UDC 004.93

RCF-ST: RICHER CONVOLUTIONAL FEATURES NETWORK WITH STRUCTURAL TUNING FOR THE EDGE DETECTION ON NATURAL IMAGES

Polyakova M. V. – Dr. Sc., Associate Professor, Professor of the Department of Applied Mathematics and Information Technologies, National University "Odessa Polytechnic", Odessa, Ukraine.

ABSTRACT

Context. The problem of automating of the edge detection on natural images in intelligent systems is considered. The subject of the research is the deep learning convolutional neural networks for edge detection on natural images.

Objective. The objective of the research is to improve the edge detection performance of natural images by structural tuning the richer convolutional features network architecture.

Method. In general, the edge detection performance is influenced by a neural network architecture. To automate the design of the network structure in the paper a structural tuning of a neural network is applied. Computational costs of a structural tuning are incomparably less compared with neural architecture search, but a higher qualification of the researcher is required, and the resulting solution will be suboptimal. In this research it is successively applied first a destructive approach and then a constructive approach to structural tuning of the based architecture of the RCF neural network. The constructive approach starts with a simple architecture network. Hidden layers, nodes, and connections are added to expand the network. The destructive approach starts with a complex architecture network. Hidden layers, nodes, and connections are then deleted to contract the network. The structural tuning of the richer convolutional features network includes: (1) reducing the number of convolutional layers; (2) reducing the number of convolutions in convolutional layers; (3) removing at each stage the sigmoid activation function with subsequent calculation of the loss function; (4) addition of the batch normalization layers after convolutional layers; (5) including the ReLU activation functions after the added batch normalization layers. The obtained neural network is named RCF-ST. The initial color images were scaled to the specified size and then inputted in the neural network. The advisability of each of the proposed stages of network structural tuning was reseached by estimating the edge detection performance using the confusion matrix elements and Figure of Merit. The advisability of a structural tuning of the neural network as a whole was estimated by comparing it with methods known from the literature using the Optimal Dataset Scale and Optimal Image Scale.

Results. The proposed convolutional neural network has been implemented in software and researched for solving the problem of edge detection on natural images. The structural tuning technique may be used for informed design of the neural network architectures for other artificial intelligence problems.

Conclusions. The obtained RCF-ST network allows to improve the performance of edge detection on natural images. RCF-ST network is characterized by a significantly fewer parameters compared to the RCF network, which makes it possible to reduce the resource consumption of the network. Besides, RCF-ST network ensures the enhancing of the robustness of edge detection on texture background.

KEYWORDS: natural image, edge detection, convolutional network, richer convolutional features, structural tuning, batch normalization.

ABBREVIATIONS

CNN is a convolutional neural network;
RCF is a Richer convolutional features;
HED is a Holistically-nested edge detection;
LPCB is a Learning to predict crisp boundaries;
BDCN is a Bi-directional cascade network;
DexiNed is a Dense extreme inception network;
DSCD is a Deep structural contour detection;
PiDiNet is a Pixel difference network;
ReLU is a rectified linear unit;
RCF-ST is a Richer convolutional features with structural tuning;
BSDS500 is a Berkeley Segmentation Dataset with 500 images;
FOM is a Figure of Merit;
ODS is an Optimal Dataset Scale;
OIS is an Optimal Image Scale.

NOMENCLATURE

n is a number of rows of the image;
m is a number of columns of the image;
(x,y) are coordinates of the image pixel;
$\mathbf{I}(x,y)$ is a vector function representing an image by color components;
$I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the functions of intensity of the red, green, blue color components respectively;
$struct_{RCF}$ is an architecture of the RCF network;
$param_{RCF}$ is a set of parameters of the RCF network;
W_{RCF} is a subset of RCF network layer weights;
B_{RCF} is a subset of RCF network bias values;
x_{ni} is a nth normalized output of the ith network layer;
γ_i is a compression of the x_{ni};
β_i is a shift of the x_{ni};
y_{ni} is a transformed with γ_i and β_i output of the network layer;
TP is apercentage of image background pixels that are correctly labeled as background;
TN is a percentage of image edge pixels that are correctly labeled as edge.
F_β is a F_β-score;

DOI 10.15588/1607-3274-2023-4-12

β is a F_β-score constant from 0 to infinity;
Pr is a precision;
Rc is a recall.

INTRODUCTION

The problem of edge detection on images remains relevant in the development of intelligence systems for a number of applications. These applications include technical and medical diagnostics, image search in databases and the Internet, face recognition, non-destructive testing, process control. The selection of the object edge detector for the implementing an intelligent system is determined primarily by the properties of the processed images, the noise level, as well as the requirements for the edge detection performance. So, in systems for monitoring the environment, transport and infrastructure, searching for images in databases and the Internet, and others, it becomes necessary to process natural images.

Natural images are characterized by a low level of noise. Objects on such images may contain texture areas or areas of smooth color change. When detecting the object edges on natural images, one should take into account not only color differences, but also the boundaries of texture areas. Then it is necessary to establish a correspondence between color differences and the boundaries of objects on natural images, ignoring the background texture and noisy pixels [1].

The object of research is the process of edge detection on natural images in intelligent systems.

In recent years, considering the problems of thick image edge contour, inaccurate positioning, and poor detection accuracy, a variety of edge detection methods based on deep learning CNN have been proposed. With the development of technology, the CNN edge detection accuracy has been increased. However, at the same time, the depth of the networks has been deepened, leading to problems such as a very large number of parameters, training difficulties, and model complexity [1].

For the effective use of a neural network, it is necessary to design its architecture and to train the weight coefficients. Network architectures are usually selected heuristically based on the experience of the developer. When image edge detecting, networks of too simple architecture are not able to adequately model the target dependence between the pixels of the original image and the edge map. Too complex architectures of neural networks imply an excessive number of free parameters, which in the learning process are tuned not only to restore the target dependence, but also noise [2]. One way to solve this problem is the structural tuning of CNN [3, 4].

The subject of the research is the structural tuning of the convolutional neural networks for edge detection on natural images.

When processing images of real scenes, the RCF network proved to be effective for edge detection [1]. However, the quality of the results of edge detection using this network is determined by the number of processing scales and by the network architecture. The latter implies the selection of such hyperparameters as the number and size of filter kernels in a layer, the adding and removing layers, including activation functions.

The aim of the research is to improve the edge detection performance of natural images by structural tuning the RCF deep learning network architecture.

1 PROBLEM STATEMENT

The color natural image is represented as $\mathbf{I}(x,y)=(I_R(x,y), I_G(x,y), I_B(x,y)\}$, where $x=1, \ldots, n$; $y=1, \ldots, m$. Then each pixel of the image is described by three features $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ which take values from the interval [0, 1]. To detect edges on the image, each pixel of the original image must be associated with the value of the target feature. There is a label of one of two classes, specifically, 0 for boundary pixels, 1 for pixels inside homogeneous areas. The values of the target feature for the natural image should be represented as a binary image which is the result of edge detection [5].

Let an RCF network RCF=$\{struct_{RCF}, param_{RCF}\}$ was preliminarily synthesized to detect the image edges. The set $struct_{RCF}$ includes layers of the synthesized network with layer hyperparameters such as the size of the convolution kernel and the number of convolutions. The set $param_{RCF}=\{W_{RCF}, B_{RCF}\}$ [6].

The problem of structural tuning of the RCF network is as follows. It is necessary to make structural changes to the existing architecture of the RCF network $struct_{RCF}$. These changes should improve the image edge detection performance compared to the initial RCF network after training the parameters of the resulting network. At the same time, the number of parameters of the resulting network should not increase [2].

2 REVIEW OF THE LITERATURE

To solve the problem of object edge detection on natural images, the deep learning CNN have been widely used recently. In [1] such methods in terms of model structure, technical difficulties, method advantages, and backbone networks are classified into three types. These are codec-based CNN, network reconstruction-based methods, and multi-scale feature fusion-based CNN.

Edge detection methods based on codec were introduced, as they can accept input images of any size and produce output images of the same size [7–9]. Since CNNs reduce the size of an image after convolutions and pooling, their final output in fact does not correspond to every pixel in the original image. Fully convolutional networks are used to retain better low-level edge information, suppress non-edge pixels, and provide detailed edge location [7]. The encoder layers are produce feature maps with semantic information. The decoder layers are transform the low-resolution feature maps which outputted by the encoder back to the size of the input image by pixel classification [1].

Edge detection methods based on network reconstruction integrate various network modules based on deep learning [10–12]. Different modules show

DOI 10.15588/1607-3274-2023-4-12

different advantages for edge detection, so the combination of such modules through network reconstruction is an important way to improve the edge detection results [13, 14].

The edge detection methods based on multi-scale feature fusion combine features of different scales. The higher layer of the network has a larger perceptual field and a strong ability to characterize semantic information while the lower layer of the network has a smaller perceptual field but a strong ability to characterize geometric details. Then combining of the local and global information of the image improves the edge detection performance [1].

This paper is focused on the edge detection methods based on multi-scale feature fusion. The backbone networks of these methods are the HED [15] and RCF [16]. The LPCB [17], BDCN [18], DexiNed [19], DSCD [20], PiDiNet [21] and other networks are proposed based on the HED and RCF networks, as well as by combining with the architectures of other networks to improve the edge detection performance.

The HED algorithm is proposed in [15], where a fully convolutional network is used to resolve ambiguity in edge and object boundary detection. Deeply-supervised side replies were interpolated to initial image size and fused to obtain nested multi-scale features. Thus HED develops rich hierarchical representation automatically directed by deep supervision on side replies [15].

In [17] the HED network is improved to solve the problem of thick contour in edge detection. The obtained LPCB network is based on VGG16 network [22] and uses the fully convolutional network of bottom-up/top-down architecture [23]. Based on image similarity a new loss function is also proposed, which is very effective for classifying unbalanced data. The LPCB network resolves ambiguities in edge detection, and obtains accurate results without post-processing. Compared to the HED network, LPBC uses fewer parameters although the last network shows better edge detection performance.

Inspired by HED [15] and Xception [24] networks, in [19] the deep learning-based edge detector DexiNed is elaborated to generate thin edges without prior training or finetuning process. DexiNed can be regarded as two sub-networks: extremely dense initial network and up-sampling block. This network includes six encoders, and each of them outputs the corresponding feature for generating intermediate edge maps using the up-sampling block, which consists mainly of convolutional and deconvolutional layers. All edge maps generated by the up-sampling block are connected at the end of the network to produce the fusion edges.

In [16] the RCF network for accurate edge detection is designed as a fully convolutional network based on the VGG16 network [22], removing the fully connected layer and the fifth pooling layer. While RCF edge detection the network estimates multi-scale features of the image by convolutional layers which have different perceptual fields and pooling layers. Then fusing the layer level features, all the weight parameters are done by automatic learning. Thus RCF network bases on the pyramid architecture, and combines the underlying feature maps for edge detection [25].

In [18] the BDCN network is proposed to detect edges using multiscale information of images. The basic components of BDCN are ID Blocks. Each ID Block is learned by a bidirectional cascade structure, thus the output of two edge detections is passed separately to the shallow and high-level structures of the network. To enhance the features output from each layer, a Scale Enhancement Module is used. It consists of multiple parallel convolutions with different perceptual field [26], and finally outputting the result to multiple multi-scale feature fusion.

In [20] the proposed DSCD network uses a VGG16 encoder [22] to extract multi-scale and multi-level features. On top of the encoder a super-convolutional module is constructed to directly abstract the high-level features and avoid overfitting problem. The decoder is fused the high-level features and restored them to the original image size. A novel loss function based on the structural similarity of two images is proposed to minimize the distance between predicted and true values. The DSCD network better classifies the background texture and noisy pixels as compared with another codec networks, and generates clear and accurate image edges.

In [21] the elaborated PiDiNet integrates a novel pixel difference convolution into network convolutional layer. As a result this network can easily capture image gradient information conducive to edge detection, while retaining the powerful learning ability of deep CNN to extract information with semantic significance. Then the direct integration of the gradient estimation into the convolution operation results in the better robustness and edge detection accuracy.

As a result of the analysis of the literature, the following was observed. Methods of the first type have a similar encoder-decoder architecture, which has been effectively used to solve a number of applied problems. This architecture assumes a relatively small number of parameters compared to other convolutional networks. However, the inclusion of pooling layers reduces the image edge detection performance. Therefore, it is advisable to use methods of the first type when solving problems that do not require a high edge detection performance, for example, for localizing objects on images.

The methods of the second type are characterized by the use of additional modules that improve the edge detection performance after or together with the use of CNN. Difficulties arise in the development and configuration of these modules, as well as the combining of additional modules with the architecture of the basic CNN. However, with a rational choice of additional modules and architecture of the CNN, it is possible to achieve high image edge detection performance.

Methods of the third type implement the ideas of the two previous types of methods. A set of scale values is defined, which depends on the size, as well as the content

DOI 10.15588/1607-3274-2023-4-12

of the image. For each scale value, the boundaries of objects of a certain size are identified. These methods do not require additional modules to improve the quality of the contour. But there is a need to elaborate an approach to evaluate multi-scale features and to fuse the results of edge detection at different scales.

The analysis of edge detection methods based on multi-scale feature fusion showed that the directions for improving the existing basic HED and RCF architectures and their combination with other architectures are as follows. Firstly, it is the elaboration of the classifier with the best separation of pixel classes by changing the loss function, as in LPCB and DSCD. Secondly, this is the evaluation of features with the better class separation, since the result of edge detection is determined by the shape of pixel clusters and the presence of data outliers. Along the way, integration gradient estimation into the convolution identifying is offered, as in the PiDiNet network, or blocks that take into account information about the edges is added, as in BDCN and DexiNed. In this context in the paper it is proposed to use the structural tuning of the RCF network. This approach allows to select the features of natural images and a way of them fusion with the better separation of edge and background pixels.

3 MATERIALS AND METHODS

In general, the edge detection performance is influenced by a neural network architecture. A simple architecture network may not provide good performance owing to its limited information processing power. A network of complex architecture may have high implementation cost and some of its elements are redundant. At the last time neural architecture search is applied to automate the defining the network structure [27]. Although this technique yields an optimal solution, its computational cost is enormous. Therefore, a different technique is used in the paper. This is a structural tuning of a neural network. Computational costs of a structural tuning are incomparably less, but a higher qualification of the researcher is required, and the resulting solution will be suboptimal.

To tune the network structure, constructive and destructive approaches can be used [28]. The constructive approach starts with a simple architecture network. Hidden layers, nodes, and connections are added to expand the network. The destructive approach starts with a complex architecture network. Hidden layers, nodes, and connections are then deleted to contract the network [28].

In this research author successively applies first a destructive approach and then a constructive approach to structural tuning of the based architecture of the RCF neural network (Fig. 1).

Thus, as a structural tuning of the RCF network, the following is proposed: (1) reducing the number of convolutional layers; (2) reducing the number of convolutions in convolutional layers; (3) removing at each stage the sigmoid activation function with subsequent calculation of the loss function; (4) addition of the batch normalization layers after convolutional layers; (5) including the ReLU activation functions after the added batch normalization layers. Let's explain these steps in more detail.

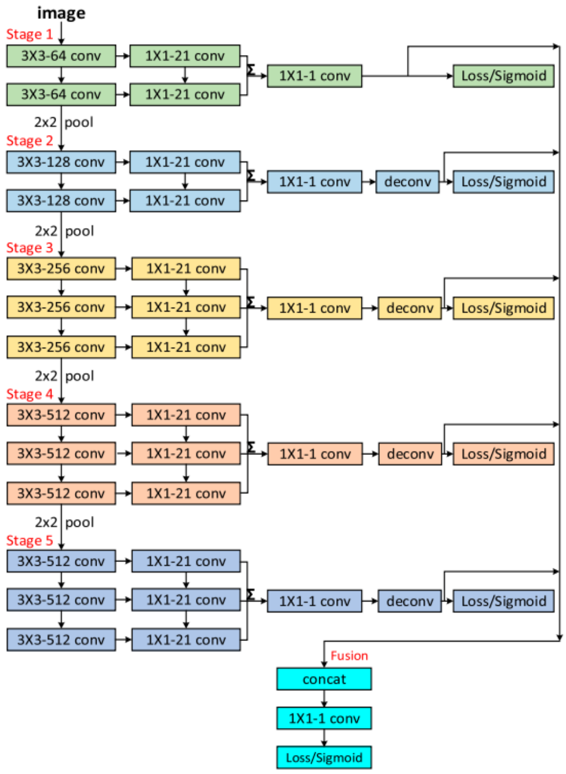

Figure 1 – The RCF architecture [16]

As a result of the destructive approach, the number of convolution layers and the number of convolutions in the remaining convolution layers were reduced (Fig. 2). Such operation is known as thinning of CNN [29].

Deep learning convolutional networks extract the image features in convolutional layers. For each such layer, the number of features to be evaluated is specified by the number of convolutions in the layer. Each feature is identified by a convolution kernel, as well as a kernel shift, and these same parameters determine the image scale on which this feature is extracted.

The redundant or poorly informative features in the resulting set reduce the rate of convergence of the network training, in particular, increases the variance of network parameter estimates. To increase the edge detection performance the noisy features can been discarded, as well as similar features. The last are processed as one feature with a large weight. The feature space dimension can be reduct by reducing the number of convolutions of CNN. This makes it possible to use a smaller training set, reduce training time, and reduce the network overfitting probability. The evaluation of features for edge detection on images influences on the separability of image classes, taking into account the fact that the number of edge pixels differs significantly from the number of background pixels. Therefore, the

DOI 10.15588/1607-3274-2023-4-12

structural tuning of the RCF network in this paper includes altering the number of convolutional layers and the number of convolutions in convolutional layers.

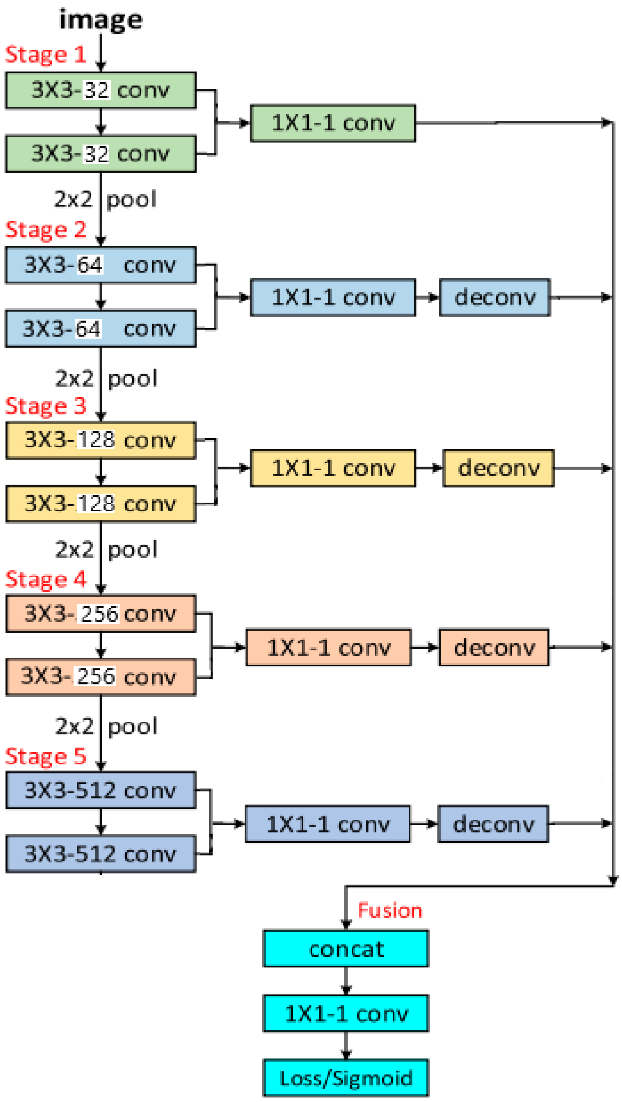

Figure 2 – The result of applying a destructive approach to the structural tuning of the RCF architecture

In addition, the layer containing the sigmoid activation function, followed by the calculation of the loss function, was removed from the architecture of the basic RCF neural network at each stage of processing (Fig. 2). This is due to the following. At each stage of edge detection by the basic RCF network edge probability map was formed as result of applying the sigmoid activation function. Then all obtaining probability edge maps were bilinear interpolated to the size of the original image. Further, for each stage, the value of the loss function was calculated taking into account the result of interpolation of the edge probability map and the ground-truth image. The values of the loss function at different stages were summed during network training.

Removing at each stage the layer containing the sigmoid activation function with subsequent calculation of the loss function avoids additional computational costs for determining the values of the loss function at different scales and introduces redundancy into the multiscale representation of the image. The latter contributes to an increase in the robustness of edge detection on natural images. In such images, it is often necessary to determine the intensity or color edges on the texture background. Natural images contain mostly statistical textures, which can be considered as noise and negatively affect the edge detection performance.

Further in the process of structural tuning of the RCF neural network, a constructive approach was used after applying the destructive approach. Namely, the layers of batch normalization and nonlinearity in the form of the ReLU activation function were added after the convolutional layers.

Batch normalization layer solves vanishing gradient problem. It is known that the error backpropagation algorithm converges faster if the input data is normalized (has zero mean and unit variance) [5, 30]. However, when a signal propagates through a neural network, its mean value and variance can change significantly. To avoid this, the standard normalization of the outputs of the convolutional layer is applied. Nevertheless, normalizing the output of a convolutional layer can change the representation of the data in the next layer. Therefore, two additional parameters γ_i and β_i are adjusted in the learning process along with the rest of the parameters and transform x_{ni} as $y_{ni}=\gamma_i x_{ni}+\beta_i$ [5].

The applying of batch normalization actually corresponds to edge contrasting which improves the edge detection performance. For CNNs, batch normalization reduces training time and reduces the chance of overfitting.

The ReLU activation function returns 0 for a negative argument, and in the case of a positive argument, returns the same. The applying of this function actually corresponds to thresholding in gradient edge detection methods. ReLU sharps object edges on an image because the advantage of this function over the sigmoid is the sparseness of activation (fewer neurons being activated).

The obtained neural network is named RCF-ST. It processes a three-channel image with a size of 320(480 pixels. Therefore, the initial color images were scaled to the specified size and then each image was inputted to the proposed neural network (Fig. 3). It is assumed that deconvolution and transposed convolution are the same operations. The architecture of RCF-ST network for the edge detection is shown in Table 1.

4 EXPERIMENTS

For experimental research of the results of each stage of the structural tuning of the neural network, the edge detection performance was evaluated for natural images from the BSDS500 dataset [31]. The dataset contains a total of 500 images, including 200 training images and 200 test images, and the remaining 100 validation images. The true values of the image edges are also presented on

DOI 10.15588/1607-3274-2023-4-12

ground-truth images which are binary images with contours selected by 5-8 experts (edge maps) [31]. The performance of edge detection was evaluated by comparing edges obtained in the RGB color space using the proposed CNN, with edges labeled by experts.

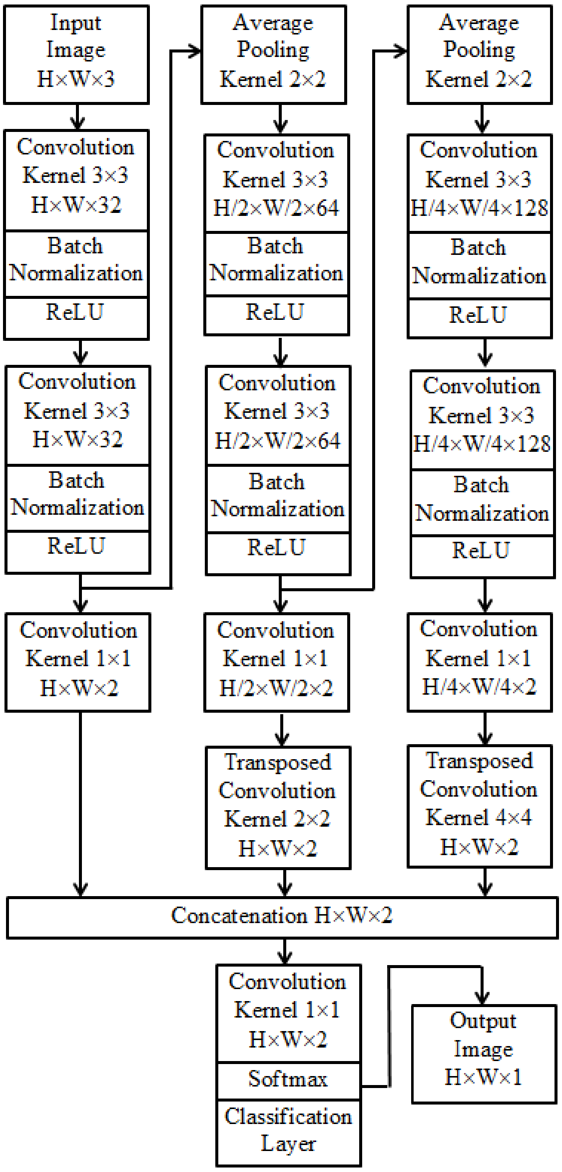

Figure 3 – The proposed RCF-ST network architecture for three stages of processing. It is assumed that deconvolution and transposed convolution are the same operations

The Adam method with an initial learning rate of 0.005 was used to train the RCF-ST network. A cross-entropy loss function was used, for which relative frequencies of the appearance of edge pixels and background pixels were taken into account [5].

To characterize the edge detection results the elements of confusion matrix *TP* and *TN* were used. In addition the *FOM* value [5] was estimated for the edge detection results. The *FOM* value is varied from 0 to 1 and normalized such that *FOM*=1 for a well detected edge.

For evaluation of edge detection results ODS and OIS are the widely used [1]. The ODS and OIS are defined based on F_β-score which is expressed as

$$F_\beta = (1+\beta^2)\, Pr\, \mathrm{Rc} \,/\, (\beta^2 Pr + Rc),$$

where $Pr = TP/(TP+FP)$, $Rc = TP/(TP+FN)$. The degree of significance of precision Pr and recall Rc can be controlled by adjusting the value of β. ODS and OIS indicate different ways of setting the threshold β in this formula. ODS is equal to F_β-score if a fixed threshold β is selected and applied to all images so that the F_β-score on the whole dataset is maximized. OIS is estimated from F_β-score if a different threshold β is selected on each image that maximizes the F_β-score of that image [1].

The experiment was conducted in accordance with the stages of structural tuning of the neural network. The advisability of each of the proposed stages was researched by estimating the edge detection performance using the *TP*, *TN*, *FOM*. The advisability of a structural tuning of the neural network as a whole was estimated by comparing it with methods known from the literature using the ODS and OIS.

First of all, as part of the experiment, the advisability of addition of the batch normalization layers after convolutional layers, and including the ReLU activation functions after the added batch normalization layers is researched. For this the proposed RCF-ST network, and RCF network were used to detect the object edges on natural images [31]. A number of stages of the RCF-ST network, and RCF network is varied from 3 to 5.

Further, the values of the selected indexes of the edge detection were evaluated depending on the number of convolutions in convolutional layers.

At the next stage of the experiment, as an alternative to transposed convolution, bicubic interpolation of image feature maps at different scales was used. The interpolated feature maps (layers 6, 12, 19 from Table 1) were concatenated. Then 1x1 convolutional layer, Softmax activation function, and pixel classification layer (layers 21-24 from Table 1) were applied.

Then the edge detection performance of the proposed RCF-ST network, and methods known from the literature is compared using ODS and OIS.

At the last part of experiment a number of parameters and the processing time of the edge detection on BSDS500 images was estimated for networks with considered architectures.

5 RESULTS

The elements of confusion matrices and *FOM* values for the results of 3, 4, 5 processing stages with the proposed RCF-ST network, and RCF network is shown in Table 2. Values in this table were obtained by averaging the *FOM*, *TP*, and *TN* for the edge detection on BSDS500 natural images.

DOI 10.15588/1607-3274-2023-4-12

Table 1 – The proposed RCF-ST network architecture

Layer number	Type	Comment	Activations	Learnables
1	Image input	320×480×3 image with zero center normalization	320×480×3	–
2	Convolution	32 3×3×3 convolutions with stride [1 1] and same padding	320×480×32	Weights: 3×3×3×32 Bias: 1×1×32
3	Batch normalization and ReLU	Batch normalization with 32 channels and activation function	320×480×32	Offset: 1×1×32 Scale: 1×1×32
4	Convolution	32 3×3×32 convolutions with stride [1 1] and same padding	320×480×32	Weights: 3×3×32×32 Bias: 1×1×32
5	Batch normalization and ReLU	Batch normalization with 32 channels and activation function	320×480×32	Offset: 1×1×32 Scale: 1×1×32
6	Convolution	2 1×1×32 convolutions with stride [1 1] and same padding	320×480×2	Weights: 1×1×32×2 Bias: 1×1×2
7	Average pooling	2×2 average pooling with stride [2 2] and padding [0 0 0 0]	160×240×32	–
8	Convolution	64 3×3×32 convolutions with stride [1 1] and same padding	160×240×64	Weights: 3×3×32×64 Bias: 1×1×64
9	Batch normalization and ReLU	Batch normalization with 64 channels and activation function	160×240×64	Offset: 1×1×64 Scale: 1×1×64
10	Convolution	64 3×3×64 convolutions with stride [1 1] and same padding	160×240×64	Weights: 3×3×64×64 Bias: 1×1×64
11	Batch normalization and ReLU	Batch normalization with 64 channels and activation function	160×240×64	Offset: 1×1×64 Scale: 1×1×64
12	Convolution	2 1×1×64 convolutions with stride [1 1] and same padding	160×240×2	Weights: 1×1×64×2 Bias: 1×1×2
13	Transposed convolution	2 2×2×2 transposed convolutions with stride [2 2] and output cropping [0 0]	320×480×2	Weights: 2×2×2×2 Bias: 1×1×2
14	Average pooling	2×2 average pooling with stride [2 2] and padding [0 0 0 0]	80×120×64	–
15	Convolution	128 3×3×64 convolutions with stride [1 1] and same padding	80×120×128	Weights: 3×3×64×128 Bias: 1×1×128
16	Batch normalization and ReLU	Batch normalization with 128 channels and activation function	80×120×128	Offset: 1×1×128 Scale: 1×1×128
17	Convolution	128 3×3×128 convolutions with stride [1 1] and same padding	80×120×128	Weights: 3×3×128×128 Bias: 1×1×128
18	Batch normalization and ReLU	Batch normalization with 128 channels and activation function	80×120×128	Offset: 1×1×128 Scale: 1×1×128
19	Convolution	2 1×1×128 convolutions with stride [1 1] and same padding	80×120×2	Weights: 1×1×128×2 Bias: 1×1×2
20	Transposed convolution	2 4×4×2 transposed convolutions with stride [4 4] and output cropping [0 0]	320×480×2	Weights: 4×4×2×2 Bias: 1×1×2
21	Concatenation	Concatenate the resulting feature maps from the output of the layers 6, 13, 20	320×480×6	–
22	Convolution	2 1×1×6 convolutions with stride [1 1] and same padding	320×480×2	Weights: 1×1×6×2 Bias: 1×1×2
23	Softmax	Activation function	320×480×2	–
24	Pixel classification	Class weighted cross-entropy loss with classes “edge” and “background”	–	–

Table 2 – The values of *TP*, *TN*, and *FOM* for the results of edge detection with the RCF-ST network (with batch normalization), and RCF network (without batch normalization)

Stages number	*TP*	*TN*	*FOM*
With batch normalization			
3	97.374	92.304	0.492
4	99.368	91.621	0.512
5	98.691	91.087	0.508
Without batch normalization			
3	84.892	84.214	0.337
4	82.369	78.315	0.301
5	77.270	74.454	0.295

Fig. 4 shows the initial BSDS500 images, the ground-truth images, edge maps, obtained by RCF-ST network with transposed convolution, and edge maps, obtained by RCF-ST network with interpolation. The multi-scale representation of BSDS500 images obtained by RCF-ST network is shown on Fig. 5. It can be seen from the Fig. 4 that edge detection by the RCF-ST network with transposed convolution is characterized by high performance.

In Table 3 the results of edge detection by the RCF-ST network with three processing stages is shown. The number of convolutions in convolutional layers is varied. The transposed convolution is applied for up-sampling of feature maps of initial image on different scales.

DOI 10.15588/1607-3274-2023-4-12

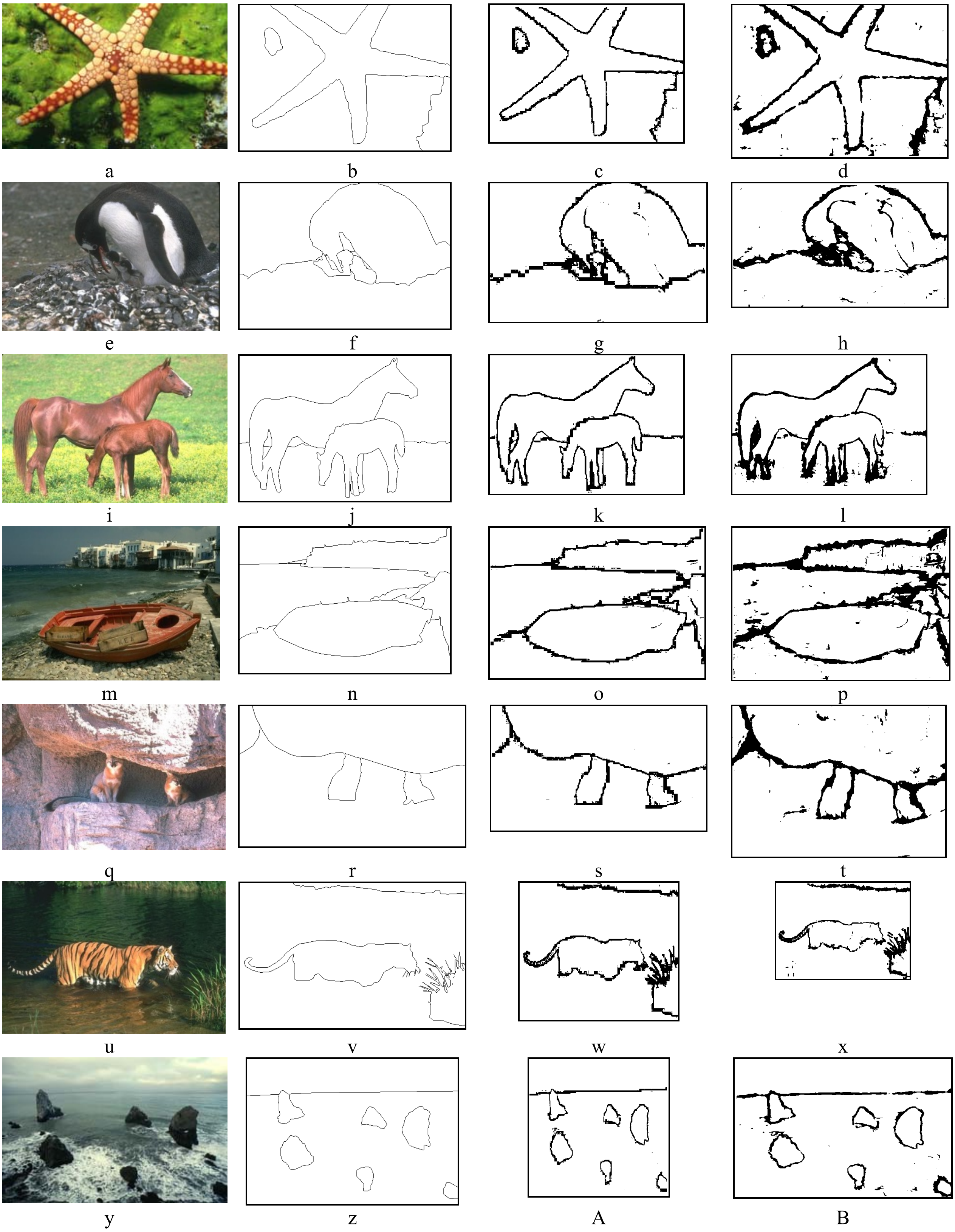

Figure 4 – The edge detection results by proposed RCF-ST network:
a, e, i, m, q, u, y – the initial BSDS500 images; b, f, j, n, r, v, z – the ground-truth images; c, g, k, o, s, w, A – edge maps, obtained by RCF-ST network with transposed convolution; d, h, l, p, t, x, B– edge maps, obtained by RCF-ST network with interpolation

DOI 10.15588/1607-3274-2023-4-12

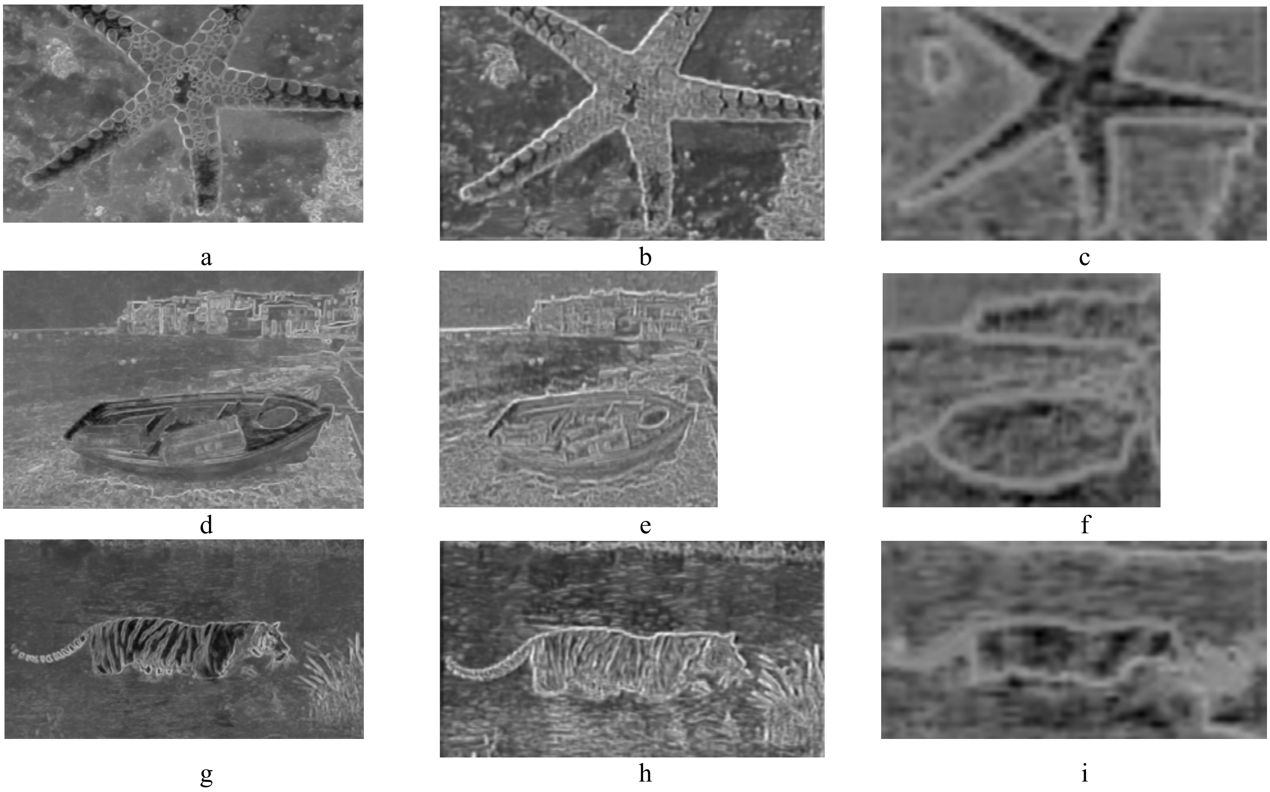

Figure 5 – The multi-scale representation of BSDS500 images from Fig. 4 obtained by RCF-ST network:
a, d, g – the fine scale images; b, e, h – the middle scale images; c, f, i – the coarse scale images

Table 3 – The results of edge detection by the RCF-ST network with the different number of convolutions in convolutional layers (three processing stages)

The number of convolutions in convolutional layers	*TP*	*TN*	*FOM*
8, 16, 32	96.423	86.705	0.343
16, 32, 64	98.062	91.219	0.445
32, 64, 128	97.374	92.304	0.492

The results of edge detection performance by the RCF-ST network with bicubic interpolation of image feature maps at different stages are presented in Table 4.

Table 4 – The values of *TP*, *TN*, and *FOM* for the results of edge detection with the RCF-ST network with interpolation

Stages number	*TP*	*TN*	*FOM*
With batch normalization			
3	96.383	87.292	0.344
4	95.149	87.026	0.377
5	95.926	86.299	0.369
Without batch normalization			
3	80.036	75.616	0.313
4	79.693	79.211	0.308
5	72.408	71.305	0.311

For comparison, the results of edge detection by the RCF network with similar architecture and bicubic interpolation of image feature maps at different stages is shown as well. The number of network processing stages is varied.

The edge detection performance of the BSDS500 images by the proposed RCF-ST network, and methods known from the literature are given in Table 5.

Table 5 – The ODS and OIS obtained from the BSDS500 images by the proposed RCF-ST network, and methods known from the literature [1, 21]

Reference, publication year, network name	ODS	OIS
[15], 2017, HED	0.788	0.808
[16], 2017, RCF	0.808	0.823
[17], 2018, LPCB	0.808	0.824
[18], 2022, BDCN	0.820	0.838
[19], 2020, DexiNed	0.831	0.845
[20], 2020, DSCD	0.826	0.857
[21], 2021, PiDiNet	0.807	0.823
Proposed RCF-ST network, 3 stages	0.823	0.853
Proposed RCF-ST network, 4 stages	0.887	0.894
Proposed RCF-ST network, 5 stages	0.862	0.872

For comparison, the edge detection performance of the Multicue dataset images by the methods known from the literature are given in Table 6.

Table 6 – The ODS and OIS obtained from Multicue dataset [32] images by the methods known from the literature [21]

Reference, publication year, network name	ODS	OIS
[15], 2017, HED	0.851	0.864
[16], 2017, RCF	0.857	0.862
[18], 2022, BDCN	0.891	0.898
[21], 2021, PiDiNet	0.858	0.863

DOI 10.15588/1607-3274-2023-4-12

6 DISCUSSIONS

Analysis of the indexes given in Table 2 showed that the using of batch normalization improves the edge detection performance. Specifically, *TP*, *TN*, and *FOM* have increased by 14–27%, 9–23%, 48–76% respectively. For reseached images more often background pixels were incorrectly assigned to the image edge.

Analysing Table 3 it should be noted the follow. If the number of convolutions in convolutional layers is decreased by two times then *FOM* are less by 9–32%. *TP* and *TN* mainly differed within the statistical error.

In addition, it is preferable to use average pooling than max pooling. The latter can enhance *TP* and *TN* by 0.5–1% depending on a number of processing stages.

The using of the interpolation instead of transposed convoluton of image feature maps at different processing stages is not advisability because the edge detection performance reduces. Specifically, *TP* is less by up to 4%, *TN* is less by 4–5%, *FOM* is less by up to 33%.

Analysis of the edge detection performance of the proposed RCF-ST network and the known methods [15–21] showed the following (Table 5). The ODS and OIS of the proposed RCF-ST network exceeds the known methods by 9–10% for BSDS500 images. The data given in Table 6 show that similar values of ODS and OIS are achievable by methods known from the literature, but on Multicue dataset [32].

A comparison of a number of parameters was made for the proposed RCF-ST network and RCF network with three processing stages. The basic RCF network with the architecture on Fig. 1 contains 1,758,600 parameters. The proposed RCF-ST network with the architecture in Table 1 contains 288,456 parameters. Thus, the number of parameters of the proposed RCF-ST network is 6 times less than the number of parameters of the basic RCF network, provided that the number of processing stages is equal.

A comparison of processing time was made for the edge detection on natural images by the proposed RCF-ST network with different number of processing stages. The researched natural images were cutted to a size of 320×480 pixels. Then the processing time of the RCF-ST network was calculated on average per image when training the network using the Adam method. It was 0.708–0.733; 0.921–0.982; 1.505–1.678 seconds per image when 3, 4, 5 processing stages are used correspondingly. The number of convolutions in convolutional layers was chosen as 32, 64, 128, 256, 512. The RCF network with similar architecture, that is, with the architecture as in Fig. 3, only without the batch normalization layers is also considered. It's average processing time for one image with 3, 4, 5 processing stages was 0.563–0.571; 0.640–0.653; 0.708–0.771 seconds correspondingly when trained by the Adam method. The research was performed using an Intel Core i5-7400 processor, 3 GHz CPU, 16GB memory, Windows 10 operating system, 64 bit. Thus, the proposed RCF-ST network requires on average 26–28, 44–50, 113–118 percents more time to process one image with 3, 4, 5 stages correspondingly than the RCF network. The number of training epochs of the RCF-ST network and RCF network for this edge detection problem is similar on average.

CONCLUSIONS

The actual scientific and applied problem of a structural tuning of a pre-synthesized neural network for edge detection on natural images has been solved.

The scientific novelty is the proposed technique of structural tuning of a deep learning neural network, which uses a sequentially destructive and constructive approach. According to the proposed technique, the network thinning and then removing at each stage the sigmoid activation function with subsequent calculation of the loss function were first performed as part of the destructive approach. Then, as part of a constructive approach, the batch normalization and ReLU layers are added after convolutional layers. As a result of applying this technique, the obtained RCF-ST network allows to improve the performance of edge detection on natural images. RCF-ST network is characterized by a significantly fewer parameters compared to the RCF network, which makes it possible to reduce the resource consumption of the network. Besides, RCF-ST network ensures the enhancing of the robustness of edge detection on texture background.

The practical significance of obtained results is that the software realizing the proposed RCF-ST network is developed, as well as experiments to research its edge detection performance are conducted. The experimental results allow to recommend the proposed RCF-ST for use in practice, as well as to determine effective conditions for the application of this network. The structural tuning technique may be used for informed design of the neural network architectures for other artificial intelligence problems.

Prospects for further research are to elaborate the postprocessing module which will thin and smooth the contours detected by the proposed RCF-ST network.

ACKNOWLEDGEMENTS

The author express their deep gratitude to V. N. Krylov, Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Information Technologies, National University "Odessa Polytechnic" for valuable and constructive advice and comments while working on this paper.

REFERENCES

1. Sun R., Lei T., Chen Q. et. al. Survey of image edge detection, *Frontiers of Signal Processing*, 2022, Vol. 2, article 826967. DOI: 10.3389/frsip.2022.826967
2. Leoshchenko S. D., Oliynyk A. O., Subbotin S. O., Hoffman E. O., Kornienko O. V. Method of structural adjustment of neural network models to ensure interpretability, *Radio electronics, computer science, management*, 2021, № 3, pp. 86–96. DOI: 10.15588/1607-3274-2021-3-8

DOI 10.15588/1607-3274-2023-4-12

3. D'souza R. N., Huang P. Y., Yeh F. C. Structural analysis and optimization of convolutional neural networks with a small sample size, *Scientific Reports*, 2020, № 10, pp. 1–13. DOI: 10.1038/s41598-020-57866-2
4. Nowakowski G., Dorogyy Y., Doroga-Ivaniuk O. Neural network structure optimization algorithm, *Journal of Automation, Mobile Robotics and Intelligent Systems*, 2018, № 12, pp. 5–13. DOI: 10.14313/JAMRIS_1-2018/1
5. Polyakova M. V. Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems, *Radio Electronics, Computer Science, Control*, 2023, № 1, pp. 51–61. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-5
6. Matychenko A. D., Polyakova M. V. The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space, *Herald of Advanced Information Technology*, 2023, Vol. 6, № 2, pp. 115–127. DOI: 10.15276/hait.06.2023.7
7. Yang J., Price B., Cohen S., Lee H., Yang M.-H. Object contour detection with a fully convolutional encoder-decoder network, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June 2016 : proceedings*. IEEE, 2016, pp. 193–202. DOI: 10.1109/cvpr.2016.28
8. Yu Z., Feng C., Liu M. Y., Ramalingam S. Casenet: Deep category-aware semantic edge detection, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings.* IEEE, 2017, pp. 5964–5973. DOI: 10.1109/cvpr.2017.191
9. Pu M., Huang Y., Guan Q., Ling H. RINDNet: Edge detection for discontinuity in reflectance, illumination, normal and depth, *Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021 : proceedings*. IEEE, 2021, pp. 6879–6888. DOI: 10.1109/iccv48922.2021.00680
10. Maninis K.-K., Pont-Tuset J., Arbeláez P., Van Gool L. Convolutional oriented boundaries, *Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)*. Springer, Cham, 2016, Vol. 9905, pp. 580–596. DOI: 10.1007/978-3-319-46448-0_35
11. Arbeláez P., Maire M., Fowlkes C., Malik J. Contour detection and hierarchical image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, Vol. 33, № 5, pp. 898–916. DOI: 10.1109/TPAMI.2010.161
12. Pont-Tuset J., Arbelaez P., Barron J., Marques F., Malik J. Multiscale combinatorial grouping for image segmentation and object proposal generation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, Vol. 39, № 1, pp. 128–140. DOI: 10.1109/TPAMI.2016.2537320
13. Xu D., Ouyang W., Alameda-Pineda X. et al. Learning deep structured multi-scale features using attention-gated CRFs for contour predict crisp boundaries, *Neural Information Processing Systems: 31st International Conference, NIPS'17, Long Beach, California, USA, 4–9 December, 2017 : proceedings*. Curran Associates Inc., 2017, pp. 3964–3973.
14. Mnih V., Heess N., Graves A., Kavukcuoglu K. Recurrent models of visual attention, *Neural Information Processing Systems: 27th International Conference, NIPS'14, Montreal, Canada, 8–13 December 2014 : proceedings*. MIT Press, 2014, Vol. 2, pp. 2204–2212.
15. Xie S., Tu Z. Holistically-nested edge detection, *International Journal of Computer Vision*, 2017, Vol. 125, № 5, pp. 1–16. DOI: 10.1007/s11263-017-1004-z
16. Liu Y., Cheng M. M., Hu X., Wang K., Bai X. Richer convolutional features for edge detection, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings*. IEEE, 2017, pp. 5872–5881. DOI: 10.1109/CVPR.2017.622
17. Deng R., Shen C., Liu S., Wang H., Liu X. Learning to predict crisp boundaries, *Computer Vision: 15th European Conference, ECCV, Munich, Germany, 8–14 September, 2018 : proceedings*. IEEE, 2018, part VI, pp. 562–578. DOI: 10.1007/978-3-030-01231-1_35
18. He J., Zhang S., Yang M., Chan Y., Huang T. BDCN: Bi-directional cascade network for all perceptual edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, Vol. 44, № 1, pp. 100–113. DOI: 10.1109/TPAMI.2020.3007074
19. Poma X. S., Riba E., Sappa A. Dense extreme inception network: towards a robust CNN model for edge detection, *Applications of Computer Vision: IEEE/CVF Winter Conference, WACV, Snowmass Village, CO, USA, 1–5 March 2020 : proceedings*. IEEE, 2020, pp. 1912–1921. DOI: 10.1109/WACV45572.2020.9093290
20. Deng R., Liu S. Deep structural contour detection, *Multimedia: 28th ACM International Conference, MM'20, Virtual Event, Seattle, WA, USA, October 12–16, 2020 : proceedings*. ACM, 2020, pp. 304–312. DOI: 10.1145/3394171.3413750
21. Su Z., Liu W., Yu Z. et al. Pixel difference networks for efficient edge detection, *Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021: proceedings*. IEEE, 2021, pp. 5117–5127. DOI: 10.1109/ICCV48922.2021.00507
22. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition, *Learning Representations: 3rd International Conference, ICLR 2015, San Diego, CA, USA, 7–9 May 2015 : proceedings* [Electronic resource]. Access mode: https://arxiv.org/pdf/1409.1556. DOI: 10.48550/arXiv. 1409.1556
23. Pinheiro P. O., Lin T.-Y., Collobert R., Dollár P. Learning to refine object segments, *Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)*. Springer, Cham, 2016, Vol. 9905, pp. 75–91. DOI: 10.1007/978-3-319- 46448-0_5
24. Chollet F. Xception: Deep learning with depthwise separable convolutions, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings*. IEEE, 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195
25. Dai J., Li Y., He K., Sun J. R-fcn: Object detection via region-based fully convolutional networks, *Neural Information Processing Systems: 30th International Conference, NIPS'16, Barcelona, Spain, 4–8 December 2016 : proceedings*. Red Hook, NY, ACM, 2016, pp. 379–387.
26. Chen L. C., Papandreou G., Kokkinos I., Murphy K., Yuille A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected Crfs, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, Vol. 40, № 4, pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184
27. Elsken T., Metzen J. H., Hutter F. Neural architecture search: a survey, *Journal of Machine Learning Research*, 2019, Vol. 20, № 55, pp. 1–21.
28. Leung F. H. F., Lam H. K., Ling S. H., Tam P. K. S. Tuning of the structure and parameters of neural network using an improved genetic algorithm, *IEEE Transactions on Neural*

DOI 10.15588/1607-3274-2023-4-12

Networks, 2003, Vol. 14, № 1, pp. 79–88. DOI: 10.1109/TNN.2002.804317

29. Xie W., Nagrani A., Chung J. S., Zisserman A. Utterance-level aggregation for speaker recognition in the wild, *Acoustics, Speech and Signal Processing: IEEE International Conference, ICASSP, Brighton, Great Britain, 12–17 May 2019 : proceedings*. IEEE, 2019, pp. 5791–5795. DOI: 10.1109/ICASSP.2019.8683120
30. Polyakova M. V., Krylov V. N. Data normalization methods to improve the quality of classification in the breast cancer diagnostic system, *Applied Aspects of Information Technology*, 2022, Vol. 5, № 1, pp. 55–63. DOI: 10.15276/aait.05.2022.5
31. The Berkeley Segmentation Dataset and Benchmark Web site (2019) [Electronic Resource]. Access mode: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.
32. The multi-cue boundary detection dataset. Video collection [Electronic Resource]. Access mode: http://serre-lab.clps.brown.edu/resources-static/multicue-dataset.tar.bz2

Received 27.08.2023.
Accepted 02.11.2023.

УДК 004.93

RCF-ST: СТРУКТУРНЕ НАЛАШТУВАННЯ НЕЙРОННОЇ МЕРЕЖІ З НАСИЧЕНІШИМИ ЗГОРТКОВИМИ ОЗНАКАМИ ДЛЯ ВИДІЛЕННЯ КОНТУРІВ НА ЗОБРАЖЕННЯХ РЕАЛЬНИХ СЦЕН

Полякова М. В. – д-р техн. наук, доцент, професор кафедри прикладної математики та інформаційних технологій Національного університету «Одеська політехніка», Одеса, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто проблему автоматизації виділення контурів на зображеннях реальних сцен в інтелектуальних системах. Предметом дослідження є згорткові нейронні мережі глибокого навчання для виділення контурів на зображеннях реальних сцен. Метою дослідження є підвищення якості виділення контурів на зображеннях реальних сцен шляхом структурного налаштування архітектури нейронної мережі з насиченішими згортковими ознаками.

Метод. Для автоматизації проектування архітектури нейронної мережі, що впливає на якість виділення контурів зображень, в роботі застосовано структурне налаштування. Обчислювальні витрати на структурне налаштування незрівнянно менші порівняно з пошуком нейронної архітектури, але потрібна більш висока кваліфікація дослідника, і отримане рішення буде субоптимальним. У цьому дослідженні послідовно застосовано спочатку деструктивний, а потім конструктивний підхід до структурного налаштування архітектури базової нейронної мережі RCF. Згідно конструктивному підходу для розширення мережі простої архітектури додаються приховані шари, вузли та з’єднання. Деструктивний підхід з мережі складної архітектури видаляє приховані шари, вузли та з’єднання щоб спростити мережу. Структурне налаштування нейронної мережі RCF з насиченішими згортковими ознаками включає: (1) зменшення кількості згорткових шарів; (2) зменшення кількості згорток у згорткових шарах; (3) видалення на кожному етапі сигмоїдної функції активації з подальшим обчисленням функції втрат; (4) додавання шарів пакетної нормалізації після згорткових шарів; (5) додавання функції активації ReLU після шарів пакетної нормалізації. Отримана нейронна мережа RCF-ST потребує масштабування початкових кольорових зображень до заданого розміру перед поданням на вхід мережі. Доцільність кожного із запропонованих етапів структурного налаштування мережі досліджувано шляхом оцінки якості виділення контурів за допомогою елементів матриці помилок та критерія Претта. Доцільність структурного налаштування нейронної мережі в цілому оцінено шляхом її порівняння з відомими з літератури методами за допомогою Optimal Dataset Scale та Optimal Image Scale.

Результати. Запропоновану згорткову нейронну мережу програмно реалізовано та досліджено для розв’язання завдання виділення контурів на зображеннях реальних сцен. Запропоновані етапи структурного налаштування можна використовувати під час обгрунтованого проектування архітектури нейронної мережі для розв’язання інших завдань штучного інтелекту.

Висновки. Отримана мережа RCF-ST дозволяє підвищити якість виділення контурів на зображеннях. Мережа RCF-ST характеризується значно меншою кількістю параметрів у порівнянні з мережею RCF, що дозволяє знизити ресурсоспоживання мережі. Крім того, мережа RCF-ST забезпечує підвищення завадостійкості виділення контурів на фоні текстури.

КЛЮЧОВІ СЛОВА: зображення реальних сцен, виділення контурів, згорткова мережа, насиченіші згорткові ознаки, структурне налаштування, пакетна нормалізація.

ЛІТЕРАТУРА

1. Survey of image edge detection / [R. Sun, T. Lei, Q. Chen et. al.] // Frontiers of Signal Processing. – 2022. – Vol. 2. – Article 826967. DOI: 10.3389/frsip.2022.826967
2. Метод структурного доналаштування нейромережевих моделей для забезпечення інтерпретабельності / [С. Д. Леощенко, А. О. Олійник, С. О. Субботін та ін.] // Радіоелектроніка, інформатика, управління. – 2021. – № 3. – С. 86–96. DOI: 10.15588/1607-3274-2021-3-8
3. D’souza R. N. Structural analysis and optimization of convolutional neural networks with a small sample size / R. N. D’souza, P. Y. Huang, F. C. Yeh // Scientific Reports. – 2020. – № 10. – P. 1–13. DOI: 10.1038/s41598-020-57866-2
4. Nowakowski G. Neural network structure optimization algorithm / G. Nowakowski, Y. Dorogyy, O. Doroga-Ivaniuk // Journal of Automation, Mobile Robotics and Intelligent Systems. – 2018. – № 12. – P. 5–13. DOI: 10.14313/JAMRIS_1-2018/1
5. Polyakova M. V. Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems / M. V. Polyakova // Radio Electronics, Computer Science, Control. – 2023. – № 1. – P. 51–61. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-5
6. Matychenko A. D. The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space / A. D. Matychenko, M. V. Polyakova // Herald of Advanced Information Technology. – 2023. – Vol. 6, № 2. – P. 115–127. DOI: 10.15276/hait.06.2023.7
7. Object contour detection with a fully convolutional encoder-decoder network / [J. Yang, B. Price, S. Cohen et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June 2016

DOI 10.15588/1607-3274-2023-4-12

: proceedings. – IEEE, 2016. – P. 193–202. DOI: 10.1109/cvpr.2016.28
8. Casenet: Deep category-aware semantic edge detection / [Z. Yu, C. Feng, M. Y. Liu, S. Ramalingam] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 5964–5973. DOI: 10.1109/cvpr.2017.191
9. RINDNet: Edge detection for discontinuity in reflectance, illumination, normal and depth / [M. Pu, Y. Huang, Q. Guan, H. Ling] // Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021 : proceedings. – IEEE, 2021. – P. 6879–6888. DOI: 10.1109/iccv48922.2021.00680
10. Convolutional oriented boundaries / [K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, L. Van Gool] // Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds). – Springer, Cham, 2016. – Vol. 9905. – P. 580–596. DOI: 10.1007/978-3-319-46448-0_35
11. Contour detection and hierarchical image segmentation/ [P. Arbeláez, M. Maire, C. Fowlkes, J. Malik] // IEEE Trans. Pattern Anal. Mach. Intell. – 2010. – Vol. 33, № 5. – P. 898–916. DOI: 10.1109/TPAMI.2010.161
12. Multiscale combinatorial grouping for image segmentation and object proposal generation / [J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marques, J. Malik] // IEEE Trans. Pattern Anal. Mach. Intell. – 2016. – Vol. 39, № 1. – P. 128–140. DOI: 10.1109/TPAMI.2016.2537320
13. Learning deep structured multi-scale features using attention-gated CRFs for contour predict crisp boundaries / [D. Xu, W. Ouyang, X. Alameda-Pineda et al.] // Neural Information Processing Systems: 31st International Conference, NIPS'17, Long Beach, California, USA, 4–9 December, 2017 : proceedings. – Curran Associates Inc., 2017. – P. 3964–3973.
14. Recurrent models of visual attention / [V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu] // Neural Information Processing Systems: 27th International Conference, NIPS'14, Montreal, Canada, 8–13 December 2014 : proceedings. – MIT Press, 2014. – Vol. 2. – P. 2204–2212.
15. Xie S. Holistically-nested edge detection / S. Xie, Z. Tu // International Journal of Computer Vision. – 2017. – Vol. 125, № 5. – P. 1–16. DOI: 10.1007/s11263-017-1004-z
16. Richer convolutional features for edge detection / [Y. Liu, M. M. Cheng, X. Hu et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 5872–5881. DOI: 10.1109/CVPR.2017.622
17. Learning to predict crisp boundaries / [R. Deng, C. Shen, S. Liu et al.] // Computer Vision: 15th European Conference, ECCV, Munich, Germany, 8–14 September, 2018 : proceedings. – IEEE, 2018. – Part VI. – P. 562–578. DOI: 10.1007/978-3-030-01231-1_35
18. BDCN: Bi-directional cascade network for all perceptual edge detection / [J. He, S. Zhang, M. Yang et al.] // IEEE Trans. Pattern Anal. Mach. Intell. – 2022. – Vol. 44, № 1. – P. 100–113. DOI: 10.1109/TPAMI.2020.3007074
19. Poma X. S. Dense extreme inception network: towards a robust CNN model for edge detection / X. S. Poma, E. Riba, A. Sappa // Applications of Computer Vision: IEEE/CVF Winter Conference, WACV, Snowmass Village, CO, USA, 1–5 March 2020 : proceedings. – IEEE, 2020. – P. 1912–1921. DOI: 10.1109/WACV45572.2020.9093290
20. Deng R. Deep structural contour detection / R. Deng, S. Liu // Multimedia: 28th ACM International Conference, MM'20, Virtual Event, Seattle, WA, USA, October 12–16, 2020 : proceedings. – ACM, 2020. – P. 304–312. DOI: 10.1145/3394171.3413750
21. Pixel difference networks for efficient edge detection / [Z. Su, W. Liu, Z. Yu et al.] // Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021: proceedings. – IEEE, 2021. – P. 5117–5127. DOI: 10.1109/ICCV48922.2021.00507
22. Simonyan K. Very deep convolutional networks for large-scale image recognition / K. Simonyan, A. Zisserman // Learning Representations: 3rd International Conference, ICLR 2015, San Diego, CA, USA, 7–9 May 2015 : proceedings [Electronic resource]. – Access mode: https://arxiv.org/pdf/1409.1556. DOI: 10.48550/arXiv.1409.1556
23. Learning to refine object segments / [P. O. Pinheiro, T.-Y. Lin, R. Collobert, P. Dollár] // Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds). – Springer, Cham, 2016. – Vol. 9905. – P. 75–91. DOI: 10.1007/978-3-319-46448-0_5
24. Chollet F. Xception: Deep learning with depthwise separable convolutions / F. Chollet // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 1800–1807. DOI: 10.1109/CVPR.2017.195
25. R-fcn: Object detection via region-based fully convolutional networks / [J. Dai, Y. Li, K. He, J. Sun] // Neural Information Processing Systems: 30th International Conference, NIPS'16, Barcelona, Spain, 4–8 December 2016 : proceedings. – Red Hook, NY, ACM, 2016. – P. 379–387.
26. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected Crfs / [L. C. Chen, G. Papandreou, I. Kokkinos et al.] // IEEE Trans. Pattern Anal. Mach. Intell. – 2017. – Vol. 40, № 4. – P. 834–848. DOI: 10.1109/TPAMI.2017.2699184
27. Elsken T. Neural architecture search: a survey / T. Elsken, J. H. Metzen, F. Hutter // Journal of Machine Learning Research. – 2019. – Vol. 20, № 55. – P. 1–21.
28. Tuning of the structure and parameters of neural network using an improved genetic algorithm / [F. H. F. Leung, H. K. Lam, S. H. Ling, P. K. S. Tam] //IEEE Transactions on Neural Networks. – 2003. – Vol. 14, № 1. – P. 79–88. DOI: 10.1109/TNN.2002.804317
29. Utterance-level aggregation for speaker recognition in the wild / [W. Xie, A. Nagrani, J. S. Chung, A. Zisserman] // Acoustics, Speech and Signal Processing: IEEE International Conference, ICASSP, Brighton, Great Britain, 12–17 May 2019 : proceedings. – IEEE, 2019. – P. 5791–5795. DOI: 10.1109/ICASSP.2019.8683120
30. Polyakova M. V. Data normalization methods to improve the quality of classification in the breast cancer diagnostic system / M. V. Polyakova, V. N. Krylov // Applied Aspects of Information Technology. – 2022. – Vol. 5, № 1. – P. 55–63. DOI: 10.15276/aait.05.2022.5
31. The Berkeley Segmentation Dataset and Benchmark Web site (2019) [Electronic Resource]. – Access mode: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.
32. The multi-cue boundary detection dataset. Video collection [Electronic Resource]. – Access mode: http://serre-lab.clps.brown.edu/resources-static/multicue-dataset.tar.bz2

DOI 10.15588/1607-3274-2023-4-12