COMPUTER DESIGN SYSTEMS. THEORY AND PRACTICE

Vol. 6, No. 1, 2024

Denys Manokhin[1], Yaroslav Sokolovskyy[2]

[1] Department of Information Systems, Ivan Franko National University, University street 1, Lviv, Ukraine, E-mail: Denys.Manokhin@lnu.edu.ua, ORCID 0000-0002-8590-7626
[2] Computer Aided Design Department,Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine, E-mail: yaroslav.i.sokolovskyi@lpnu.ua, ORCID 0000-0003-4866-2575

INTRACRANIAL HEMORRHAGE SEGMENTATION USING NEURAL NETWORK AND RIESZ FRACTIONAL ORDER DERIVATIVE-BASED TEXTURE ENHANCEMENT

Received: March 04, 2024 / Revised: April 01, 2024 / Accepted: April 05, 2024

https://doi.org/

Abstract. This paper explores the application of the U-Net architecture for intracranial hemorrhage segmentation, with a focus on enhancing segmentation accuracy through the incorporation of texture enhancement techniques based on the Riesz fractional order derivatives. The study begins by conducting a review of related works in the field of computed tomography (CT) scan segmentation. At this stage also a suitable dataset is selected. Initially it is used to train the U-Net, one of the widely adopted deep learning models in the field of medical image segmentation. Training is performed using parallel algorithm based on CUDA technology. The obtained results are compared with the established baseline for this dataset, assessing segmentation accuracy using the Jaccard and Dice coefficients. Subsequently, the study investigates a texture enhancement technique based on the Riesz fractional order derivatives, applied to the CT-scan images from the dataset. This technique aims to capture finer details and subtle textures that may contribute to improved segmentation accuracy. The U-Net model is then retrained and validated on the texture-enhanced images, and the experimental results are analyzed. The study reveals a modest yet notable enhancement in accuracy, as measured by the Jaccard and Dice coefficients, demonstrating the potential of the proposed texture enhancement technique in refining intracranial hemorrhage segmentation.

Keywords: fractional order derivative, deep learning, Python, PyTorch framework, CUDA technology, texture enhancement, medical image segmentation.

Introduction

Timely diagnosis is extremely important for successful treatment of intracranial hemorrhages. Computed tomography (CT) is most often used for this purpose. CT-scans are analyzed by therapeutic radiologists. Since the analysis is carried out manually, it can take a lot of time, which is precious for saving the patient's life, and the correctness of the diagnosis depends entirely on the doctor's qualifications. Automating this process with a sufficient level of accuracy to completely trust the patient's life to it will probably not be possible soon. Therefore, the current goal is to develop algorithms that are reliable enough to help the radiologist make a decision by identifying suspicious areas on the image.

Modern architectures of artificial neural networks (ANNs) demonstrate fairly good accuracy in segmenting medical images. However, there is still an opportunity to improve the results by using computer vision algorithms that consider the specificity of the data.

Medical images typically have low resolution, high noise levels, and complex textures. At the same time, they carry huge amounts of information. The presence of noise greatly complicates the interpretation of such images. Therefore, texture enhancement and noise reduction is a key step in medical image analysis.

Fractal analysis methods are actively used in the field of digital signal processing and have demonstrated good results in combating noise of a complex nature. Since an image can be viewed as a two-dimensional signal, some of these methods have been adapted to solve computer vision problems. Namely, several methods of image quality improvement based on derivatives of fractional order have been proposed. In this article, one of these methods was implemented, and its influence on the automatic segmentation of intracranial hemorrhages on CT-scans using ANN was investigated.

Problem Statement

The aim of this work is to develop software and algorithms for medical image texture enhancement using fractal analysis methods and apply them to solving a practical problem of intracranial hemorrhage segmentation based on CT data. To achieve this goal, the following tasks were performed:

• The existing developments in the field of application of artificial neural networks to intracranial hemorrhage segmentation according to CT data were analyzed.

• A set of CT-scans of patients with intracranial hemorrhages correctly identified by professionals was selected for training of ANNs.

• Among the ANN architectures, the one that is most suitable for solving the problem was selected. A parallelized ANN training algorithm was developed using CUDA technologies. The optimal parameters were selected to maximize the accuracy of the results.

• Algorithms for texture enhancement based on fractal analysis methods were analyzed. The algorithm was implemented and applied to the selected dataset. The impact of this algorithm on the accuracy of segmentation of intracranial hemorrhages was investigated.

Review of Modern Information Sources on the Subject of the Paper

Classification and segmentation of intracranial hemorrhages based on CT scans using ANN

Automatic segmentation of intracranial hemorrhages based on CT-scan data has not been sufficiently studied to date. The problem is that, as with many medical image processing tasks, until recently there were no publicly available datasets with labeled areas of hemorrhage. The reason for this is both the difficulty of preparing such information in a legal environment, since the images may contain personal data of patients, and the fact that the segmentation task requires professionals to spend a lot of time manually marking problem areas on the scans.

Since it is easier to mark only the presence and type of hemorrhage on an image than to indicate its location, datasets for the task of classifying CT-scans are more common. A very popular dataset for validating the classifier is CQ500, a set containing 491 CT-scans of the head, for each of which the corresponding type of hemorrhage or its absence is indicated. Very good results were obtained for it with about 95% recognition accuracy for each class [1]. Also recently, a competition organized by the Radiological Society of North America (RSNA) was held on the Kaggle platform, for which a dataset with more than 25,000 images was made public [2].

As can be seen, the availability of a large amount of data in the public domain has made it possible to make significant progress in solving the problems of classifying CT-scans of the brain for the presence of one or another type of hemorrhage. While segmentation is more challenging, mostly due to the lack of standard datasets suitable for this task. As shown by Hssayeni and his colleagues in their review of modern methods of segmentation of intracranial hemorrhages [3], both methods based on classical computer vision algorithms and methods based on deep learning are quite successfully used to solve this problem. However, the existing methods were tested on different datasets with limited access, so we cannot properly compare these models on the same dataset. Therefore, the same authors created an open dataset for training and validation of segmentation models of intracranial hemorrhages [4]. It was published on the PhysioNet platform [5].

The latter dataset was chosen for this study. Its structure is described in detail in paragraph 3.1.

Analysis of results for the selected dataset

In the mentioned paper [3], in addition to the creation of the dataset, the authors also described the process of training the U-Net network on this dataset and provided estimates of the accuracy of the output of the trained model in various metrics.

The U-Net network proposed by Olaf Ronnenberger et al. 2015 [6], is a fully convolutional network designed specifically for medical image segmentation tasks. Its feature is that it can learn effectively even on small datasets, as in the field of medicine there are often problems with the availability of large datasets. In fact, this network has become a classic architecture for medical image segmentation, which is used as the basis for many more modern methods.

Separate layers selected from three-dimensional scans with a window (WL: 40, WW: 120) were used for training, and samples both from patients with and without hemorrhage were used. Since there are much more samples without hemorrhage, only a certain randomly selected part of them was used. Also, since there is not much data, and due to the wide age range, the skull sizes are quite different, random affine transformations were applied to the data during training to improve the generalization of the model.

Two experiments were conducted, both with full images and with images divided into 160x160 blocks. More attention was paid specifically to the second method.

The training was carried out for 150 epochs using an Nvidia GeForce RTX2080 GPU with 11 GB of memory. About 5 hours were spent on this.

To train the model, the Adam optimizer was used with a learning coefficient of 10^{-5}. Cross entropy was used as the loss function. The training took place with a batch size of 32.

The results of testing the trained network can be seen in Table 1. To better understand the result, the authors provided the minimum, maximum and average value of each metric, as well as their standard deviations calculated for all images from the test dataset. The Dice and Jaccard coefficients have small values, but are commensurate with other experiments where the network was trained on a fairly small dataset.

A high sensitivity value is also an advantage, because in the field of medicine, it is better for the model to give a "false alarm" than for it to miss a really problematic area. Therefore, it can be said that in this paper the authors showed the possibility of creating an ANN to assist radiologists and provided a basic model that other researchers can rely on when developing improved algorithms.

Table 1

Results of testing the trained ANN presented in [3]

Test dataset statistics	Jaccard coefficient	Dice coefficient
Min	0.00	0.00
Max	0.528	0.677
STD	0.163	0.211
Average	0.218	0.315

Image quality improvement based on fractional order derivatives

Most popular methods for image improvement are based on integer order derivatives. They are computed for representing an image using the Sobel, Roberts, or Prewitt masks (also known as filters, operators, or templates). However, such operators have some drawbacks, the biggest of which is high sensitivity to noise [7].

Integer-order operators in computer vision are also often called high-pass filters because they amplify the intensity of high-frequency components of an image while attenuating low frequencies.

Image frequencies determine the change in gray levels relative to changes in distance. For example, edges and noise are high-frequency components because they are characterized by a significant change in pixel intensity over a short distance. On the contrary, textures and backgrounds are examples of low-

frequency components, since they are characterized by slight changes in gray levels [8].

Since noise is characterized by a high frequency, applying a high-pass filter to the image, in addition to enhancing the elements of interest to us, will significantly increase the noise level. This drawback can be overcome by the methods proposed in 2010 by Yui-Fei Pu et al. [9] — masks based on fractional order derivatives (also known as the YuiFeiPU operators). In the article, the authors offer several options for constructing such masks based on the Riemann-Liouville and Grünwald-Letnikov derivatives.

Such a mask preserves low frequencies in smooth areas of the image and high frequencies are preserved where there are significant changes in intensity, while enhancing texture details in areas where changes in intensity are less noticeable.

The mask proposed by Yui-Fei consists of 8 parts. For the left-hand derivatives along the x-axis, the y-axis, and the two diagonals and the corresponding right-hand derivatives. This ensures that the resulting operator is rotation-independent.

However, as shown in [7], the use of the Riesz fractional derivative in the proposed masks makes it possible to further improve noise resistance. The authors named the modified operator FCD. Table 2 shows the Signal to Noise Ratio — SNR obtained for different operators, as well as for the FCD operator with different mask sizes.

Table 2

SNR for different operators and mask sizes [7]

Operator	SNR	Mask size	SNR
Sobel	3.5	3x3	13.3
Laplacian	3.3	5x5	10.2
YiFeiPU	8.3	7x7	8.8
FCD	10.2	9x9	8.0

Objectives and Problems of Research

In [9] it is assumed that fractional order operator can be useful when working with CT-scans, and in [7] and [8] some fractional operators were applied to improve the texture of medical images. However, as a result of the literature review, no studies were found on the impact of such image enhancement on automatic segmentation using ANN. Therefore, the purpose of this work is to analyze the influence of the fractional order derivative operator on the quality of intracranial hemorrhage segmentation by using the U-Net. Namely, according to Table 2, the FCD operator was considered.

Main Material Presentation

Dataset for segmentation of intracranial hemorrhages

The data to create the dataset was collected between February and August 2018 at the Al Hillah Hospital, Iraq. A Siemens / SOMATOM Definition AS computer tomograph was used for scanning. A total of 82 patients were examined, including 46 men and 36 women, with an average age of 27.8±19.5 years. The total age range is from 1 day to 72 years. There were 27 patients under 18 years old, and 55 older ones. Intracranial hemorrhage was found in 36 people, and skull fracture - in 22 [3]. If we consider individual layers, we get 318 images with existing hemorrhage and 2,491 - without them.

The images are presented in the NIfTI format with an extension of 512x512 and a variable number of layers for each patient (usually about 30). In addition, each image corresponds to a mask where, in case of hemorrhage, the corresponding pixels are highlighted in white. Also, a csv file is provided with the data, in which the type of hemorrhage or its absence is indicated for each layer of each image.

The U-Net network architecture

U-Net network consists of two symmetrical parts which are often called encoder and decoder. The encoder works like a classic CNN, where each layer reduces the resolution of the input image and increases the number of feature maps at the output. Due to this, we obtain many low-resolution feature

maps from a given image. The decoder, on the contrary, restores a full-size image from the feature maps received at the input. Both the encoder and the decoder use blocks of two consecutive convolutions with a filter size of 3x3 and the ReLU activation function. However, in the encoder, the 2x2 max pooling is used between blocks to reduce the resolution, and in the decoder; on the contrary, the 2x2 inverse convolution is used.

In addition, the corresponding feature maps from the encoder are concatenated to the input feature maps for each decoder block, which makes it possible to transmit additional information about low-level features and improve the quality of the result [6].

It should be noted that modern modifications of U-Net are usually somewhat different from the classic version of this network. One of these modifications was used in this work. In it, before convolutions, frames are added to the input data in order to obtain an output image of the same size as at the input. Also, instead of inverse convolution, bilinear interpolation is performed. In addition, batch normalization and dropout are used in blocks with double convolution.

To assess the accuracy of segmentation, two popular metrics were used — the Jaccard and Dice coefficients.

The Jaccard coefficient between the true image (y) and the resulting image at the output of the ANN $(\hat{y})$ is defined as the ratio between the number of pixels belonging to the intersection of the selected segment masks and the total number of pixels of these segments

$$I(y,\hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|}. \tag{1}$$

The Dice coefficient is defined as twice the number of pixels of intersection of segments, divided by the sum of the numbers of pixels of each segment

$$D(y,\hat{y}) = \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}. \tag{2}$$

Algorithm for improving image quality based on the Riesz derivative

This section presents the main stages of the algorithm for applying the FCD operator. This section describes the main stages of the algorithm for applying the FCD operator [10]. We also use the approximation dependencies of the Riesz operator (3). Other approximations were used in [11, 12].

The FCD operator is based on the Riesz derivative. The Riesz fractional derivative D_R^{ν} of ν order for an infinite interval $-\infty<x<+\infty$ is given by the formula:

$$\mathrm{D}_{\mathrm{R}}^{\nu} s(x) = \frac{\partial^{\nu} s(x)}{\partial |x|^{\nu}} = -\frac{1}{2\cos\left(\frac{\pi\nu}{2}\right)}\left(\frac{\partial^{\nu}}{\partial x^{\nu}} + \frac{\partial^{\nu}}{\partial(-x)^{\nu}}\right) s(x), \tag{3}$$

where $\nu \neq 1$, $p - 1 < \nu \leq p \leq 2$ for $p \in N$ and

$$\frac{\partial^{\nu} s(x)}{\partial x^{\nu}} = \frac{1}{\Gamma(p-\nu)} \frac{\partial^{p}}{\partial x^{p}} \int_{-\infty}^{x} \frac{s(\xi)d\xi}{(x-\xi)^{\nu+1-p}}, \tag{4}$$

$$\frac{\partial^{\nu} s(x)}{\partial(-x)^{\nu}} = \frac{1}{\Gamma(\nu-p)} \frac{\partial^{p}}{\partial x^{p}} \int_{x}^{+\infty} \frac{s(\xi)d\xi}{(\xi-x)^{\nu+1-p}}. \tag{5}$$

In the algorithm of the FCD operator, discretization with a step h=1 is used to calculate the Riesz derivative of order ν ($0 < \nu < 1$)

$$\frac{\partial^{\nu} s(x)}{\partial |x|^{\nu}} = -\frac{1}{2\cos\left(\frac{\pi\nu}{2}\right)}\left(\frac{\partial^{\nu}}{\partial x^{\nu}} + \frac{\partial^{\nu}}{\partial(-x)^{\nu}}\right)$$
$$\approx -\frac{1}{2\cos\left(\frac{\pi\nu}{2}\right)h^{\nu}}\left[\sum_{k=0}^{+\infty}\omega_k s(x-kh) + \sum_{k=-\infty}^{0}\omega_k s(x-kh)\right], \quad (6)$$

where

$$\omega_0 = -\frac{\Gamma(1-\nu/2)}{\nu\Gamma(1+\nu/2)\Gamma(-\nu)},$$
$$\omega_k \frac{(-1)^{k+1}\Gamma(\nu/2)\Gamma(1-\nu/2)}{\Gamma(\nu/2-k+1)\Gamma(\nu/2+k+1)\Gamma(-\nu)}, \quad (7)$$

for $k = \pm1,\pm2,\ldots$.

The algorithm for applying the FCD operator, based on the Riesz derivative, to the image requires three input parameters:

- s —matrix of the image to which the operator should be applied
- ν —the Riesz derivative order ($0 < \nu < 1$)
- m —positive integer specifying the size of the window (since we need the mask to always have a center, the size of the window will be $(2m + 1)$ x $(2m + 1)$)

The first step of the algorithm is to calculate the Cs coefficients using the formula resulting from discretization (6), (7):

$$C_{s_k} = -\frac{1}{2\cos\left(\frac{\pi\nu}{2}\right)h^{\nu}}\omega_k, \quad (8)$$

for $k = 0,1,2,\ldots,2m.$

Having these coefficients, it is possible to construct 8 W_l masks which are shown in Fig. 1 and Fig. 2

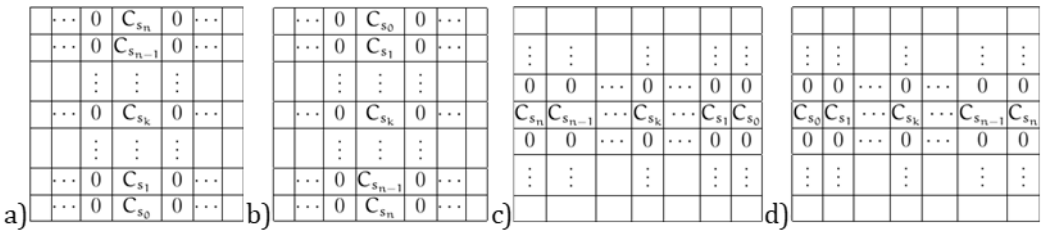


Fig. 1. Masks for four of the eight directions a) W_1 is the left-hand derivative with respect to x; b) W_2 is the right-hand derivative with respect to x; c) W_3 is the left-hand derivative with respect to y; d) W_4 is the right-hand derivative with respect to y [7]

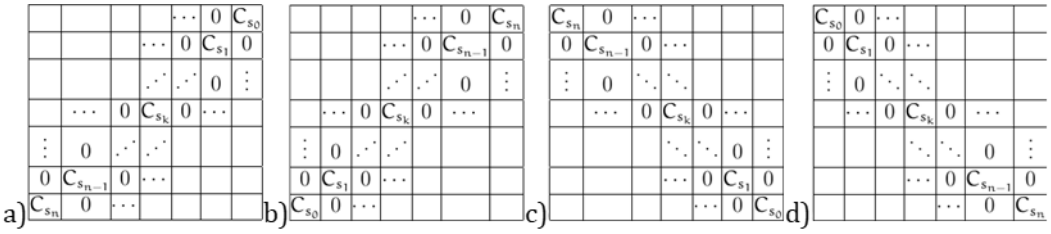


Fig. 2. Masks for the other four of the eight directions a) W_5 — diagonal to the left downwards; b) W_6 — diagonal to the right upwards; c) W_7 — diagonal to the left upwards; d) W_8 — diagonal to the right upwards [7]

These masks are applied to the image using a convolution operation, centered at the point C_{S_0}, that is, the coefficient C_{S_0} corresponds to the current pixel. To do this, for each pixel *s*(*x*, *y*), where *x* is a row in the matrix *s*, and *y* is a column, we calculate:

$$s_l(x,y) = \sum_{i=M_l}^{N_l} \sum_{j=P_l}^{Q_l} W_l(i,j)s(x+i, y+j), \tag{9}$$

for $l = 1,2,3,4$;

$$s_l(x,y) = \sum_{i=M_l}^{N_l} \sum_{j=P_l}^{Q_l} 2^{-\nu/2} W_l(i,j)s(x+i, y+j) + (1-2^{-\nu/2})W_l(0,0)s(x,y), \tag{10}$$

for $l = 5,6,7,8$; where

$$M_1 = -2m \quad N_1 = 0 \quad P_1 = -m \quad Q_1 = m$$

$$M_2 = 0 \quad N_2 = 2m \quad P_2 = -2m \quad Q_2 = m$$

$$M_3 = -m \quad N_3 = m \quad P_3 = -2m \quad Q_3 = 0$$

$$M_4 = -m \quad N_4 = m \quad P_4 = 0 \quad Q_4 = 2m$$

$$M_5 = 0 \quad N_5 = 2m \quad P_5 = -2 \quad Q_5 = 0$$

$$M_6 = -2m \quad N_6 = 0 \quad P_6 = 0 \quad Q_6 = 2m$$

$$M_7 = -2m \quad N_7 = 0 \quad P_7 = -2m \quad Q_7 = 0$$

$$M_8 = 0 \quad N_8 = 2m \quad P_8 = 0 \quad Q_8 = 2m$$

After that, we sum up the results of applying all the masks to get the final value:

$$s(x,y) = \frac{\sum_{l=1}^{8} s_l(x,y)}{4\left[\sum_{k=0}^{n} C_{sk} + \sum_{k=1}^{n} 2^{-\nu/2} C_{s_k} + C_{s_0}\right]} \tag{11}$$

As a result, we obtain an image with enhanced texture [7].

Data preparation

In the selected dataset, CT-scans are presented as 12-bit images, each pixel of which contains a value in the Hounsfield units (denoted as HU from Hounsfield Unit). The Hounsfield unit is a linear transformation of the absorption coefficient μ of body tissue measured using a tomograph:

$$HU = 1000 \times \frac{\mu - \mu_B}{\mu_B - \mu_\Pi}, \tag{12}$$

where μ_B — water absorption coefficient, μ_Π — air absorption coefficient.

The Hounsfield scale establishes the correspondence between these values and materials. Thus, air has a minimum value of 1,000 HU under normal conditions (temperature 0°C and pressure 105 Pa), distilled water corresponds to 0 HU, for some bones the value can reach 2,000 HU, and metals correspond to values greater than 3,000 HU [13, 14].

To analyze CT-scans, a window of a narrower range is selected from the image, it covers only part of the Haunsfield scale, which corresponds to the tissues of interest. The window is specified by the position of the center on the Haunsfield scale (WL is short for window level) and width (WW – short for window width). The minimum and maximum values (I_{min} and I_{max}, respectively) falling within a given

window can be found using the following formulas:

$$I_{\min} = WL - (WW \div 2), \tag{13}$$

$$I_{\max} = WL + (WW \div 2). \tag{14}$$

In the resulting image, all values less than I_{min} will be black, and values greater than I_{max} will be white (Fig. 3).

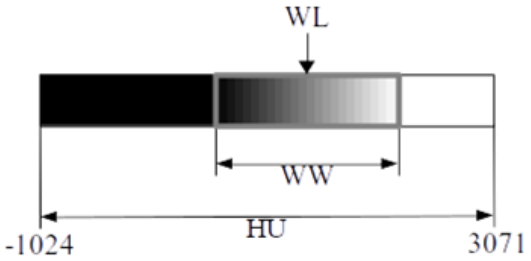


Fig. 3. Schematic representation of the window on the Hounsfield scale

By changing the position of the center of the window, we can concentrate on the examination of the tissues that interest us. In this case, the wider the window, the more types of tissue will be covered, but at the cost of lower contrast, and vice versa, a narrower window will allow you to obtain a contrast image, but will discard a lot of information that does not fall within its confines. Therefore, it is important to choose the optimal parameters of the window.

As in the original article, the images were divided into separate layers to which a window was applied (WL: 40, WW: 120) to highlight the brain tissue (Fig. 4). The resulting images were saved in 8-bit PNG format in two directories: scans — in image, and corresponding masks with the same file name — in label. The dataset was divided into training and test sets, so that 80% of randomly selected images were allocated for training and the rest for testing.

a) b) c)

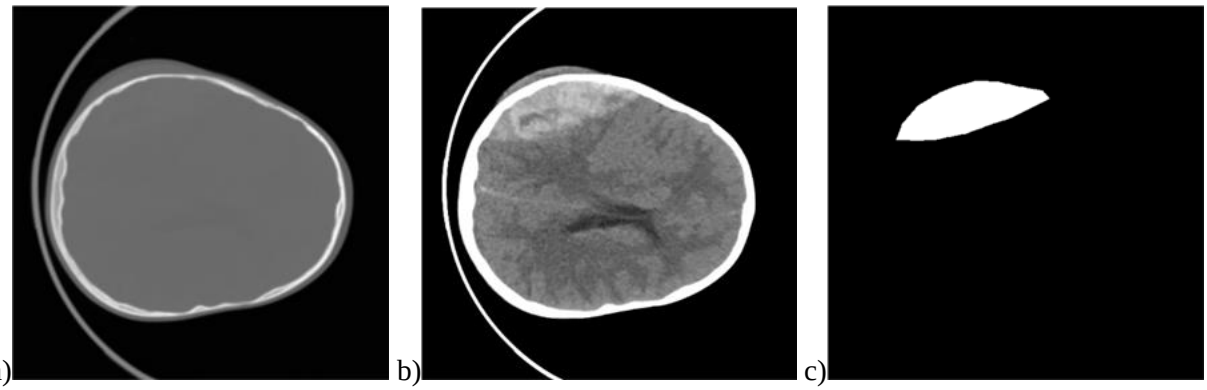


Fig. 4. Example of an image from the dataset a) before processing; b) after applying a window (WL: 40, WW: 120); c) mask highlighting the area with hemorrhage

Since, in addition to developing a model capable of segmenting intracranial hemorrhages, the goal of this work is also the study of masks based on fractional derivatives, it was decided to simplify the first task. Therefore, the network is only required to segment any type of hemorrhage in the image. That is, the usual problem of binary segmentation, where each pixel needs to be assigned a label of one or another class. Moreover, only images where hemorrhage is definitely present will be considered. As mentioned earlier, there are only 318 such images in the dataset. This limitation is not too artificial, since given the accuracy with which whole CT- scans can be classified today, it seems quite possible that in the future a two-model ensemble will be created where one network classifies the image, and if it belongs to the class with existing hemorrhage, it is transmitted to the input of the other network to pinpoint the location of this damage.

Technologies used

The studies were conducted using the Python programming language. The OpenCV library of computer vision algorithms was used for image preprocessing

The PyTorch framework [15, 16] was used to work with neural networks, as it provides a convenient and, at the same time, flexible interface for parallel training of ANNs on GPU using the CUDA technology.

In addition, the advantage of the PyTorch is an active community. Because of this, there are many additional packages to simplify work with this framework. One of these is MedicalTorch [17] — a small library containing implementations of popular ANN architectures for working with medical data and some auxiliary functions for training them. This library was also used in this work.

ANN training requires powerful GPUs, access to which is provided by many cloud services. The Google Colab [18] cloud service was used in this paper, because it is free, and the interactive program execution mode it offers is convenient for conducting research. The disadvantage of the free version of this service is that it is not possible to choose exactly which GPU will be provided, since it is automatically selected by the load balancer. At the time of the research, an Nvidia Tesla T4 with 16 GB of video memory was used.

U-Net training procedure based on PyTorch framework

First, the ICHDataset class was implemented, which is derived from the standard Dataset class of the PyTorch framework. Two objects, train_dataset and test_dataset were created to represent the training and test subsets, respectively.

In this study, it was decided to transfer whole images (512x512) to the input of the network, without dividing them into parts (160x160), as described in paragraph 2.2. To reduce memory usage, the batch size was reduced to 1. To increase the diversity of data, a transformation in the form of a random affine transformation was added to the training part. Code responsible for data loaders creation is shown below:

```
train_dataset.dataset.set_transform(torchvision.transforms.Compose([
    mt_transforms.ToPIL(),
    mt_transforms.RandomAffine(degrees=20, translate=(0.1, 0.1), scale=(0.8, 1.2)),
    mt_transforms.ToTensor()
]))
train_loader = DataLoader(train_dataset, batch_size=1, shuffle=True, num_workers=4)
test_loader = DataLoader(test_dataset, batch_size=1, shuffle=True, num_workers=4)
```

The standard U-Net architecture from the MedicalTorch library was used as a neural network for segmentation. The dropout and batch normalization parameters were set to 0.4 and 0.1, respectively.

By calling .to("cuda") for the created model, we indicate that we need to transfer this object to the GPU memory and then for all further operations with it, PyTorch will use the corresponding implementations parallelized using CUDA [16, 19].

Network weights were optimized using the Adam optimizer with a learning factor of 0.001. The Dice coefficient with a negative sign is used as the loss function. Because of this, its values lie in the range from 0 to -1, and during training, the value of the function decreases.

Below is a code fragment responsible for creating the network objects and the optimizer:

```
model = mt_models.Unet(drop_rate=0.4, bn_momentum=0.1).float().to("cuda")
loss_fn = mt_losses.dice_loss
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3, weight_decay=1e-8)
```

The network was directly trained in the main training cycle for 100 epochs. The main part of this training cycle, except for the code for visualizing the results and logging, is shown below:

```
for epoch in range(100):
    loop = tqdm.notebook.tqdm(dataloader)
    for i, batch in enumerate(loop, 0):
```

```
        # send inputs to selected device
        inputs = batch["input"].to("cuda")
        labels = batch["gt"].to("cuda")

        # zero the parameter gradients
        optimizer.zero_grad()

        # forward + backward + optimize
        outputs = model(inputs)
        loss = loss_fn(outputs, labels)
        loss.backward()
        optimizer.step()
```

Since the model is in the GPU memory, at each iteration it is necessary to transfer part of the training data to the same device using the .to("cuda") method. After that, the input image is transmitted to the network, the value of the loss function is calculated, and the backpropagation of the error and the optimization of the network parameters take place.

Implementation of masks based on the Riesz fractional order derivatives

The algorithm described in paragraph 3.3 was implemented in the Python programming language as a function that expects an image, the degree of the Riesz derivative and the window size as the input, and returns the image with an enhanced texture as a result of applying the FCD operator.

Inside the function, using the NumPy module, 8 W_l matrices are built, the coefficients of which are calculated according to formula (8). Then the convolution function from the OpenCV library is used to effectively apply each mask separately to the image according to formulas (9) and (10). The final result is obtained by adding the results of applying individual masks according to formula (11).

To study the impact of the Riesz derivative-based operator on the quality of segmentation, a second version of the dataset was prepared, in which this operator with a derivative order of v = 0.5 and a mask size of 3x3 was previously applied to all images.

Results and Discussion

Analysis of the results of solving the segmentation task

Let us consider how well the trained network copes with the assigned applied task — segmentation of intracranial hemorrhages based on computed tomography data. For this purpose, 20% of previously separated images, which were not used to train the network, are used.

During training, after each epoch, the ANN was applied to segment a random image from the test sample, in order to see how the segmentation quality changes over time. Figs. Figs. 5-8 show some of these images, where one can see the input image on the left, the expected result in the middle, and the network output on the right.

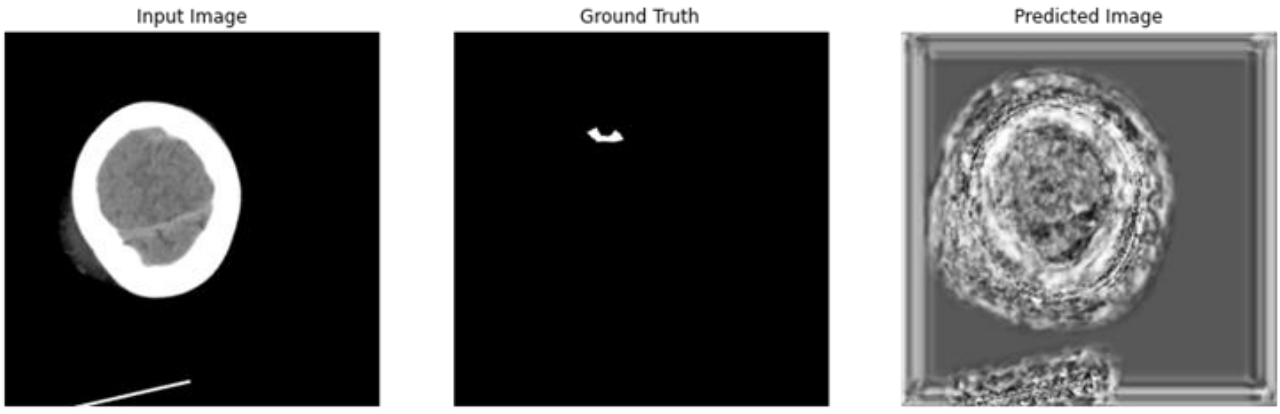


Fig. 5. ANN output before training

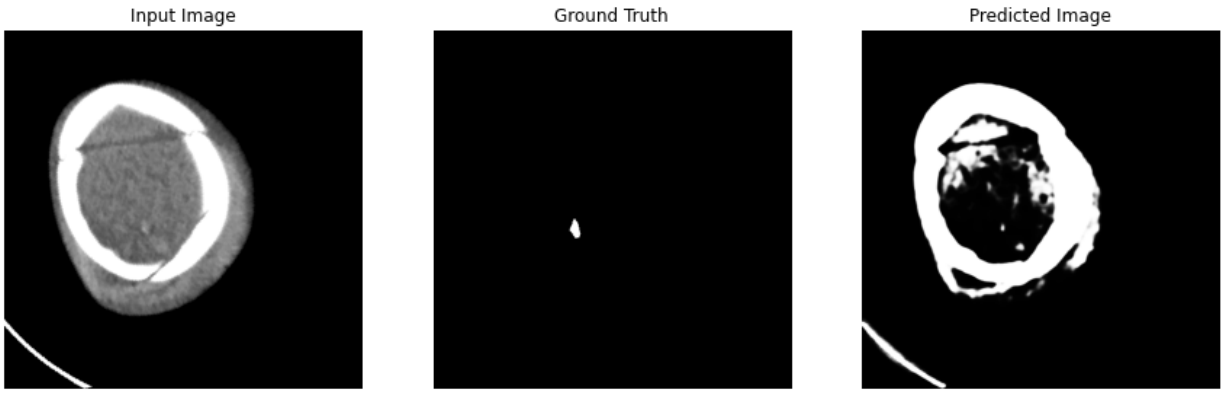


Fig. 6. ANN output after epoch 1

First, one can see the result of image processing by a network with random parameter values. Already during the first epoch, the ANN learned to binarize the image, but so far it does not highlight the areas that are needed. More or less successfully highlighted hemorrhage was observed after the ninth epoch. However, it is also quite pronounced in the input image. After the hundredth epoch, i.e. at the end of the training, you can see that this model can also detect very weakly expressed hemorrhages, which at first glance are not noticeable.

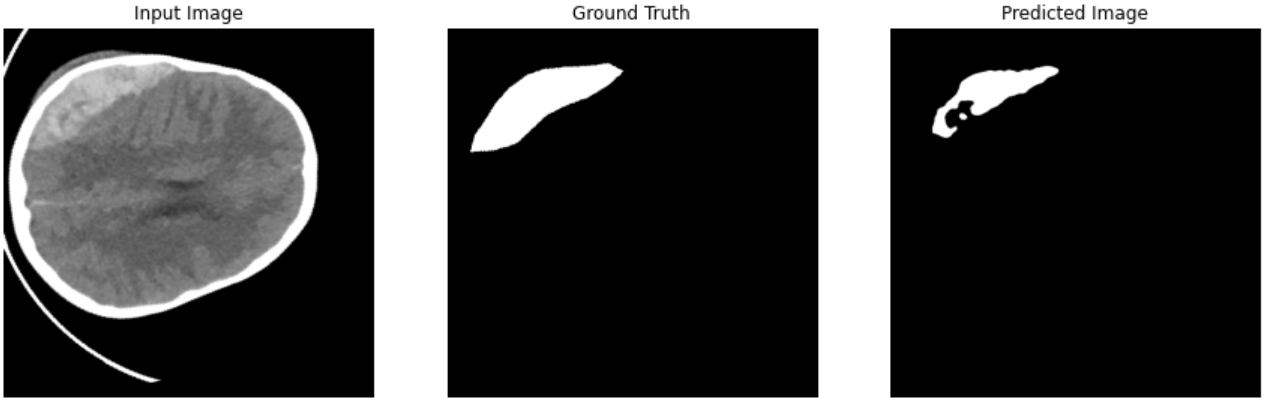


Fig. 7. ANN output after epoch 9

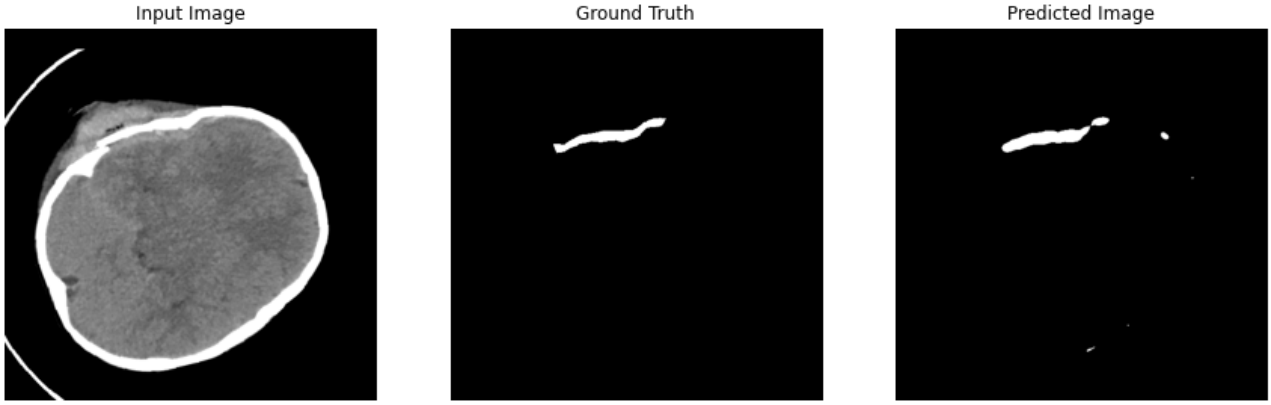


Fig. 8. ANN output after epoch 100

Table 3 shows the results of testing a network trained during 100 epochs. To compare the results, the Table is presented in the same format as Table 1.

Table 3

Assessment of segmentation accuracy

Test dataset statistics	Jaccard coefficient	Dice coefficient
Min	0.000	0.000
Max	0.780	0.876
STD	0.289	0.333
Average	0.353	0.452

The maximum and average values of both metrics for the investigated network are greater than those obtained in [3]. But it should not be forgotten that the simplified task of segmentation of hemorrhages from images where they are guaranteed to be present was considered here. However, as already mentioned, there are ANN models that can accurately screen out images of healthy patients. And the fact that the better results were achieved for segmentation only among images with affected areas suggests the advisability of using a combined model, in which one part will be responsible for classification and the other for segmentation. The study of such a model may become one of the directions for further expansion of the topic presented in this paper.

Analysis of image quality improvement using the Riesz derivative

Now let us consider the results of applying the Riesz derivative-based operator to our dataset. Figure 9 shows the results of applying the FCD operator with different window sizes based on the Riesz derivative of order $v = 0.5$ to an image from the dataset. As can be seen, for the smallest possible window, the texture of the brain becomes more visible. As the window size increases, the contrast increases, but with it, the effect of noise increases. This conclusion also corresponds to the results obtained by the authors of the FCD operator (Table 2). Therefore, for further research it is recommended to use small masks measuring 3x3 or 5x5.

In addition, different values of the order of the derivative were also checked. Figure 10 shows the results of applying the 7x7 FCD operator based on the Riesz derivative of orders 0.1, 0.5, and 0.9. Visually, the result for the derivative of order 0.5 looks the best, since for order $v = 0.1$ the texture enhancements are almost not noticeable, and for $v = 0.9$ noise begins to appear more strongly.

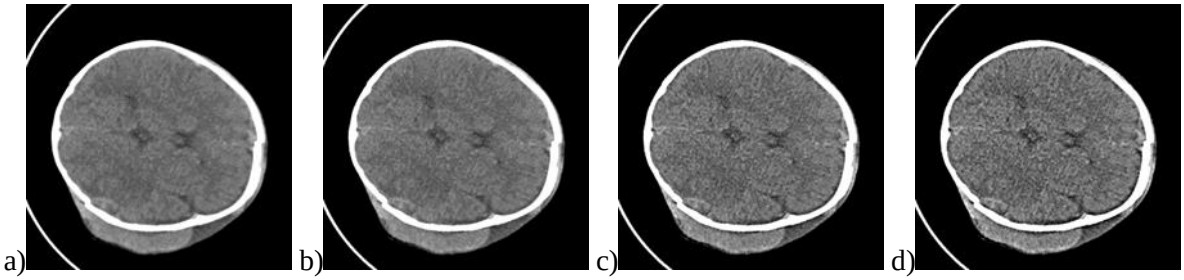


Fig. 9. Results of applying the FCD operator based on the Riesz derivative of order $v = 0.5$ with different mask sizes: a) input image; b) 3x3 mask; c) 7x7 mask; d) 11x11 mask

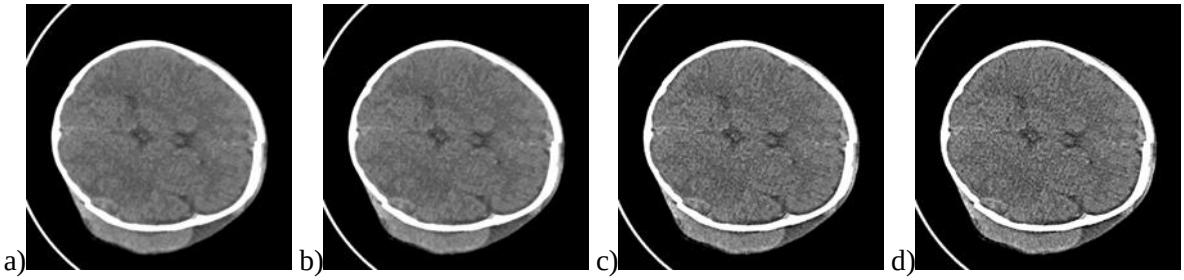


Fig. 10. Results of applying the 7x7 FCD operator based on the Riesz derivative of different orders: a) input image; b) $v = 0.1$; c) $v = 0.5$; d) $v = 0.9$

To quantitatively analyze the improvement in image texture quality, a gray-level co-occurrence matrix (GLCM) approach was used, as proposed in [8].

GLCM is a matrix, each element c_{ij} of which corresponds to the number of times a pixel with brightness j is at a certain distance relative to the pixel with brightness i in the image under study [20]. For texture analysis, several such matrices are usually constructed, each of which corresponds to a certain relative arrangement of pixels.

In our case, 4 matrixes were constructed:

• for vertical — takes into account how many times the pixel with brightness j is one pixel above or below the pixel with brightness i;

• for horizontal — takes into account how many times the pixel with brightness j is one pixel to the left or right of the pixel with brightness i;

• for the main diagonal;

• for the secondary diagonal.

Since the images under consideration can contain values in the range [0, 255], matrices of size 256x256 are constructed. Also, since all-black pixel pairs ($c_{0,0}$) in the studied dataset mainly correspond to the background, and all-white pixel pairs ($c_{255,255}$) correspond to bones, they were ignored in the texture study. For this, the corresponding elements of the matrix were assigned the value 0.

After that, the obtained 4 matrices were averaged into one matrix, for which the following characteristics were calculated: contrast (15), correlation (16), energy (17) and homogeneity (18) [20].

$$Contrast = \sum_i \sum_j (i-j)^2 c_{i,j}, \tag{15}$$

$$Correlation = \sum_i \sum_j \frac{(i-\mu_i)(j-\mu_j)c_{i,j}}{\sigma_i \sigma_j}, \tag{16}$$

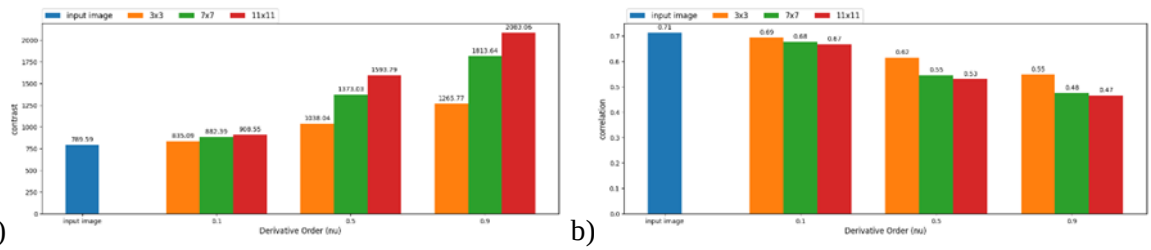
where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums of the GLCM matrix C

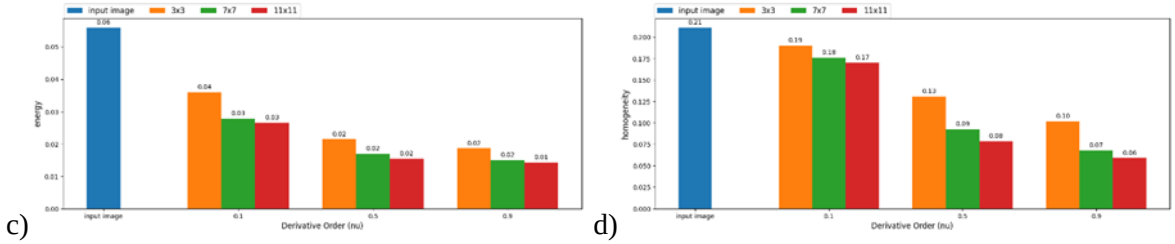
$$Energy = \sum_i \sum_j c_{i,j}^2, \tag{17}$$

$$Homogeneity = \sum_i \sum_j \frac{c_{i,j}}{1+|i-j|}. \tag{18}$$

These characteristics were calculated both for the original image and for the images obtained as a result of using the FCD operator with different parameters, namely, 3 values of the order v of the Riesz derivative (0.1, 0.5, 0.9) and 3 values of the mask size (3x3, 7x7, 11x11) were considered. Such calculations were performed for all images of the dataset and the results were averaged. The resulting diagrams are shown in Fig. 11.

a) b)





c) d)

Fig. 11. Graphs of the main characteristics of GLCM for various parameters of the FCD operator for the studied dataset: a) contrast; b) correlation; c) energy; d) homogeneity

From the results obtained, as the order v of the Riesz derivative and the window size increase, the contrast increases while the correlation, energy, and homogeneity decrease. Moreover, at small values of v, the effect of window size on contrast, correlation and homogeneity is quite insignificant. In general, we can say that these results correlate with the results obtained in [8] for other fractional operators.

An increase in contrast indicates an increase in the difference between adjacent pixels, which may indicate an increase in image clarity, while a decrease in homogeneity indicates a decrease in the number of monochromatic regions. A decrease in correlation and energy indicates an increase in the complexity of the image texture. In general, medical images are not characterized by simple and predictable textures, so the energy values may be quite low, but this may also indicate increased noise. Therefore, for the subsequent experiment, it was decided to use the FCD operator with the smallest mask size of 3x3 and the order of the Riesz derivative $v = 0.5$.

The influence of the FCD operator on the quality of segmentation

In addition to analyzing the improvement of image texture, we are interested in whether the use of the FCD operator will help in solving the application task of intracranial hemorrhage segmentation, which we considered earlier. In order to verify this, the entire training and test data sets were processed by the FCD operator with a window size of 3x3 based on the Riesz derivative of order $v = 0.5$ and the processed data was applied to the U-Net ANN training, just as described in paragraph 3.3.

The obtained results are shown in Table 4. Comparing these results with those shown in Table 3, we can conclude that, indeed, the use of an operator based on fractional derivatives, in addition to the visual improvement of the texture, also improves the quality of segmentation. The average value of the Jaccard coefficient for the test dataset increased from 0.353 to 0.398, and that of the Dice coefficient - from 0.452 to 0.507.

Table 4

Assessment of segmentation accuracy for data processed by the FCD operator

Test dataset statistics	Jaccard coefficient	Dice coefficient
Min	0.000	0.000
Max	0.854	0.921
STD	0.276	0.317
Average	0.398	0.507

Conclusions

This paper analyzes the current developments in the fields of ICH segmentation and image quality improvement based on fractional order fractal operators.

As a result of the literature review in these areas, it was decided to use the U-Net architecture to solve the problem of segmentation of intracranial hemorrhages based on CT data. Also considered is the impact of processing this data, using an operator based on the Riesz derivative of fractional order, on the quality of segmentation.

Using the Python programming language, software was implemented with which the quality of segmentation of intracranial hemorrhages using the U-Net network, as well as the influence of the FCD operator, was investigated.

The main results obtained in this paper are as follows:

• The parallel training algorithm for U-Net was implemented utilizing the PyTorch framework based on CUDA technology. The trained U-Net network showed good results based on the Jaccard and Dice metrics in solving the problem of segmenting intracranial hemorrhages. Of course, it is still too early to talk about the automatic diagnosis of real patients, but it was shown that this direction has potential and with certain further improvements, it is possible to create a system that, based on the ANN, will help doctors pay attention to problem areas on a CT-scan.

• The influence of the FCD operator on the selected dataset was studied. The influence of the order of the Riesz derivative ($0 < v < 1$) and the window size on the texture quality of the processed image was investigated visually and quantitatively using GLCM. The Riesz derivative of the order $v = 0.5$ with a small window of size 3x3 or 5x5 was selected as the optimal parameters.

• It has been demonstrated that the use of the FCD operator of order 0.5 with a 3x3 window improves the quality of hemorrhage segmentation by approximately 12%.

References

[1] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, P. Warier, Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. Lancet 01.12 (2018): 2388-2396. https://doi.org/10.1016/S0140-6736(18)31645-3

[2] Radiological Society of North America, RSNA Intracranial Hemorrhage Detection, 2019. URL: https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection/overview.

[3] M. D. Hssayeni, M.S. Croock, A. D. Salman, H. F. Al-khafaji, Z. A. Yahya, B. Ghoraani, Intracranial Hemorrhage Segmentation Using a Deep Convolutional Model, *Data* 5 (2020): 14-32. https://doi.org/10.3390/data5010014

[4] M. D. Hssayeni, Computed Tomography Images for Intracranial Hemorrhage Detection and Segmentation, version 1.3.1, PhysioNet, 2020. doi: 10.13026/4nae-zg36.

[5] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, 2000. URL: https://physionet.org. https://doi.org/10.1161/01.CIR.101.23.e215

[6] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical image computing and computer-assisted intervention, Springer, Cham, 2015, pp. 234-241. doi: 10.1007/978-3-319-24574-4_28.

[7] Q. Yu, F. Liu, I. Turner, K. Burrage, V. Vegh, The use of a Riesz fractional differential-based approach for texture enhancement in image processing, ANZIAM Journal 54 (2012): 590-607. https://doi.org/10.1007/978-3-319-24574-4_28

[8] J. Hamid, I. Rabha, Texture Enhancement for Medical Images Based on Fractional Differential Masks, Discrete Dynamics in Nature and Society, 28.03 (2013). https://doi.org/10.1155/2013/618536

[9] Y.-F. Pu, J.-L. Zhou, X. Yuan, Fractional Differential Mask: A Fractional Differential-Based Approach for Multiscale Texture Enhancement, IEEE Transactions on Image Processing 19 (2010): 491-511. https://doi.org/10.1109/TIP.2009.2035980

[10] Ya. Sokolovskyy, M. Levkovych and I. Sokolovskyy, The study of heat transfer and stress-strain state of a material, taking into account its fractal structure. Mathematical Modeling and Computing. 7(2), 2020, pp. 400–409. https://doi.org/10.23939/mmc2020.02.400

[11] Ya. Sokolovskyy, M. Levkovych, O. Mokrytska, and Ya. Kaplunskyy, Mathematical models of biophysical processes taking into account memory effects and self-similarity, CEUR Workshop Proceedings, 2018, vol. 2255, pp. 215–228.

[12] F. Liu, P. Zhuang, V. Anh, I. Turner, and K. Burrage, Stability and convergence of the difference methods for the space–time fractional advection–diffusion equation, Applied Mathematics and Computation, vol. 191, issue 1 (2007): 12-20. https://doi.org/10.1016/j.amc.2006.08.162

[13] M. Nadrljanski, A. Campos, R. Chieng, et al. Computed tomography. Reference article, Radiopaedia.org, 2024. https://doi.org/10.53347/rID-9027

[14] K. Greenway, R. Sharma, D. Vargas Carvajal, et al. Hounsfield unit. Reference article, Radiopaedia.org, 2024. https://doi.org/10.53347/rID-38181

[15] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: Proceedings of the 31st Conference on Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA, 2017. URL: https://openreview.net/forum?id=BJJsrmfCZ.

[16] PyTorch Contributors, PyTorch Documentation, 2023. URL: https://pytorch.org/docs/stable/index.html.

[17] Christian S. Perone, cclauss, Elvis Saravia, Pedro Lemos Ballesteri, MohitTare, "perone/medicaltorch: Release v0.2", Zenodo, 2018. doi: 10.5281/zenodo.1495335.

[18] Colaboratory, Frequently Asked Questions, 2023. URL: https://research.google.com/colaboratory/faq.html.

[19] NVIDIA Corporation & Affiliates, CUDA Toolkit Documentation 12.2, 2023. URL: https://docs.nvidia.com/cuda/archive/12.2.0/.

[20] L. G. Shapiro, G. C. Stockman, Co-occurrence Matrices and Features, in: Computer Vision, 1st. ed., Pearson, 2001, pp. 240-243.

Денис Манохін[1], Ярослав Соколовський[2]

[1] Кафедра інформаційних систем, Львівський національний університет ім. Івана Франка, вул. Університетська 1, Львів, Україна, E-mail: Denys.Manokhin@lnu.edu.ua, ORCID 0000-0002-8590-7626

[2] Кафедра систем автоматизованого проектування, Національний університет "Львівська політехніка", вул. С. Бандери 12, Львів, Україна, E-mail: yaroslav.i.sokolovskyi@lpnu.ua, ORCID 0000-0003-4866-2575

СЕГМЕНТАЦІЯ ВНУТРІШНЬОЧЕРЕПНОГО КРОВОВИЛИВУ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ ТА ПОКРАЩЕННЯ ТЕКСТУРИ НА ОСНОВІ ДРОБОВОГО ОПЕРАТОРА РІСА

Отримано: березень 04, 2024 / Переглянуто: квітень 01, 2024 / Прийнято: квітень 05, 2024

Анотація. У статті досліджується застосування архітектури U-Net для сегментації внутрішньочерепних крововиливів, зосереджуючись на підвищенні точності сегментації шляхом включення методів покращення текстури на основі похідних дробового порядку Ріса. Дослідження починається з проведення огляду суміжних робіт у галузі сегментації комп'ютерної томографії (КТ). На цьому етапі також вибирається відповідний набір даних. Спочатку він використовувався для навчання U-Net, однієї з широко поширених моделей глибокого навчання в області сегментації медичних зображень. Навчання здійснюється за паралельним алгоритмом на основі технології CUDA. Отримані результати порівнюють із встановленою базовою моделлю для цього набору даних, оцінюючи точність сегментації за допомогою коефіцієнтів Жаккара та Дайса. Згодом досліджується техніка покращення текстури, заснована на дробових похідних Ріса, і застосована до зображень комп'ютерної томографії з вибраного набору даних. Ця техніка спрямована на захоплення дрібніших деталей і тонких текстур, які можуть сприяти підвищенню точності сегментації. Потім модель U-Net перенавчається та перевіряється на зображеннях із покращеною текстурою, а результати експерименту аналізуються. Дослідження виявило помітне підвищення точності, обгрунтованого за допомогою коефіцієнтів Жаккара та Дайса. Це демонструє потенціал запропонованої методики покращення текстури для уточнення сегментації внутрішньочерепного крововиливу.

Ключові слова: похідна дробового порядку, глибоке навчання, Python, структура PyTorch, технологія CUDA, покращення текстур, сегментація медичних зображень