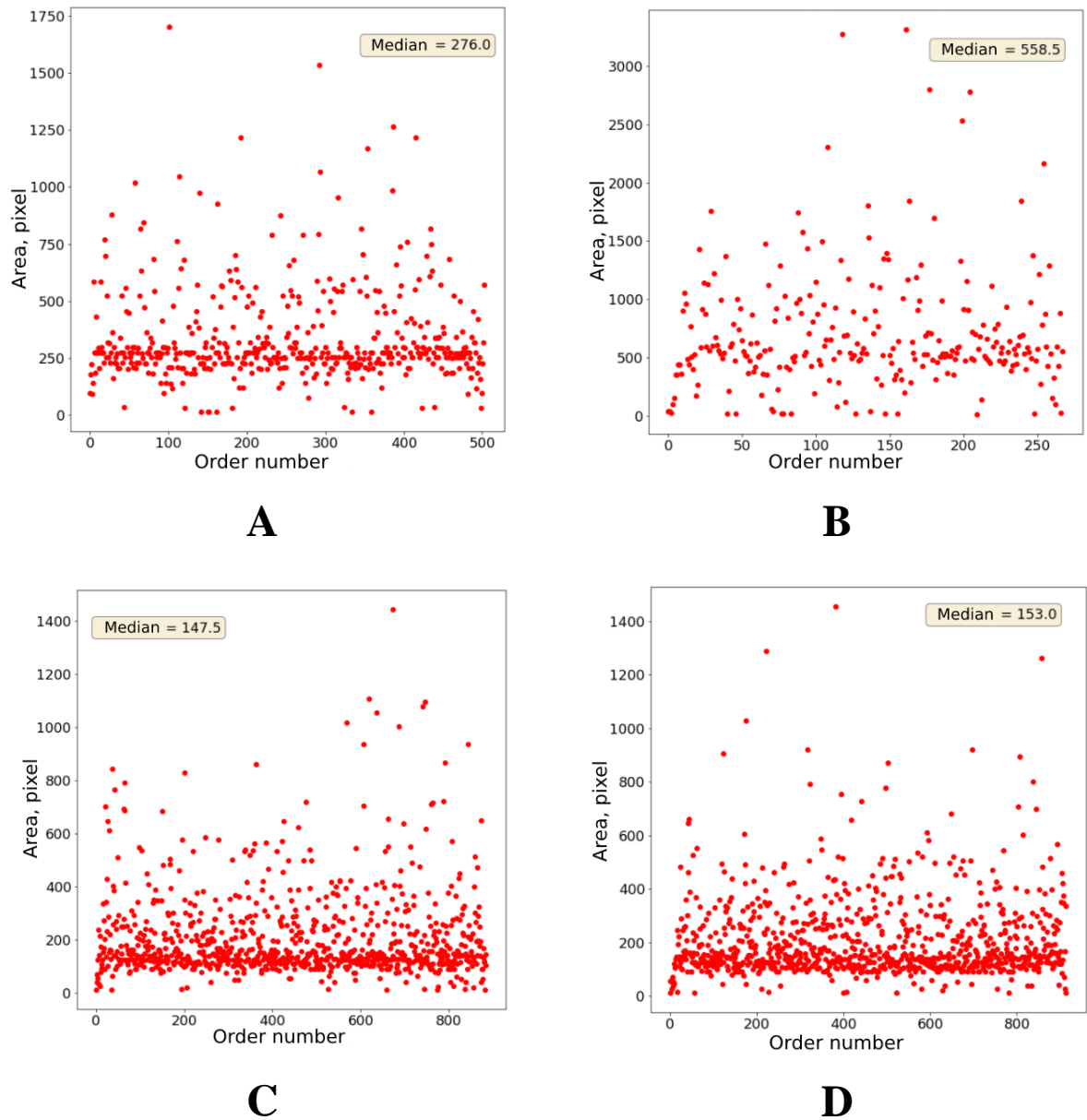


Fig. 19. Area distribution

After obtaining masks with the use of a trained model it is possible to calculate areas of white regions as was described above. Using implemented in Open CV library algorithms it is possible to find contours of all pores and their center of masses and after that calculate areas. Using the described heuristic rule, it is possible to filter pores to get rid of artifact segmentations. The result of pore localization is presented in Fig. 18.

Using calculate values of white regions' areas it is possible to plot area distribution to find median area, which is according to heuristic rule is the area of

one pore on the surface of the track membrane. This knowledge helps to calculate the number of pores and localize overlapping pores on the image.

Using another heuristic rule which was described above and iterating over all white regions classification of the region is provided. If the heuristic rule is satisfied then the region contains overlapping pores and the number of pores that overlapped can be approximately calculated by simply dividing the area of the region by the median area. Summing the number of calculating overlapped pores with single pores it is possible to find the total number of pores.

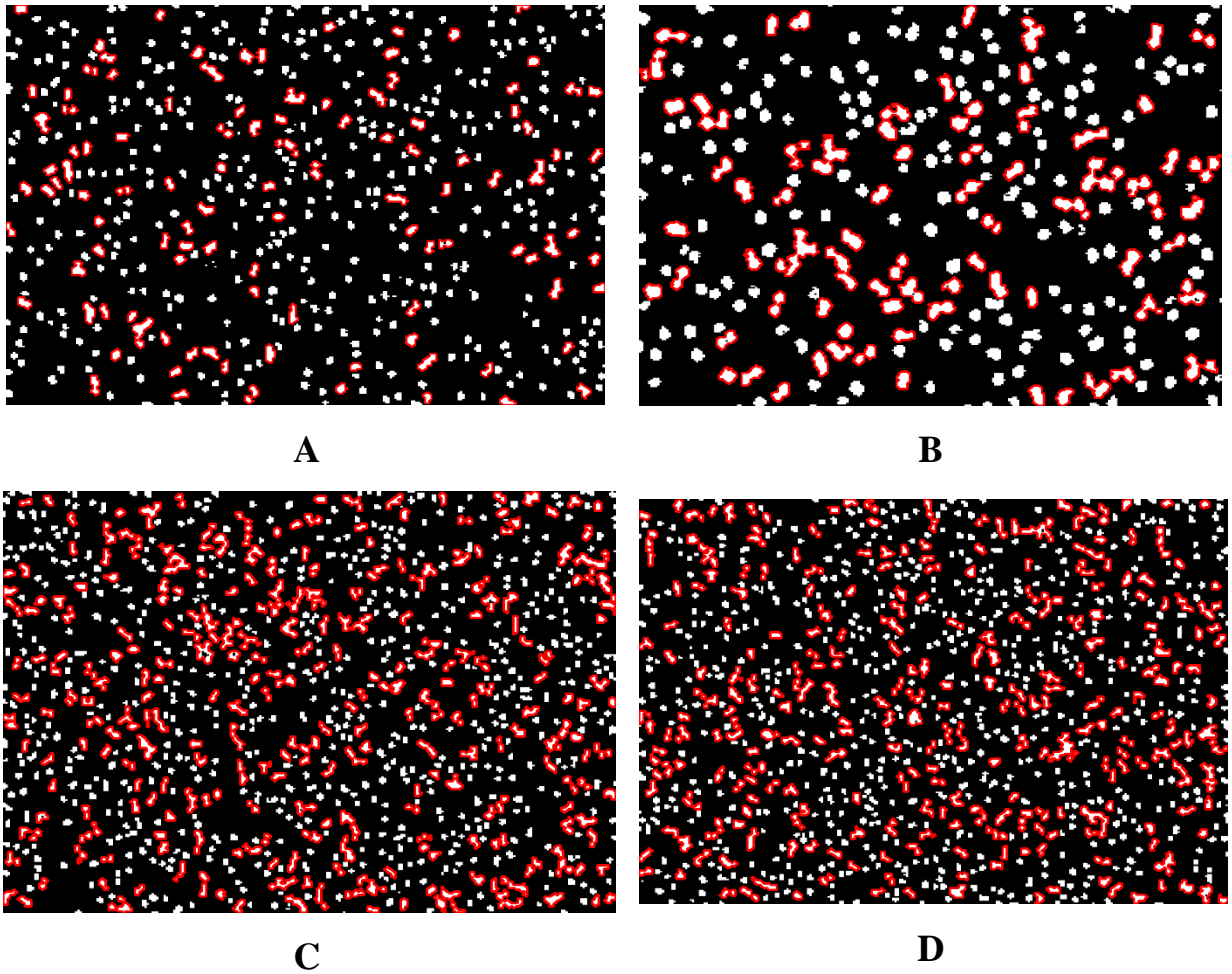


Fig. 20. Overlapping pores localization

Characteristics calculations

From provided mask analysis it is possible to calculate the characteristics of the track membrane surface. For example, porosity or the number of pores on the surface. The results of such calculations are presented in Table 2.

Sample	Method	Porosity, [%]	Number of pores
A	Manually	13.73415	624
	Using model	12.14294	633
B	Manually	22.00269	359
	Using model	18.06040	350
C	Manually	39.79699	1426
	Using model	19.31616	1257
D	Manually	39.25559	1405
	Using model	19.66472	1267

Table 2. Characteristic comparison

From the table, it follows that the use of a trained machine learning model helps to receive values of interesting characteristics with high accuracy corresponding to manual calculations, but the main advantage of this method is its speed. Using machine learning it is possible to calculate porosity as well as manually in case of a low density of track membrane pores. If the density of pores is high, then the model application shows better results comparing with manual calculation, because in manual case firstly number of pores calculated and after that, this number is multiplied by the area of one pore from the assumption that diameter is known from etching procedure. But such an approach does not consider overlapping pores, that is why porosity is bigger than it should be. On the other hand, calculation by trained model avoids this problem and calculates porosity more accurately. The same situation is seen in the case of pore calculation. If the pore density of the membrane is high enough, then the number of overlapping pores prevails, and the heuristic rule adds more error, in this case, comparing with the case of low pore density. From these results, it is possible to make a conclusion that the machine learning approach gives fast results with the same accuracy as manual calculation in the case of the low pore density of track membranes. In the case of high pore density, there could be errors in the number of pores calculation, but porosity calculation gives an even more accurate result than the manual approach.

Conclusion

In this work a machine learning approach to calculate characteristics of track membranes using images from electron-beam microscopy was provided. It was shown that using modern architectures like U-Net and UNet++, it is possible to realize qualitative image segmentation with the goal of simplification of further analysis. The reached results of IoU metrics are equal to 0.99 for the test part of the dataset and 0.98 for the validation part of the dataset. Also, it was shown that with help of the OpenCV library and implemented algorithms in it, it is possible to analyze masks, that was generated by the trained model, to calculate characteristics of track membrane surfaces. From comparison analysis of manual calculation and calculation with help of the model, it follows that suggested approach can be used for accurate calculation of porosity and number of pores in case of low pore density of membrane and with some errors it can be used for the number of pores calculation in case of high pore density of membrane. Also, there was shown that porosity calculation in the case of high pore density gives a more accurate value than manual calculation.

References

1. Francis M. R., Sarkar R., Roy S., Jaffar S., Mohan V. R., Kang G., Balraj V. Effectiveness of Membrane Filtration to Improve Drinking Water: A Quasi-Experimental Study from Rural Southern India // *Am J Trop Med Hyg.* – 2016. – T. 95, № 5. – C. 1192-1200.
2. Siami G. A., Siami F. S. Membrane Plasmapheresis in the United States: A Review Over the Last 20 Years //. – 2001. – T. 5, № 4. – C. 315-320.
3. Martin C. R. Membrane-Based Synthesis of Nanomaterials // *Chemistry of Materials.* – 1996. – T. 8, № 8. – C. 1739-1746.
4. Ion tracks and microtechnology: principles and applications. / Spohr R., 1990.
5. Gumirova V. N., Abdurashidova G. S., Bedin S. A., Zabalueva N. P., Kuvaitseva M. A., Razumovskaya I. V. Specific features of the fracture of track membranes and related polymer/metal composites prepared by template synthesis // *Physics of the Solid State.* – 2015. – T. 57, № 2. – C. 344-348.
6. Kozhina E. P., Bedin S. A., Nechaeva N. L., Podoynitsyn S. N., Tarakanov V. P., Andreev S. N., Grigoriev Y. V., Naumov A. V. Ag-Nanowire Bundles with Gap Hot Spots Synthesized in Track-Etched Membranes as Effective SERS-Substrates //. – 2021. – T. 11, № 4. – C. 1375.
7. Price P. B., Walker R. M. Molecular sieves and methods for producing same // *Book Molecular sieves and methods for producing same / Editor.* – USA, 1962.
8. Fleischer R. L., Price P. B., Walker R. M. Tracks of Charged Particles in Solids // *Science.* – 1965. – T. 149, № 3682. – C. 383-393.
9. Young D. A. Etching of radiation damage in lithium fluoride // *Nature.* – 1958. – T. 182, № 4632. – C. 375-7.
10. Westgate J., Sandhu A., Shane P. Fission-Track Dating // *Chronometric Dating in Archaeology / Taylor R. E., Aitken M. J.* – Boston, MA: Springer US, 1997. – C. 127-158.
11. Flerov G. N. // *Bulletin of the USSR Academy of Sciences.* – 1984. – T. №4.

12. Akapyev G. N., Barashenkov V. S., Samoylova L. I., Tretyakova S. P., Shchegolev V. A. On the method of manufacturing nuclear filters // JINR Publications Department. – 1974.
13. Fleisher R. L., Price P. B., Walker R. M. Nuclear Tracks in Solids (Principles & Applications) // Nuclear Technology. – 1981. – T. 54, № 1. – C. 126-126.
14. Apel P. Y., Fink D. Ion-Track Etching // Transport Processes in Ion-Irradiated Polymers. – Berlin, Heidelberg: Springer, 2004.
15. Apel P. Y. // Chemistry of High Energy. – 1991. – T. 25.
16. Membranes and membrane technologies. / Yaroslavtsev A. B. – Moscow: Scientific World, 2013.
17. Briggs D., Brewis D. M., Konieczo M. B. // Journal of Material Science. – 1976. – T. 11.
18. Gómez Álvarez-Arenas T. E., Apel P., Orelovitch O. L. Ultrasonic Propagation in the Micropores of Ion-Track Membranes // IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control - IEEE T ULTRASON FERROELECTR. – 2008. – T. 55.
19. Flerov G. N., Barashenkov V. S., Tretyakova S. P., Schegolev V. A. Method of manufacturing microfilters // Book Method of manufacturing microfilters / Editor. – USSR, 1974.
20. Luck H. B., Gemende B., Heinrich B. // Nuclear Tracks and Radiation Measurements. – 1991. – T. 19.
21. Cornell R. // Experimental Cell Research. – 1969. – T. 56.
22. Flerov G. N., Apel P. Y., Didyk A. Y., Kuznetsov V. I., Oganessian R. C. // Atomic energy. – 1989. – T. 67.
23. Nash G. B. // Clin. Hemorheol. – 1990. – T. 10.
24. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain // Psychol Rev. – 1958. – T. 65, № 6. – C. 386-408.
25. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biology. – 1990. – T. 52, № 1. – C. 99-115.

26. Cybenko G. Approximation by superpositions of a sigmoidal function // Mathematics of Control, Signals and Systems. – 1989. – T. 2, № 4. – C. 303-314.
27. Artificial intelligence : a modern approach. / Russell S. J.: Third edition. Upper Saddle River, N.J. : Prentice Hall, [2010] ©2010, 2010.
28. Foundations of Machine Learning. / Mohri M., Rostamizadeh A., Talwalkar A.: The MIT Press, 2012.
29. Machine Learning. / Mitchell T. M.: McGraw-Hill, Inc., 1997.
30. Introduction to Machine Learning. / Alpaydin E.: The MIT Press, 2010.
31. Computer Science Handbook, Second Edition. / Tucker A. B.: Chapman & Hall/CRC, 2004.
32. Ratner A., Bach S., Varma P., Ré C. Weak supervision: the new programming paradigm for machine learning //. – 2019. – C. 05-09.
33. Kaelbling L. P., Littman M. L., Moore A. W. Reinforcement learning: A survey //. – 1996. – T. 4. – C. 237-285.
34. Reddy S. M., Patel S., Weyrich M., Fenton J., Viswanathan M. Comparison of a traditional systematic review approach with review-of-reviews and semi-automation as strategies to update the evidence // Systematic Reviews. – 2020. – T. 9, № 1. – C. 243.
35. Optimization for Machine Learning. / Sra S., Nowozin S., Wright S. J.: MIT Press, 2012.
36. Stehman S. V. Selecting and interpreting measures of thematic classification accuracy // Remote Sensing of Environment. – 1997. – T. 62, № 1. – C. 77-89.
37. Powers D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation // Mach. Learn. Technol. – 2008. – T. 2.
38. Computer Vision Metrics: Survey, Taxonomy, and Analysis. / Krig S.: Apress, 2014.
39. Evaluation Metrics for Unsupervised Learning Algorithms. / Palacio Niño J., 2019.
40. Preuss B., Argiolas M. Customer Metrics and How Store Performance is Related to Them, 2016.

41. Hasegawa A., Lo S.-C., Lin J.-S., Freedman M., Mun S. A Shift-Invariant Neural Network for the Lung Field Segmentation in Chest Radiography // VLSI Signal Processing. – 1998. – T. 18. – C. 241-250.
42. Zhang W., Itoh K., Tanida J., Ichioka Y. Parallel distributed processing model with local space-invariant interconnections and its optical architecture // Applied Optics. – 1990. – T. 29, № 32. – C. 4790-4797.
43. Oord A. v. d., Dieleman S., Schrauwen B. Deep content-based music recommendation // Book Deep content-based music recommendation / Editor. – Lake Tahoe, Nevada: Curran Associates Inc., 2013. – C. 2643–2651.
44. Collobert R., Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning // Book A unified architecture for natural language processing: deep neural networks with multitask learning / Editor. – Helsinki, Finland: Association for Computing Machinery, 2008. – C. 160–167.
45. Tsantekidis A., Passalis N., Tefas A., Kannianen J., Gabbouj M., Iosifidis A. Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks // 2017 IEEE 19th Conference on Business Informatics (CBI). – T. 01 –, 2017. – C. 7-12.
46. Fukushima K., Miyake S., Ito T. Neocognitron: a neural network model for a mechanism of visual pattern recognition // Artificial neural networks: theoretical concepts IEEE Computer Society Press, 1988. – C. 136–144.
47. Hubel D. H., Wiesel T. N. Receptive fields and functional architecture of monkey striate cortex //. – 1968. – T. 195, № 1. – C. 215-243.
48. Matsugu M., Mori K., Mitari Y., Kaneda Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network // Neural Networks. – 2003. – T. 16, № 5. – C. 555-559.
49. Flexible, High Performance Convolutional Neural Networks for Image Classification. / Ciresan D., Meier U., Masci J., Gambardella L. M., Schmidhuber J., 2011. – 1237-1242 c.
50. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // Book ImageNet classification with deep

convolutional neural networks / Editor. – Lake Tahoe, Nevada: Curran Associates Inc., 2012. – C. 1097–1105.

51. Debnath S., Roy P. Speaker Independent Isolated Word Recognition based on ANOVA and IFS // Book Speaker Independent Isolated Word Recognition based on ANOVA and IFS / Editor. – Sydney, Australia: Association for Computing Machinery, 2018. – C. 92–97.

52. Ciregan D., Meier U., Schmidhuber J. Multi-column deep neural networks for image classification // 2012 IEEE Conference on Computer Vision and Pattern Recognition 10.1109/CVPR.2012.6248110 –, 2012. – C. 3642-3649.

53. Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE. – 1998. – T. 86, № 11. – C. 2278-2324.

54. A Shallow Convolutional Neural Network for Accurate Handwritten Digits Classification. / Golovko V., Egor M., Brich A., Sachenko A., 2017. – 77-85 c.

55. Steinkrau D., Simard P. Y., Buck I. Using GPUs for Machine Learning Algorithms // Book Using GPUs for Machine Learning Algorithms / EditorIEEE Computer Society, 2005. – C. 1115–1119.

56. Hinton G. E., Osindero S., Teh Y.-W. A Fast Learning Algorithm for Deep Belief Nets // Neural Computation. – 2006. – T. 18, № 7. – C. 1527-1554.

57. Oh K.-S., Jung K. GPU implementation of neural networks // Pattern Recognition. – 2004. – T. 37, № 6. – C. 1311-1314.

58. Khvostikov A., Aderghal K., Benois-Pineau J., Krylov A., Catheline G. 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies //. – 2018.

59. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR.2016.90 –, 2016. – C. 770-778.

60. Krull A., Hirsch P., Rother C., Schiffrin A., Krull C. Artificial-intelligence-driven scanning probe microscopy // Communications Physics. – 2020. – T. 3, № 1. – C. 54.

61. Huang B., Li Z., Li J. An artificial intelligence atomic force microscope enabled by machine learning // *Nanoscale*. – 2018. – T. 10, № 45. – C. 21320-21326.
62. Hebert V., Porcher T., Planes V., Léger M., Alperovich A., Goldluecke B., Rodriguez O., Youssef S. Digital core repository coupled with machine learning as a tool to classify and assess petrophysical rock properties // *E3S Web of Conferences*. – 2020. – T. 146. – C. 01003.
63. Chin C.-L., Yang Z.-Y., Su R., Yang C.-S. J. t. I. C. o. A. S., Technology. A Facial Pore Aided Detection System Using CNN Deep Learning Algorithm //. – 2018. – C. 90-94.
64. Zhu Y., Ouyang Q., Mao Y. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy // *BMC Bioinformatics*. – 2017. – T. 18, № 1. – C. 348.
65. Oktay A., Gürses A. Automatic Detection, Localization and Segmentation of Nano-Particles with Deep Learning in Microscopy Images // *Micron*. – 2019. – T. 120.
66. Xie Y., Heath D., Grant-Jacob J., Benita S., Michael D., Praeger M., Eason R., Mills B. Deep learning for the monitoring and process control of femtosecond laser machining // *Journal of Physics: Photonics*. – 2019. – T. 1.
67. Moen E., Bannon D., Kudo T., Graf W., Covert M., Van Valen D. Deep learning for cellular image analysis // *Nature Methods*. – 2019. – T. 16, № 12. – C. 1233-1246.
68. Nishida K., Hotta K. J. P. O. Robust cell particle detection to dense regions and subjective training samples based on prediction of particle center using convolutional neural network //. – 2018. – T. 13.
69. Turkson R. E., Baagyere E. Y., Wenya G. E. A machine learning approach for predicting bank credit worthiness // *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR) –IEEE, 2016*. – C. 1-7.
70. Munkhdalai L., Munkhdalai T., Namsrai O.-E., Lee J. Y., Ryu K. H. J. S. An empirical comparison of machine-learning methods on bank client credit assessments //. – 2019. – T. 11, № 3. – C. 699.

71. Ruangthong P., Jaiyen S. Bank direct marketing analysis of asymmetric information based on machine learning // 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE) –IEEE, 2015. – C. 93-96.
72. Manning C. D. J. C. L. Computational linguistics and deep learning //. – 2015. – T. 41, № 4. – C. 701-707.
73. Rawski J., Heinz J. J. L. No free lunch in linguistics or machine learning: Response to pater //. – 2019. – T. 95, № 1. – C. e125-e135.
74. Korhonen A. Automatic Lexical Classification–Balancing between Machine Learning and Linguistics // Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1 –, 2009. – C. 19-28.
75. Mosavi A., Varkonyi-Koczy A. R. Integration of machine learning and optimization for robot learning // Recent Global Research and Education: Technological ChallengesSpringer, 2017. – C. 349-355.
76. Wang Y., de Silva C. W. J. E. A. o. A. I. A machine-learning approach to multi-robot coordination //. – 2008. – T. 21, № 3. – C. 470-484.
77. Granda J. M., Donina L., Dragone V., Long D.-L., Cronin L. J. N. Controlling an organic synthesis robot with machine learning to search for new reactivity //. – 2018. – T. 559, № 7714. – C. 377-381.
78. Computer Vision. / Shapiro L. G., Stockman G. C.: Prentice Hall, 2001.
79. Milletari F., Navab N., Ahmadi S. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation // 2016 Fourth International Conference on 3D Vision (3DV) 10.1109/3DV.2016.79 –, 2016. – C. 565-571.
80. Lin T., Goyal P., Girshick R., He K., Dollár P. Focal Loss for Dense Object Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2020. – T. 42, № 2. – C. 318-327.
81. Bozkir A., Aydos M. LogoSENSE: A Companion HOG based Logo Detection Scheme for Phishing Web Page and E-mail Brand Recognition // Computers & Security. – 2020. – T. 95.
82. Jaccard P. THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1 //. – 1912. – T. 11, № 2. – C. 37-50.

83. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 – Cham: Springer International Publishing, 2015. – C. 234-241.
84. Zhou Z., Rahman Siddiquee M. M., Tajbakhsh N., Liang J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support – Cham: Springer International Publishing, 2018. – C. 3-11.
85. Yakubovskiy P. Segmentation Models Pytorch // GitHub repository. – 2020.