LEARNED PARAMETERIZED CONVOLUTIONAL APPROXIMATION OF IMAGE FILTERS

Olga Chaganova
Institute for Information
Transmission Problems, RAS
Bolshoy Karetny per. 19, Moscow, 127051, Russia;
Moscow Institute of Physics and Technology
(National Research University)
Institutskiy per. 9, Dolgoprudny, 141701, Russia
E-mail: o.chaganova@visillect.com

Anton Grigoryev
Institute for Information
Transmission Problems, RAS
Bolshoy Karetny per. 19, Moscow, 127051, Russia
E-mail: me@ansgri.com

KEYWORDS

Convolutional neural networks, grayscale morphology, edge detection, computational efficiency, image processing

ABSTRACT

Multilayer neural networks are considered universal approximators applicable to a wide range of problems. There are quite detailed theoretical and applied studies for fully connected networks, while for convolutional networks the results are more scarce. In this paper, we tested the approximating capability of deep neural networks with typical architectures like ConvNet, ResNet, and UNet as applied to classical image processing algorithms. Canny edge detector and grayscale morphological dilation with the disk structuring element were selected as target algorithms. As we have seen, even relatively lightweight neural models are able to approximate a filter with fixed parameters. Since image processing algorithms are parameterized, we considered different approaches to the parameterization of the neural networks and discovered that even the simplest one, which is an adding parameters in the input image channels, works well for filter with low parameter count. Also, we measured an inference time of a neural network approximation and a classical implementation of the grayscale dilation with the disk structuring element. Starting from a certain radius, a neural network works faster than the algorithm even on one core of the CPU without fine-tuning the architecture for performance, thus confirming the viability of ConvNets as a differentiable approximation technique for optimization of classical-based methods.

INTRODUCTION

Nowadays, artificial neural networks (ANNs) are considered universal approximators for analytic functions of any complexity. Theoretical studies have been conducted for feedforward neural networks; in particular, Hornik et al. (1989) proved that standard perceptron is capable of approximating any Borel measurable function from one finite-dimensional space to another to any desired degree of accuracy. Similar studies for convolutional networks showed that a deep convolutional neural network can approximate any continuous function to an arbitrary accuracy when its depth is large enough (Zhou 2019). But there is still a lack of studies about the neural networks capability of approximating algorithms, particularly image processing algorithms. Many convolutional network architecture families, such as ConvNet, ResNet, UNet, etc., are successfully used in various image processing tasks (Li et al. 2021; Andreeva et al. 2019), including ones with existing algorithmic solutions (Karnaushkin and Sklyarenko 2022; Panfilova and Kunina 2020). However, it is not obvious that especially for tasks where specialized algorithmic solutions are known, a general-purpose neural network could solve the same task with comparable computational and representational efficiency. To research this aspect, we decided to test the ability of typical NN architectures to approximate individual image processing operations, since difficulties with this simplified problem would indicate deeper issues for use of neural networks as a general-purpose tool for image processing. We chose Canny edge detector and morphological dilation as examples of typical image processing operations since the former is comparatively well-researched from the neural approximation point of view (Fernández et al. 2011), and for the latter there is no computationally efficient algorithm anyway.

The rest of the paper is structured as follows: the next section provides a brief theoretical overview of existing work on approximating classical algorithms with neural networks and approaches to parameterization of the neural networks. The following section includes a description of the experiments and a discussion of the results. Finally, the paper is concluded with a summary and pointers to the possible future work.

RELATED WORK

In this section, we will describe the main existing results on the approximation of classical filters, which can

Communications of the ECMS, Volume 36, Issue 1,
Proceedings, ©ECMS Ibrahim A. Hameed, Agus Hasan,
Saleh Abdel-Afou Alaliyat (Editors) 2022
ISBN: 978-3-937436-77-7/978-3-937436-76-0(CD) ISSN 2522-2414

be considered as baselines for further research. Since classical algorithms have the important property of natural parametrization, which means they can be customized for specific requirements, we will also consider approaches to the parametrization of neural networks.

Neural network approximation of classical algorithms

The first works on the research topic appeared even in the 20th century. For example, De Ridder et al. (1999) investigated the application of neural networks to nonlinear image processing using Kuwahara filter for image smoothing as a target. The authors experimented with several architectures of a network, including single- and double-layer perceptrons with different sizes of the hidden layers and a specially designed modular neural network. Although the authors managed to train the model to smooth the images, the result was still too far from true Kuwahara filter smoothing. The reason for this could be due to unsuitable model architecture and loss function.

Fernández et al. (2011) attempted to approximate the Canny and Sobel edge detectors with single-layer perceptron. To take into account the spatial dependence in the images, a pixel and its eight adjacent pixels were fed to the input of the model, which almost corresponds to a convolution with a kernel size of 3x3. The authors did not manage to achieve a good quality of the approximation, which can probably be explained by the inability of the single-layer model to learn complex dependencies in the data. However, these studies can be taken as a baseline for subsequent experiments.

Fernández et al. (2011); De Ridder et al. (1999)) used neural networks with typical architecture: just a multilayer perceptron. A totally different approach was proposed by Zhukovskiy et al. (2018). The authors developed a complete reproduction of the Canny, Niblack, and Harris filters using only neural network operations: convolution, pulling, concatenation, etc., so the algorithm's parameters do not need to be selected manually because their optimal values will be determined during training. But this approach has its disadvantage since it requires creating a new architecture for each filter, which is very time-consuming and requires a deep understanding of the algorithm's inner workings. It also does not correlate with the aim of our study.

Parameterization approaches

In the previous works, the ability of neural networks to approximate a particular filter with fixed parameters was investigated, and those researches are of more theoretical interest. Even though in real-world applications this approach is acceptable, because a filter with specified parameters is often used, it is necessary to be able to customize the filter parameters when creating and optimizing prototypes of the final system. For example, Karnaushkin and Sklyarenko (2022) used Canny edge detector to develop a computer vision-based method of pre-alignment of a channel optical waveguide and a lensed fiber, and the values of the algorithm parameters were being tuned during the development stage. Sometimes it is necessary to give a user the ability to specify the system parameters, as in the case of an ANN-based image pre-compensation system, which must adjust to the parameters of a particular person's eye (Yu et al. 2021) and thus cannot be retrained for each possible set of parameters.

Model parameterization can be achieved in several ways:

(1) by adding parameters values as metadata to the input image channels. This approach was used by Tziolas et al. (2020). In order to enrich the data and improve the quality of the model predicting the clay content of the soil, not only remotely sensed values, which constitute the spectral input channel, but also geographic coordinates, which constitute the auxiliary input channels, are fed as an input. This is the simplest way to provide dependence of the model on auxiliary parameters, which does not entail a significant increase in the size of the model, but in some cases, it might be not enough to learn more complex dependencies between parameters and expected output.

(2) by parameterizing the convolution kernel. Traditional convolutional kernels are shared for all examples in a dataset, which can significantly decrease the quality of the model in case the data are characterized by a large variability. Conditional convolutions (CondConv), proposed by Yang et al. (2020), and Dynamic convolutions (DynamicConv), proposed by Chen et al. (2020)), share a common idea: instead of using a single convolution kernel per layer, which is the same for any input, these types of convolutions aggregates multiple convolution kernels, which are input dependent. DynamicConv aggregates several kernels based upon their attentions while CondConv parameterizes the convolutional kernels as a linear combination with example-dependent scalar weights computed using a routing function with learned parameters. Another example of this approach is the Adaptive convolutional neural network (ACNN), presented in Kang et al. (2017). In ACNN the filter weights are generated with a learned sub-network, which is a simple multilayer perceptron and input of which is the side information or metadata. Replacing traditional convolution layers by convolutions with adaptive weights can disentangle the variations related to the side information and extract discriminative features related to the current context, but it can significantly increase the size of the model and slow down its inference time.

(3) by adding extra input layers for metadata to the neural network. Wei (2020) and Gessert et al. (2020) used a multiple-input neural network, which consists of two branches: the first one is a single- or multilayer perceptron that process a vector of metadata and the second one is a convolutional neural network that process an input image. The outputs of these sub-networks are then concatenated and processed with the common part of the network. Ellen et al. (2019) developed this idea and proposed several schemes for metadata incorporation. While in the previous works metadata was added

to a convolutional neural network, it can be added to a graph neural network too (Mudiyanselage et al. 2021).

For our experiments, we have chosen two classical algorithms: Canny edge detector and grayscale morphology, namely, dilation with the disk structuring element. Both of them are often used in industrial recognition systems (e.g. Panfilova and Kunina (2020), Panfilova et al. (2021), Erlygin and Teplyakov (2021)). There is a baseline on an approximation of Canny operator (Fernández et al. 2011), but there was a feedforward neural network used, while research for convolutional neural networks was not conducted. Also, we did not manage to find any published studies for approximation of morphological dilatation. Moreover, there is no computationally efficient implementation of dilation with the disk structuring element compared, at least, to the rectangular structuring element, for which fast window size-independent algorithm (vHGW) is available (Limonova et al. 2020).

METHODS AND RESULTS

In this section, we will describe the results of experiments on approximation Canny edge detector and grayscale dilation with the disk structuring element. Using them as target filters allowed us to investigate two different approaches to approximating filters: regression and classification. Since Canny detects edges on the image, it might be considered as a classification of each pixel whether it is a part of an edge or not. On the other hand, morphological filters do not „classify“each pixel; instead, they transform an input image, and this transformation might be considered as the image-to-image regression.

Approximation with fixed parameters

At first, we wanted to test the ability of the neural networks to approximate a particular filter with specified parameters. There are three key parameters of Canny edge detector (Canny 1986): standard deviation for Gaussian kernel σ, lower bound (low threshold) and upper bound (high threshold) for hysteresis thresholding. We specified these parameters as $\sigma = 1$, low threshold $= 0.1$, high threshold $= 0.2$. For morphological dilation, we approximated the filter with two sizes of the radius: $R_1 = 5$ as a baseline and $R_2 = 20$ to be sure that the receptive field of the network is large enough to approximate dilation with a big radius.

We tried three architectures: ConvNet, which is a stack of convolutional layers with ReLU as activation function, ResNet, and UNet. These architectures might be considered as the gold standard of convolutional networks because they are well-known and often used in many cases of image processing and computer vision (Li et al. 2021). We considered only the most typical neural network architectures without optimizing them for a particular filter. The reason is that our study aims to investigate the universality of standard neural network models for solving classical image processing problems for which there already exists a well-working analytical algorithm.

As we mentioned earlier, the problem of Canny edge detector approximation can be formulated in terms of a pixel-by-pixel classification, thus we used the binary cross-entropy loss function. The metrics are Matthew's correlation coefficient, or MCC, (Cheng et al. 2021), and Intersection over Union, or IoU, (Zheng et al. 2019). For morphological dilation approximation, we used the mean squared error loss function and mean absolute error as a metric. Also, we compared different models using the loss function value on the test dataset.

For each filter, all three networks were trained with the same hyperparameters. Particularly, we used Adam optimizer with learning rate 0.001 and parameters $\beta_1 = 0.9, \beta_2 = 0.999$; cosine annealing scheduler with warm restarts with a period of 8 epochs; 50 epochs for training. As the dataset, we used Linnaeus5 with images of size 128x128 pixels split into training, validation, and test sets with 6000, 1000, and 1000 images, respectively.

All three models showed comparable results for morphology approximation, but for Canny detector approximation, the best quality was achieved with ResNet (table 1), that is why in the following experiments we used only this architecture. Our implementation of ResNet reproduces the classic implementation and differs only in the number of layers and the number of filters in convolutional layers. It consists of 7 residual blocks, each of them containing 2 convolutional layers with 12 filters. A visualization of the ResNet work is shown in the fig. 1 and fig. 2 for Canny edge detector and grayscale morphology with the disk structuring element, respectively.

Model	Num. of parameters	Metrics		
		MCC	IoU	Test BCE
ConvNet	25.8k	0.9185	0.9230	0.0462
ResNet	29.1k	**0.9286**	**0.9322**	**0.0425**
UNet	28.7k	0.8291	0.8481	0.1020

TABLE 1: Metrics for Canny edge detector approximation

Moreover, we compared inference times of the ResNet (fig. 3) with the skimage.morphology (v.0.17.2) implementation and noticed that starting from a certain radius, neural network approximation works faster than the classical algorithm even in single-threaded inference mode. It should be noted that the inference time of the neural network is constant and depends only on technical conditions of implementation, since the input image size is fixed and independent of the approximated filter parameters.

In fully parallel GPU inference, the ResNet is faster than the CPU implementation in all cases, and while such performance comparison is unfair, it illustrates one of the benefits of NN approximations, which is the availability of highly parallelized implementations. It is worth mentioning that we did not apply any optimization techniques to speed up inference of the ResNet, although there is a more computational efficient implementation of it (e.g. Lobanov and Sholomov (2019), which provides a threefold increase in the inference performance). The parameters of the testing system are:

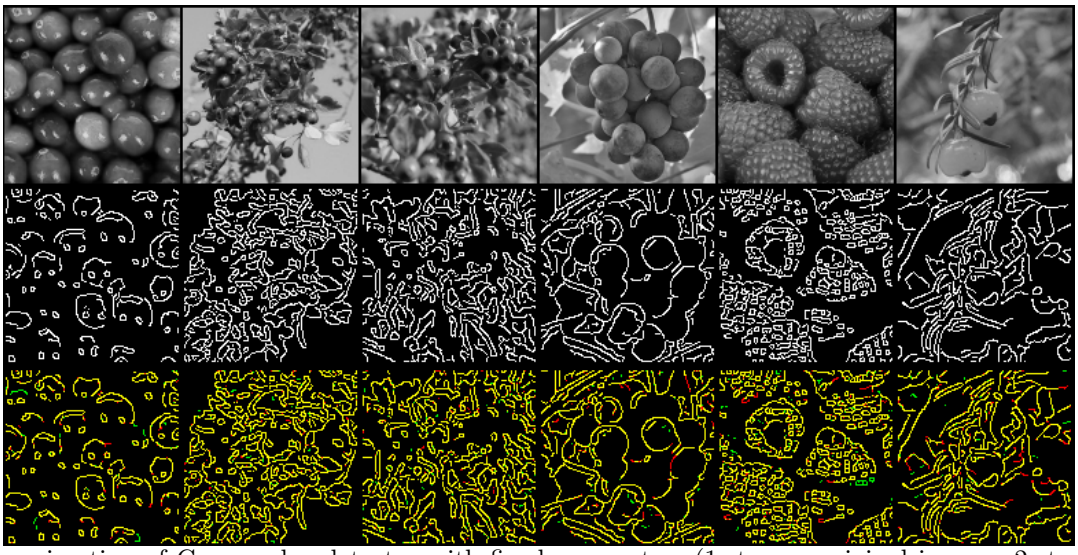

Fig. 1: Approximation of Canny edge detector with fixed parameters (1-st row: original images, 2-st row: Canny edge detector, 3-st row: pixel difference with the neural network output, where black, yellow, green, and red colours stand for true negative, true positive, false positive, and false negative pixels, respectively).

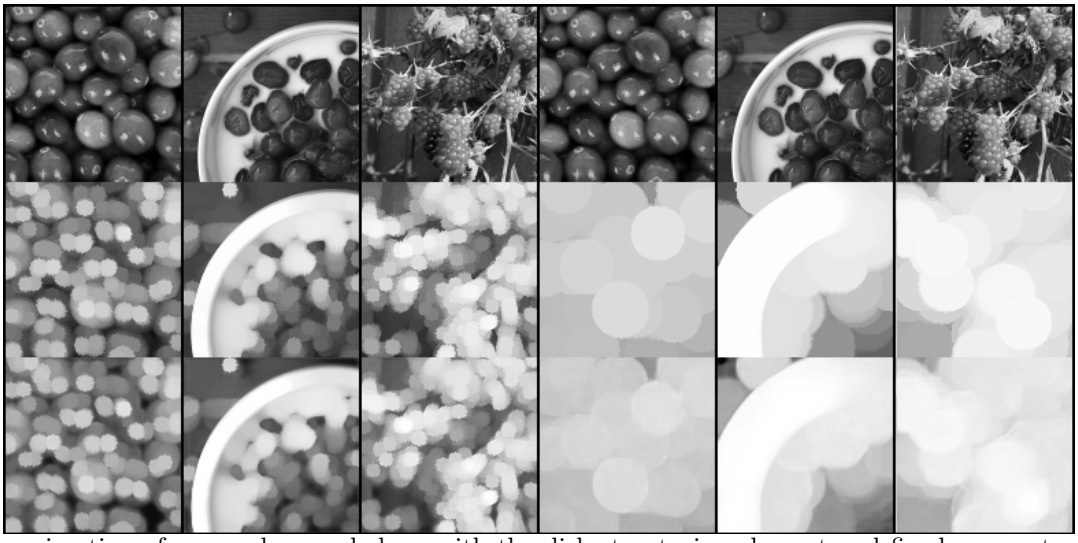

Fig. 2: Approximation of grayscale morphology with the disk structuring element and fixed parameters (1-st row: original images, 2-st row: morphological dilation, 3-st row: neural network). The first three images correspond to dilation with a radius of 5, while the second three correspond to dilation with a radius of 20. Two networks were trained to approximate the morphological dilation with different radius independently.

Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz, 6 cores, 12 threads; GeForce RTX 2080 Ti 11 Gb GPU.

Thus, we found that typical neural network architectures are able to approximate at least some of the typical image processing algorithms, while having few enough weights to be computationally competitive with purpose-built algorithms.

Parameterized approximation

An important property of classical image processing algorithms is parameterization, therefore they can be customized for a particular task while neural networks usually implement a fixed operation. In real-world applications, it is often necessary for a developer or a user to be able to tune the parameters. That is why another aim of our research is to consider the approaches to the parameterization of neural networks. We started with the simplest approach, which is adding filter parameters as auxiliary channels of an input image.

Canny detector has three key parameters: sigma, low threshold, and high threshold. The high threshold was parameterized via the low threshold multiplied by a coefficient k. We consider these parameters as independent uniformly distributed in the interval [0; 3] for sigma, [0.1; 0.2] for low threshold, and [1.05; 2] for k, random variables. Since the distribution is multidimensional and its components are independent, it is crucial to sample from it with as maximum coverage of the probability space as possible. Latin hypercube sampling (LHS), proposed by Mckay et al. (1979), ensures that the set of samples is a very good representative of the real variability. That is why we used it for sampling the set of parameters during training phase.

The training was organised as follows: at each epoch,

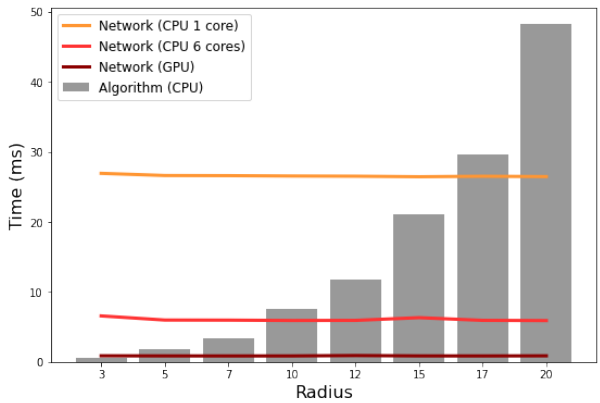

Fig. 3: Inference time of classical algorithm (disk-structured morphological dilation) and its neural approximation with (unparameterized) ResNet. The CPU algorithm implementation is single-threaded.

we sampled 1000 sets of parameters with LHS. For each image from the training set, we randomly chose a set of parameters and added its values to the channels of the image. Since there are 6000 images in the training set, each set of parameters was applied to roughly six images. In the case of continuous random variables, the number of possible sets of parameters is infinite, that is why it makes sense to increase the number of epochs so that the network can „see „as many sets of parameters as possible and learn how to approximate them. We trained our model for 200 epochs, which means there were 200000 sets of parameters. Also, we had to increase the number of convolutional kernels in the hidden layers from 12 to 16 compared to the unparameterized version. A visualization of the ResNet work is shown in the fig. 4. A dependence on the parameters is pronounced, but the quality of the approximation is slightly reduced compared to the approximation with fixed parameters. The Matthew's correlation coefficient is 0.8917 and IoU is 0.9021.

Disk-structured morphological dilation has one parameter, which is the radius of the disk. Since the radius must be a positive integer, we consider it a discrete random variable, uniformly distributed in the interval [1; 20]. The model was trained in the same way as the Canny approximation network, but we did not use LHS because it is not necessary in case of one-dimensional discrete probability space. A visualization of the ResNet work is shown in the fig. 5. Received results showed that the simplest parameterization approach via channels of input image works well enough for some applications.

CONCLUSIONS

In this paper, we tested the hypothesis that neural networks can be effectively used to approximate and replace classical image processing operations. We chose two typical image filters, Canny edge detector and grayscale morphological dilation with the disk structuring element. This choice of algorithms allowed us to compare two different approaches to the approximation: classification and regression. As we have seen, neural networks are equally good at both tasks, all that is required is to choose an appropriate loss function.

We used typical convolutional networks architectures like ConvNet, ResNet, and UNet with the standard loss functions (binary cross-entropy and mean squared error), not trying to optimize the architecture for a particular filter. As we have seen, even a neural network model with a relatively small number of parameters is able to approximate a filter with fixed parameters. The only requirement is the large enough number of hidden layers, which determine the size of the model's receptive field. These results might be considered as an argument in favour of the tested hypothesis, obtained experimentally.

Since in real-world applications adjustability of filters might be crucial, we also tested the ability of a single neural network model to learn to approximate chosen filters with different parameter values. We have considered various approaches to parameterization of the neural networks, but even the simplest one, which is adding parameters in the input image channels, works well enough. There are still some artifacts, so for the better quality of the approximation, it is worthwhile to apply more complex approaches previously discussed.

Approximation of classical image processing algorithms with neural networks is not only interesting from a theoretical point of view but also leads to practical benefits:

(1) a neural network is trainable, and thus can potentially achieve a better solution than a classical filter, which is deterministic and parameters of which must be chosen manually. For example, Jampani et al. (2016) proposed learnable bilateral filters, the performance of which was improved compared to a fixed parametric form;

(2) starting from the specified size of the radius, a neural network model (with ResNet-like architecture, in our case) works faster than the classical implementation of the grayscale morphology with the disk structuring element (fig. 3) even on one core of the CPU without applying any optimization techniques;

(3) despite the fact that inference speed of a classical algorithm might be lower compared to its neural network approximation in a single-core mode, neural networks can be easier parallelized both on multi-cores CPUs and GPUs, development of which, in addition, is constantly evolving in speed and power consumption (Di Febbo et al. 2018);

(4) a neural network approximation can replace the classical filter within a more complex system based on ANNs as well, allowing for end-to-end system training (Yi et al. 2016).

Thus, in this paper we have found arguments in favour of common hypothesis that neural networks are universal approximators experimentally. The direction of further research is the investigation of other approaches to parameterization of neural networks, which can help to achieve the better quality of approximation without significant increasing of the model's parameters count.

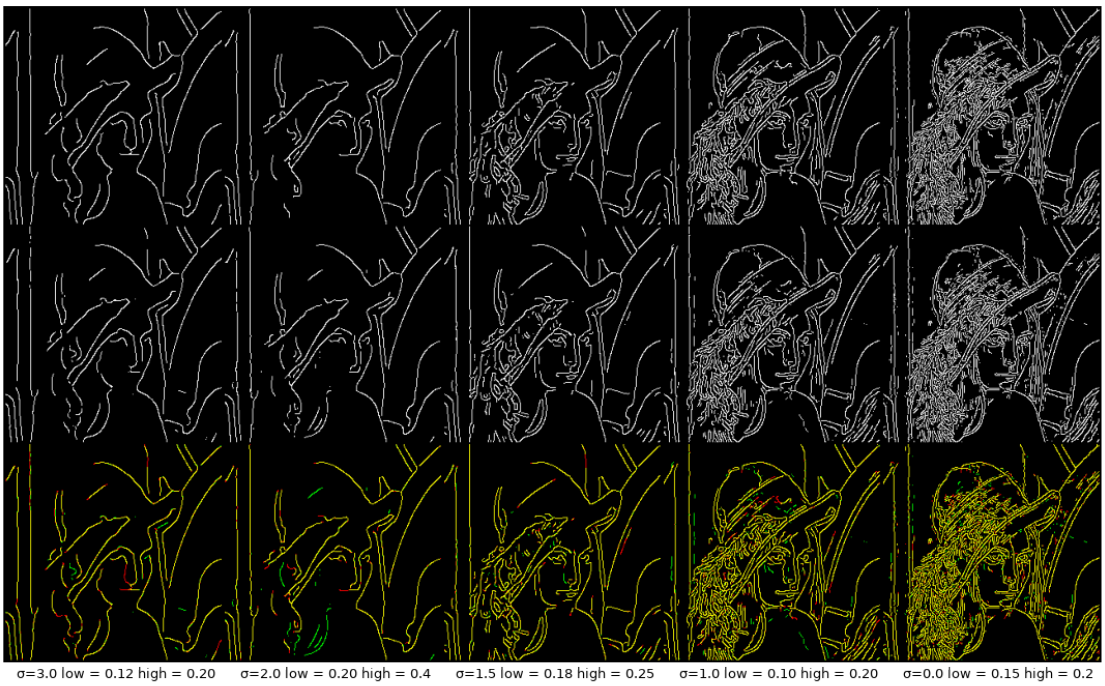

Fig. 4: Parameterized approximation of Canny edge detector (1-st row: Canny edge detector, 2-st row: neural network, 3-st row: pixel difference, where black, yellow, green, and red colours stand for true negative, true positive, false positive, and false negative pixels, respectively).

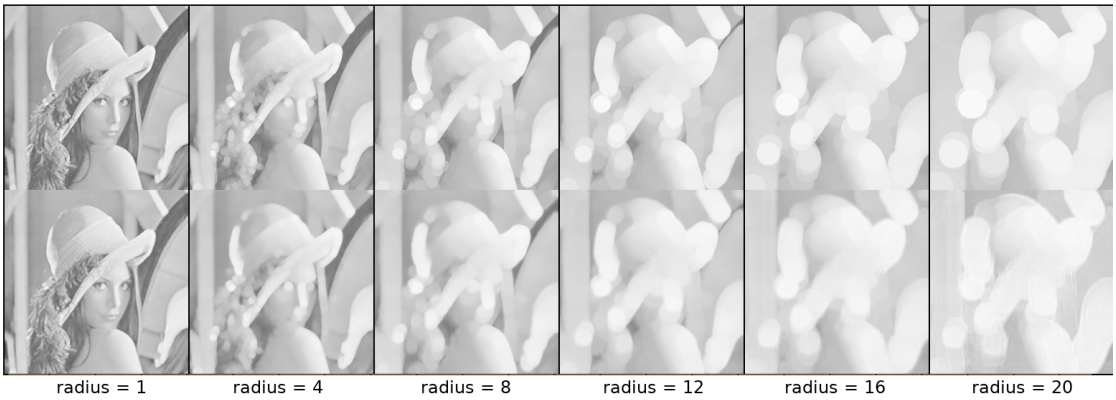

Fig. 5: Parameterized approximation of grayscale morphology with the disk structuring element (1-st row: morphological dilation, 2-st row: neural network).

ACKNOWLEDGMENT

This work is partially financially supported by Russian Foundation for Basic Research (project 18-29-26037).

REFERENCES

Andreeva, Elena; Vladimir Arlazarov; Aleksandr Gayer; Evgeniy Dorokhov; Aleksandr Sheshkus; and Oleg Slavin. 2019. "Document recognition method based on convolutional neural network invariant to 180 degree rotation angle." *ITiVS*, (4):87–93. DOI: 10.14357/20718632190408.

Canny, John. 1986. "A computational approach to edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698. doi: 10.1109/TPAMI.1986.4767851.

Chen, Yinpeng; Xiyang Dai; Mengchen Liu; Dongdong Chen; Lu Yuan; and Zicheng Liu. 2020. "Dynamic convolution: Attention over convolution kernels." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cheng, Xiaoqi; Junhua Sun; and Fuqiang Zhou. 2021. "A fully convolutional network-based tube contour detection method using multi-exposure images." *Sensors*, 21(12). ISSN 1424-8220. doi:10.3390/s21124095.

De Ridder, Dick; Robert Duin; Piet. W. Verbeek; and Lucas J. Van Vliet. 1999. "The applicability of neural networks to non-linear image processing." *Pattern Anal. Appl.*, 2(2):111–128. ISSN 1433-7541. doi: 10.1007/s100440050022.

Di Febbo, Paolo; Carlo Dal Mutto; Kinh Tieu; and Stefano Mattoccia. 2018. "Kcnn: Extremely-efficient hardware keypoint detection with a compact convolutional neural network." In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 795–7958. doi:10.1109/CVPRW.2018.00111.

Ellen, Jeffrey S.; Casey A. Graff; and Mark D. Ohman. 2019. "Improving plankton image classification using context metadata." *Limnology and Oceanography:*

Methods, 17(8):439–461. doi:10.1002/lom3.10324.

Erlygin, Leonid and Lev Teplyakov. 2021. "Improvement of a line segment detector based on a neural network by adding engineering features." *Sensory systems*, 35(1):50–54. DOI: 10.31857/S0235009221010042.

Fernández, Andrea; Carlos Delgado-Mata; and Ramiro Velázquez. 2011. "Training a single-layer perceptron for an approximate edge detection on a digital image." *Proceedings - 2011 Conference on Technologies and Applications of Artificial Intelligence, TAAI 2011*. doi: 10.1109/TAAI.2011.40.

Gessert, Nils; Maximilian Nielsen; Mohsin Shaikh; René Werner; and Alexander Schlaefer. 2020. "Skin lesion classification using ensembles of multi-resolution efficientnets with meta data." *MethodsX*, 7:100864. ISSN 2215-0161. doi:https://doi.org/10.1016/j.mex.2020.100864.

Hornik, Kurt; Maxwell Stinchcombe; and Halbert White. 1989. "Multilayer feedforward networks are universal approximators." *Neural Networks*, 2(5):359–366. ISSN 0893-6080. doi:10.1016/0893-6080(89)90020-8.

Jampani, Varun; Martin Kiefel; and Peter V. Gehler. 2016. "Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4452–4461.

Kang, Di; Debarun Dhar; and Antoni Chan. 2017. "Incorporating side information by adaptive convolution." In I. Guyon; U. V. Luxburg; S. Bengio; H. Wallach; R. Fergus; S. Vishwanathan; and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30 (Curran Associates, Inc., 2017).

Karnaushkin, Pavel and Maksim Sklyarenko. 2022. "Computer vision-based method of pre-alignment of a channel optical waveguide and a lensed fiber." *Computer Optics*, 46(1):71–82. Doi:10.18287/2412-6179-CO-919.

Li, Zewen; Fan Liu; Wenjie Yang; Shouheng Peng; and Jun Zhou. 2021. "A survey of convolutional neural networks: Analysis, applications, and prospects." *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. doi:10.1109/TNNLS.2021.3084827.

Limonova, Elena; Arseny Terekhin; Dmitry Nikolaev; and Vladimir Arlazarov. 2020. "Fast implementation of morphological filtering using ARM NEON extension." *arXiv preprint arXiv:2002.09474*.

Lobanov, Mikhail and Dmitry Sholomov. 2019. "On the acceleration of the convolutional neural network architecture based on resnet in the task of road scene objects recognition." *ITiVS*, 69(3):57–65. DOI: 10.14357/20718632190305.

Mckay, M.; Richard Beckman; and William Conover. 1979. "A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code." *Technometrics*, 21:239–245. doi: 10.1080/00401706.1979.10489755.

Mudiyanselage, Thosini Bamunu; Nipuna Senanayake; Chunyan Ji; Yi Pan; and Yanqing Zhang. 2021. "Covid-19 detection from chest x-ray and patient metadata using graph convolutional neural networks."

Panfilova, Ekaterina and Irina Kunina. 2020. "Using window hough transform for detecting elongated boundaries in an image." *Sensory systems*, 34(4):340–353. DOI: 10.31857/S0235009220030075.

Panfilova, Ekaterina; Oleg Shipitko; and Irina Kunina. 2021. "Fast hough transform-based road markings detection for autonomous vehicle." In *ICMV 2020*, volume 11605 (Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Washington 98227-0010 USA, 2021). DOI: 10.1117/12.2587615.

Tziolas, Nikolaos; Nikolaos Tsakiridis; Eyal Ben-Dor; John Theocharis; and George Zalidis. 2020. "Employing a multi-input deep convolutional neural network to derive soil clay content from a synergy of multi-temporal optical and radar imagery data." *Remote Sensing*, 12(9). ISSN 2072-4292. doi:10.3390/rs12091389.

Wei, Chih-Chiang. 2020. "Comparison of river basin water level forecasting methods: Sequential neural networks and multiple-input functional neural networks." *Remote Sensing*, 12(24). ISSN 2072-4292. doi:10.3390/rs12244172.

Yang, Brandon; Gabriel Bender; Quoc V. Le; and Jiquan Ngiam. 2020. "Condconv: Conditionally parameterized convolutions for efficient inference."

Yi, Kwang Moo; Eduard Trulls; Vincent Lepetit; and Pascal Fua. 2016. "Lift: Learned invariant feature transform." In Bastian Leibe; Jiri Matas; Nicu Sebe; and Max Welling, editors, *Computer Vision – ECCV 2016*, 467–483 (Springer International Publishing, Cham, 2016). ISBN 978-3-319-46466-4.

Yu, Xunbo; Hanyu Li; Xinzhu Sang; Xiwen Su; Xin Gao; Boyang Liu; Duo Chen; Yuedi Wang; and Binbin Yan. 2021. "Aberration correction based on a pre-correction convolutional neural network for light-field displays." *Opt. Express*, 29(7):11009–11020. doi: 10.1364/OE.419570.

Zheng, Zhen; Bingting Zha; Youshi Xuchen; Hailu Yuan; Yanliang Gao; and He Zhang. 2019. "Adaptive edge detection algorithm based on grey entropy theory and textural features." *IEEE Access*, PP:1–1. doi:10.1109/ACCESS.2019.2927655.

Zhou, Ding-Xuan. 2019. "Universality of deep convolutional neural networks." *Applied and Computational Harmonic Analysis*, 48. doi:10.1016/j.acha.2019.06.004.

Zhukovskiy, Aleksandr; Elena Limonova; and Dmitry Nikolaev. 2018. "Exact implementation of common image processing algorithms using fully convolutional networks." *Trudy ISA RAN*, 68(Special issue S1):108–116. DOI: 10.14357/20790279180512.

AUTHOR BIOGRAPHIES

OLGA CHAGANOVA was born in Orenburg, Russia. Currently she is a M.Sc. student in Applied Mathematics and Physics at Moscow Institute of Physics and Technology. Since 2021, she works as a junior research fellow at the Vision Systems Lab of the Institute for Information Transmission Problems. Her research interests include deep learning and computer vision. Her email address is o.chaganova@visillect.com.

ANTON GRIGORYEV was born in Petropavlovsk-Kamchatskiy, Russia. Having graduated from Moscow Institute of Physics and Technology, he has been developing industrial computer vision systems with the Vision Systems Lab at the Institute for Information Transmission Problems since 2010. His research interests are image processing and enhancement methods, autonomous robotics and software architecture. His e-mail address is me@ansgri.com.