Object segmentation approach in image processing

Narzullo Mamatov[1*], *Vohid* Fayziyev[1], *Malika* Jalelova[1], and *Boymirzo* Samijonov[2]

[1]Tashkent Institute of Irrigation and Agricultural Mechanization Engineers, National Research University, Tashkent, Uzbekistan
[2]Sejong University, South Korea, Seoul, Korea

Abstract. Image segmentation is a crucial and complex process in image processing, fundamental to object recognition. While neural network-based methods are widely used for segmentation, they require substantial resources and are vulnerable to noise and artifacts. This study addresses the need for improved segmentation approaches by proposing a novel four-step sequence with corresponding algorithms for object segmentation in images. The research methodology involves developing a systematic approach to image segmentation, implementing the proposed algorithms, and conducting computational experiments using three distinct image databases. The results of the proposed approaches are compared with those of the DeepLabV3+ Resnet50 model, a deep learning-based image segmentation technique. Our findings demonstrate that the proposed approaches outperform the deep learning model in segmenting untrained objects, while the latter excels only with trained objects. This research contributes to the field by offering a more versatile and robust segmentation method, potentially applicable to a wider range of image processing tasks without the need for extensive training data or computational resources. The study highlights the importance of developing adaptive segmentation techniques that can handle diverse object types efficiently.

1 Introduction

Image processing and analysis are two closely related image processing steps. Image analysis uses the results of processing to extract important features and image segmentation and extract meaningful information. It allows making conclusions and decisions based on the content of the image [1]. Image analysis includes such tasks as classification, localization, and segmentation of objects. Among them, image segmentation is considered one of the important tasks, and segmentation results have a fundamental impact on image analysis processes, such as object description and description, feature calculation, and even solving high-level tasks such as object classification. Therefore, image segmentation is a crucial process to simplify the definition, description, and visualization of regions of interest in any image.

The purpose of image segmentation is to:

* Corresponding author: m_narzullo@mail.ru

- distinguish objects by defining the boundaries of objects in images;
- to determine that each of the separated regions belongs to a certain class or category;
- simplify the structure of the image to simplify the analysis, highlight the important elements, and ignore the unimportant ones;
- ensuring high recognition accuracy;
- calculating the size and volume of objects.

Image segmentation, particularly medical image segmentation, is being used in clinical research to manage and monitor disease progression [2]. Also, segmented images are currently used in a variety of applications such as disease diagnosis, localization of pathology, tissue volume determination, treatment planning, computer-integrated surgery, and anatomical study.

In recent years, many studies have been carried out on the development of image segmentation methods. In particular, segmentation methods based on deep learning are studied in [3], where considerable work has been done by researchers on MRI and ultrasound image segmentation. However, in this work, the result was 96% accuracy when there was noise in the image. Also, new deep-learning approaches for ultrasound image segmentation are proposed. The authors mainly used convolutional neural networks (CNN) and recurrent neural networks (RNN), autoencoder machines, and hybrid approaches. Problems related to the performance of CNNs for the segmentation of microscopic images and their computation time are described in [4]. Research work on image segmentation without using neural networks has been done by many researchers. In particular, an image segmentation model using an area growth-based method for segmentation of Landsat (land satellite) images for land cover change detection was proposed in [5]. Also, several methods of image segmentation clustering [6], methods based on areas and contours, and neural networks [7] have been developed. However, not all of them always meet the specified requirements, for example, high-speed algorithms do not always allow to identification of particularly important non-local features of the image.

The most reliable segmentation of objects in images is performed by an expert. However, manual segmentation of images by an expert is not only a tedious and time-consuming method, but also requires the determination of structures piece by piece, causing many errors due to fatigue. Reliable automated segmentation methods are needed to provide accurate and consistent results [8]. However, the development of automatic image segmentation approaches is difficult because images are usually noisy, complex in nature, and rarely linearly simple. In addition, factors such as image quality, scene complexity, intensity uniformity, and artifacts may affect the result of segmentation algorithms. Therefore, despite years of research and many significant advances in technology, several problems remain that are difficult to solve. In particular, noise and artifacts that degrade image quality can have a strong negative impact on the performance of segmentation methods. Such problems are usually solved based on image preprocessing methods and algorithms.

Image pre-processing is an important quality in solving the problems of image segmentation, object recognition, and analysis by performing certain operations on the image, in particular, applying filters, contrast enhancement [9-10], noise reduction [11], and normalization. is a process that helps to obtain images. Based on the analysis of contrast enhancement literature and extensive experiments, it was found that histogram equalization, CLAHE (Contrast Limited Adaptive Histogram Equalization), and linear contrast algorithms are widely and effectively used in image contrast enhancement. There are well-known methods such as Weiner [12], Gaussian [13], median [14], anisotropic diffusion [15], bilateral [16], BM3D [17], NLM, and TV for removing noise in images. However, the lack of information about when and in what cases it is effective to use these methods of image processing, the lack of rules for their application, or the insufficient number of studies on the

formation of automation conditions indicate the need to develop new and improved approaches to image segmentation. Therefore, this work, it is aimed to formulate recommendations for applying a sequence of image processing steps and corresponding rules and algorithms for the segmentation of objects in images.

2 Methods

In this work, the steps shown in the figure below are performed sequentially to segment the objects in the images.

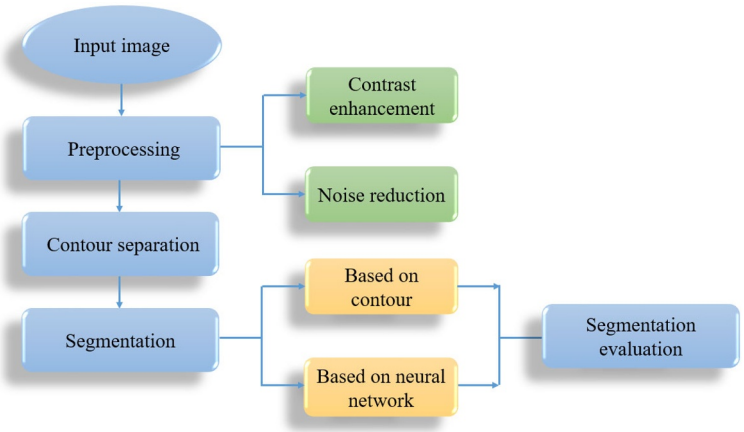

Fig.1. Image segmentation implementation sequence.

Stage 1. Images are captured through various imaging devices. Including medical imaging equipment (CT, MRI, X-ray, microscope), unmanned flying devices (drones), imaging devices (cameras), and others. Usually, the incoming image is not clean, but it is desirable to have it clean. In practice, it is very difficult to get a clean image and it is almost impossible to find. Therefore, it is required to improve the quality of the incoming image, and this is done by applying pre-processing algorithms to it.

Stage 2. The use of objective evaluation criteria is required to automate contrast enhancement during image preprocessing. In this case, A_j contrast enhancement algorithms were used for I_{gray} gray images, B contrast evaluation criterion was used to evaluate I_c^j output images, and the corresponding optimal algorithm or sequence of algorithms was determined. This is done in the following steps:

Step 1. Input $I_{gray}(x,y)$;

Step 2. Compute $B\left(I_{gray}(x,y)\right)$;

Step 3. $I_c^j = A_j\left(I_{gray(x,y)}\right), j=\overline{1,m}$;

Step 4. Compute $B\left(I_c^j(x,y)\right), j=\overline{1,m}$;

Step 5. If $B\left(I_{gray}(x,y)\right) < B\left(I_c^j(x,y)\right), j=\overline{1,m}$ then “Algorithm useful” else “Algorithm useless”;

Step 6. optimal_algorithm $= \max_q \left\{ \frac{1}{n} \sum_{k=1}^{n} \left| B^q \left(I_c(x,y) \right) - B^q \left(I_{gray}(x,y) \right) \right| \right\}, q = \overline{1,n}$;

Step 7. A_q optimal algorithm ва $\left\{ B_1, A_q \right\}$ optimal pair. Where m and $n-$ are the number of algorithms and images, respectively.

In experimental studies, algorithms such as histogram smoothing, CLAHE, linear contrast, and their various sequences were applied to the image, and their contrasts were changed. Experiments have shown that the histogram smoothing algorithm is optimal for the RMS value, and it is recommended to use the histogram smoothing and CLAHE algorithms sequentially when the RMS value is within the range of the image contrast, otherwise, do not change the image contrast.

In automating the preprocessing of color images, the image is divided into three color channels and each of them is processed separately, and then they are combined into a single image. In this case, the image contrast was evaluated using the GCF (global contrast factor) criterion, and the color image contrast enhancement was carried out based on the rule presented in the figure below.

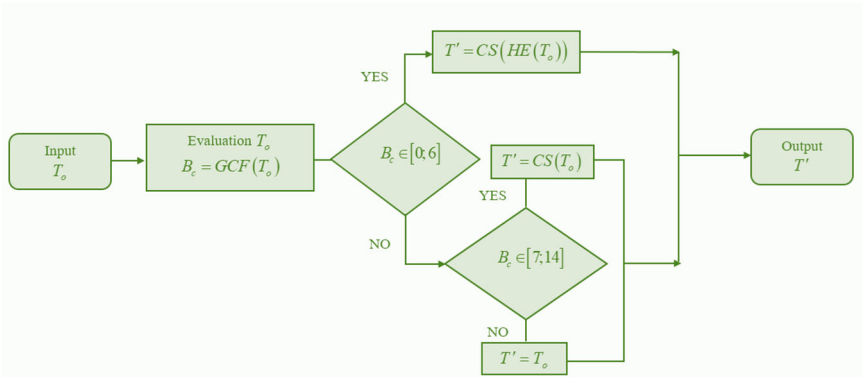

Fig.2. A block diagram of the contrast enhancement rule.

In Figure 2, T_0 is the original input image, T'-is the processed, that is, output image, HE – histogram equalization and CS – -linear contrast algorithms.

One of the main elements that degrade image quality is noise. Noise is the main focus of researchers in the field of image processing, as it is an urgent problem that hinders image quality presentation and has not been developed a clear solution. In many literatures, cases where only one or two types of noise are mixed into the images are studied. However, usually the image is not spoiled by only one or two types of noise, the combination of more types of noise due to various factors leads to a decrease in image quality.

Suppose we are given an original clean image I_{org} and $n_i, \left(i = \overline{1,k} \right)$ k noises.

Definition 1. An image I_{noisy} formed by adding k combinations of noise to the image I_{org} is called a mixed noisy image.

In the study, the case where $k = 3$ is analyzed, that is, the case where a combination of n_1, n_2, n_3 noises is added to the original image. Mixed noise consisting of combinations of n_1 – Gaussian, n_2 – salt and pepper, and n_3 – Poisson noise types, which are the main types of noise in the images, were studied. These noises are defined according to the sequence of their addition to the original image, and the following steps are taken to develop a rule for removing mixed noise from images:

Step 1. Mixed noisy images are generated:

$I_{GPS} = I_{org} + n_{GPS}$, $I_{GSP} = I_{org} + n_{GSP}$, $I_{PGS} = I_{org} + n_{PGS}$, $I_{PSG} = I_{org} + n_{PSG}$, $I_{SGP} = I_{org} + n_{SGP}$, $I_{SPG} = I_{org} + n_{SPG}$.

Step 2. The optimal filter for each type of noise is selected based on the literature review, and various combinations of filters are applied to the mixed noise image, and the resulting image is denoted as $\hat{I}$.

$$F_{GPS}(I^j_{noisy}), F_{GSP}(I^j_{noisy}), F_{PGS}(I^j_{noisy}), F_{PSG}(I^j_{noisy}), F_{SGP}(I^j_{noisy}), F_{SPG}(I^j_{noisy}),\ j = \overline{1,6}$$

where $I^1_{noisy} = I_{GPS}, I^2_{noisy} = I_{GSP}, I^3_{noisy} = I_{PGS}, I^4_{noisy} = I_{PSG}, I^5_{noisy} = I_{SGP}, I^6_{noisy} = I_{SPG}$.

$F_{GPS} : BM3D \to TV \to Median$, $F_{GSP} : BM3D \to Median \to TV$, $F_{PGS} : TV \to BM3D \to Median$, $F_{PSG} : TV \to Median \to BM3D$, $F_{SGP} : Median \to BM3D \to TV$, $F_{SPG} : Median \to TV \to BM3D$

Step 3. The quality of the output image is evaluated and the effectiveness of using the filter sequence is determined based on the following condition:

$$B\left(F\left(I_{noisy}\right)\right) < B\left(I_{org}\right)$$

where BRISQUE is taken as the evaluation criterion B.

Step 4. The filter sequence with the highest efficiency is determined by the condition in the third step, and it is used as the main F filter in the image processing rule.

As a result of the above steps, an image processing rule was formed and it is shown in the figure below.

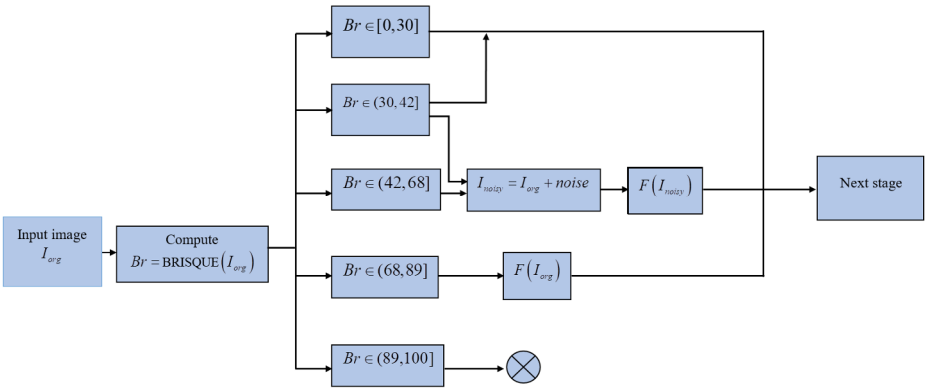

Fig. 3. Noise reduction rule according to the BRISQUE criterion.

Using the proposed rule can give good results in automating the image preprocessing process.

Stage 3. Contour segmentation is an important step after image preprocessing, and it allows preserving important image details by reducing the amount of processed data [18]. Therefore, the problem of object contour detection during image processing was studied in the work. To do this, the following sequence of steps was performed in the research to determine the contours of the image object:

Step 1. The original image T_o and the image T_o^k with its corresponding contour extracted by the expert are input;

Step 2. u_i contouring methods are applied to the original image:

$$T_i = u_i\left(T_o\right),\ i = \overline{1,m}$$

where m - is the number of filters used in contour separation.

Step 3. The effectiveness of the contour separation filters is evaluated based on the pixel compatibility of the original T_o and T_i filtered images:

$$B = \frac{\left|T_o^k \cap T_i\right|}{\left|T_o^k\right|} \cdot 100\%, \; i = \overline{1,m}$$

Step 4. The optimal filter is determined according to the value of B evaluation:

$$u_{opt} = \max_i \left\{B_i\right\}, i = \overline{1,m}$$

As a result of performing the above sequence of steps, the rule for determining image contours is formed. For this purpose, Roberts [19], Preuitt [20], Sobel [21], Robinson [22], LoG [23], DoG [24], Kenny [25], Kitchen-Malin [26], Scharr [27] , 11 filters such as Kayalli [28], Orhei [29] were obtained. As a result of experimental studies, a rule for applying the appropriate method of contour separation according to the BRISQUE criterion value was developed:

- If $0 < B_r < 21$, then Scharr filter is applied;
- If $21 \le B_r < 41$, then Kayalli filter is applied;
- If $41 \le B_r < 61$, then Sobel filter is applied or the image is sent back to the preprocessing;
- If $61 \le B_r < 81$, then Kitchen-Malin, DoG or LoG filter is applied or the image is sent back to the preprocessing;
- If $81 \le B_r < 100$, then methods are ineffective.

The above-mentioned rule for determining the contours of image objects is important in the automation of image analysis, and it is desirable to use the method presented in [30] to expand or fill the contour lines. These results allow professionals to obtain valuable information faster.

Once the contours of the image objects are defined, their thinning allows the collection of the most important information useful for the image analysis and recognition process [31], and many algorithms have been developed to achieve this goal. However, they have certain disadvantages [32]. Therefore, in the study, it is recommended to use a new algorithm of contour thinning, and this algorithm is called "semicircle and triangle algorithm". It was implemented on the basis of changing the steps of the existing Zhang-Suen algorithm [33] in the 1st and 2nd iterations, as well as adding conditions for checking the semicircle of the checked pixel p_1 (Fig. 4).

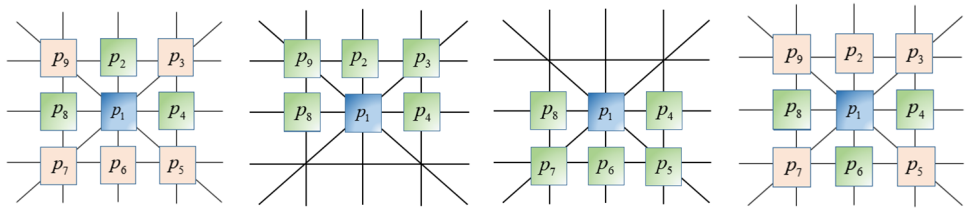

Fig.4. Cases of pixels to be checked in the semicircle and triangle algorithm a) $p_2 p_4 p_8$, b) $p_4 p_3 p_2 p_9 p_8$, c) $p_8 p_7 p_6 p_5 p_4$, d) $p_4 p_6 p_8$

The semicircle and triangle algorithm consists of the following two iteration steps:

Iteration 1.

Step 1. Initialization.

Let $P=\{(x_i,y_i)\}-$ be a set of pixels on the contour, where (x_i,y_i) represents the pixel coordinates on the contour.

Step 2. The pixel is checked.

For each (x_i,y_i) pixel of the contour, the number of 8-connected neighboring pixels is calculated:

$$k=\sum_{j=1}^{8} I\left((x_i,y_i)+d_j\right)$$

where d_j -is the displacement vector for the 8 connected neighbors, $I(x,y)$ is a function that is equal to 1 if pixel (x,y) is in the foreground, and 0 otherwise.

Step 3. The number of pixels that went from 0 to 1 among their octet-linked neighbors is determined.

$$A=\sum_{j=1}^{7} I\left((x_i,y_i)+d_j\right)\cdot I\left((x_i,y_i)+d_{j+1}\right)$$

Step 4. The conditions are checked to delete the pixel.

If $2\le k\le 6$, $A=1$,

$I\left((x_i,y_i)+d_2\right)\cdot I\left((x_i,y_i)+d_4\right)\cdot I\left((x_i,y_i)+d_8\right)=0$ and

$I\left((x_i,y_i)+d_4\right)\cdot I\left((x_i,y_i)+d_3\right)\cdot I\left((x_i,y_i)+d_2\right)\cdot I\left((x_i,y_i)+d_9\right)\cdot I\left((x_i,y_i)+d_8\right)=0$, then the (x_i,y_i) pixel is selected for deletion.

Step 5. The pixel is deleted.

Pixels that satisfy the above conditions are deleted.

Iteration 2.

Step 1. Initialization.

After the first iteration, the updated set of pixels is used.

Step 2. The pixel is checked.

The second step of the first iteration is repeated for a new set of pixels.

Step 3. The number of pixels that went from 0 to 1 among their octet-linked neighbors is determined. Using the updated data, the calculation of A is repeated for each point

Step 4. The conditions are checked to delete the pixel.

If $2\le k\le 7$, $A=1$

$I\left((x_i,y_i)+d_8\right)\cdot I\left((x_i,y_i)+d_7\right)\cdot I\left((x_i,y_i)+d_6\right)\cdot I\left((x_i,y_i)+d_5\right)\cdot I\left((x_i,y_i)+d_4\right)=0$

and $I\left((x_i,y_i)+d_4\right)\cdot I\left((x_i,y_i)+d_6\right)\cdot I\left((x_i,y_i)+d_8\right)=0$, then the (x_i,y_i) pixel is selected for deletion.

Step 5. The pixel is deleted.

Pixels that satisfy the above conditions are deleted.

After the contour lines of the image object are thinned, it becomes possible to perform segmentation based on these lines.

Stage 4. The segmentation step is performed in two different ways, using the contour image generated by the above three steps and using the DeepLabV3+ Resnet50 ready-made model based on deep learning for segmentation. The architecture information for the DeepLabV3 + Resnet50 model used in this is presented in Table 1.

Table 1. Details on DeepLabV3+ Resnet50 model architecture

Component	Operation	Output Tensor Size	Number of Parameters
Input Image	-	H x W x 3	-
ResNet-50 Backbone	-		-
Conv1	7x7 convolution, stride 2	H/2 x W/2 x 64	9,408
Max Pooling	3x3 max pooling, stride 2	H/4 x W/4 x 64	-
Conv2_x	1x1, 3x3, 1x1 convolutions (3 blocks)	H/4 x W/4 x 256	213,248
Conv3_x	1x1, 3x3, 1x1 convolutions (4 blocks)	H/8 x W/8 x 512	1,426,944
Conv4_x	1x1, 3x3, 1x1 convolutions (6 blocks)	H/16 x W/16 x 1024	8,512,512
Conv5_x	1x1, 3x3, 1x1 convolutions (3 blocks)	H/16 x W/16 x 2048	14,965,248
DeepLabV3+			
Atrous Spatial Pyramid Pooling (ASPP)	1x1, 3x3(6, 12, 18) convolutions, global pooling	H/16 x W/16 x 256	2,048,512
Upsample 1	Bilinear interpolation to H/4 x W/4	H/4 x W/4 x 256	-
Concatenation	Concatenate features from backbone	H/4 x W/4 x 512	-
Convolution	3x3 convolution	H/4 x W/4 x 256	1,179,904
Upsample 2	Bilinear interpolation to H x W	H x W x 256	-
Final Convolution	1x1 convolution	H x W x N (N — number of classes)	N * 256 + N
Output	Segmentation map	H x W x N	-

The DeepLabV3+ Resnet50 model is mainly implemented by training the following 4 classes with a total of 20 objects:

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: airplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

The mean squared error (MSE) was used to compare the results of the two segmentation approaches. Different image sets containing manually segmented images of the original image were used, and details of these sets are presented in the next Results section.

3 Results

In the computational experiments, the tests were performed on three images containing different objects. These are PascalVoc 2012 and BSDS 500 at www.kaggle.com, Weizman at www.wisdom.weizmann. An example of applying the contour separation segmentation approach using the processing rules proposed in this work is shown in Figure 5.

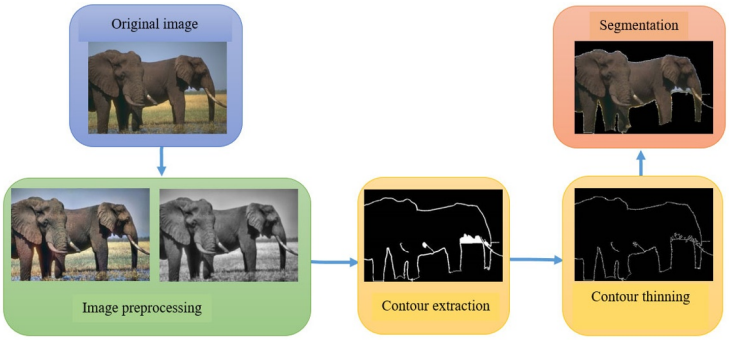

Fig. 5. An example of a contour-based segmentation approach.

Examples and MSE results of the application of DeeLabV3+ and contour approach to image segmentation on PascalVoc 2012, BSDS 500, and Weizman bases are shown in Figure 6.

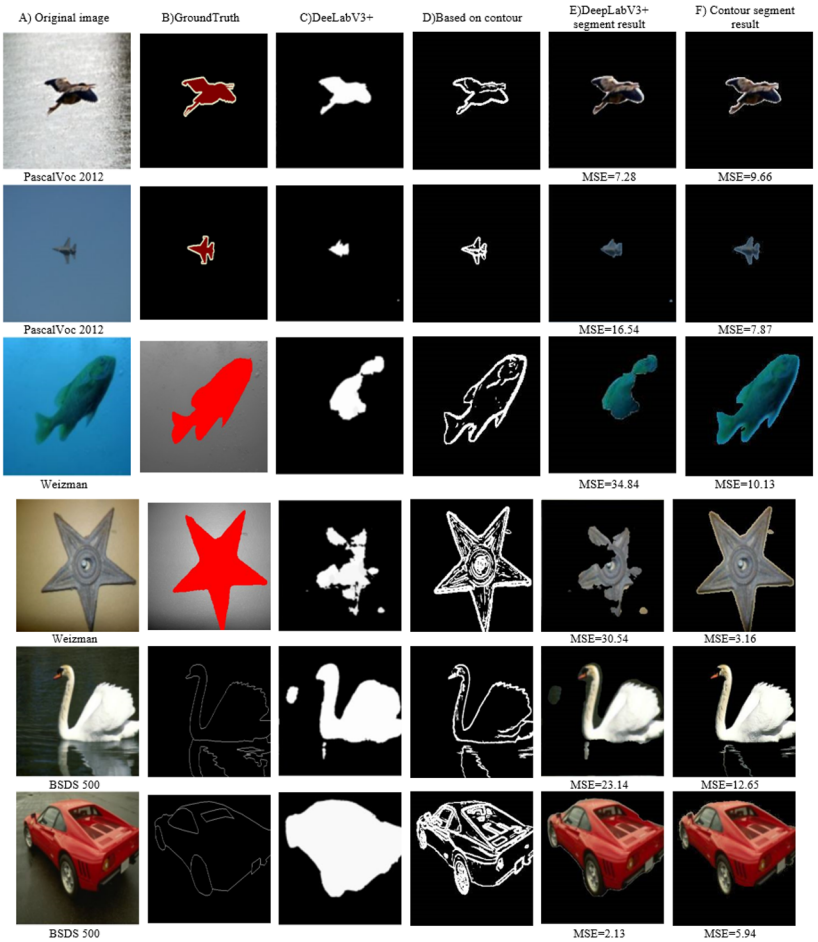

Fig. 6. Test results of the proposed contour segmentation approach and DeepLabV3+ ResNet50 model on different image bases.

4 Conclusion

In this work, the task of object segmentation for image processing and its goals are widely covered. Also. shortcomings of the existing segmentation methods were recognized and the need to formulate a new approach to image processing and a set of rules was justified. It was proposed to use the following rules and algorithms for image processing in the work:

- approaches to increase the contrast of gray and color images;
- image noise elimination rule;
- contour separation rule developed based on image quality indicator values;
- semicircle and triangle" algorithm for thinning contour lines.

Successive execution of the above four approaches for image segmentation allows for detailed separation of image object contours and relatively more accurate object segmentation. Various image databases were used in computational experiments and the results of the DeepLabV3+ ResNet50 ready model obtained for segmentation were compared with the results of the contour approach proposed in the work. As a result, DeepLabV3 + ResNet50 was found to segment the images with trained objects well, while the untrained objects could not segment the existing images well. The contour-based approach, on the other hand, showed good results in almost all image databases while preserving the alternative. Also, these methods do not require training of object images like neural network-based methods and do not take much time. In addition, neural network-based segmentation methods require parameter settings. Therefore, the application of the proposed contour approach to segmentation in the development of automated segmentation can be widely applied to the field of image processing.

References

1. X. Liu, L. Song, S. Liu, and Y. Zhang, Sustainability **13**, 1224 (2021)
2. Z. Han, M. Jian, and G.-G. Wang, Knowledge-Based Systems **253**, 109512 (2022)
3. S. Liu, Y. Wang, X. Yang, B. Lei, L. Liu, S. X. Li, D. Ni, and T. Wang, Engineering **5**, 261 (2019)
4. F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, IEEE Transactions on Neural Networks and Learning Systems **29**, 4550 (2017)
5. M. Brilliant, S. Y. Irianto, S. Karnila, and R. Z. A. Aziz, Journal of Physics Conference Series **1529**, 022066 (2020)
6. K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, Computerized Medical Imaging and Graphics **30**, 9 (2005)
7. V. Badrinarayanan, A. Kendall, and R. Cipolla, IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 2481 (2017)
8. H. P. A. Tjahyaningtijas, IOP Conference Series Materials Science and Engineering **336**, 012012 (2018)
9. X. Liu, H. Yang, and R. Li, IEEE Access **9**, 148925 (2021)
10. Luque-Chang, E. Cuevas, A. Chavarin, and M. Perez, IEEE Access **11**, 6060 (2023)
11. N. N. Hien, D. N. H. Thanh, U. Erkan, and J. M. R. S. Tavares, IEEE Access **10**, 71584 (2022)
12. P.S. Bindhya, J. Chitra and V.S. Bibin Raj, International Journal of Computer Sciences and Engineering 8(12), 1-9 (2017)
13. S. Chattopadhyay, Artificial Intelligence Evolution **87,** 82-106 (2022)

14. S. Bharati, T. Z. Khan, P. Podder, and N. Q. Hung, in *Studies in Systems, Decision and Control* (2020), pp. 49–66
15. G. Garg and M. Juneja, Multimedia Tools and Applications **78**, 12689 (2018)
16. D. Bhonsle, V. Chandra, and G. R. Sinha, International Journal of Image Graphics and Signal Processing **4**, 36 (2012)
17. S. Shrestha, Natural Gradient Methods: Perspectives, Efficient-Scalable Approximations, and Analysis arXiv https://arxiv.org/abs/2303.05473 (2014)
18. M. Mittal, A. Verma, I. Kaur, B. Kaur, M. Sharma, L. M. Goyal, S. Roy, and T.-H. Kim, IEEE Access **7**, 33240 (2019)
19. H. X. Gong and L. Hao, Journal of Chemical and Pharmaceutical Research **6**, 7 (2014).
20. L. Yang, X. Wu, D. Zhao, H. Li, and J. Zhai, An improved Prewitt algorithm for edge detection based on noised image, In *4th International Congress on Image and Signal Processing, Shanghai, China,* pp. 1197–1200 (2011). https://www.doi.org/10.1109/CISP.2011.6100495
21. O. Vincent and O. Folorunso, A Descriptive Algorithm for Sobel Image Edge Detection in: *Proceedings of Informing Science & IT Education Conference (InSITE) 2009* (2009)
22. S. Chakraborty, S. H. Shaikh, A. Chakrabarti, and R. Ghosh, Multimedia Tools and Applications **81**, 33459 (2022)
23. J. Hu, X. Tong, Q. Xie, and L. Li, in *Lecture Notes in Computer Science* (2016), pp. 474–483
24. H. Winnemöller, J. E. Kyprianidis, and S. C. Olsen, Computers & Graphics **36**, 740 (2012)
25. J. Wittmann and G. Herl, E-Journal of Nondestructive Testing **28**, (2023) DOI:10.58286/27751
26. L. J. Kitchen and J. A. Malin, Computer Vision Graphics and Image Processing **47**, 243 (1989)
27. N. Bibi and H. Dawood, Arabian Journal for Science and Engineering **49(3),** 3435-3451 (2023)
28. E. Kawalec-Latała, Studia Geotechnica Et Mechanica **36**, 29 (2014)
29. Orhei, S. Vert, and R. Vasiu, A Novel Edge Detection Operator for Identifying Buildings in Augmented Reality Applications. In *International Conference on Information and Software Technologies* (2020) pp. 208-219
30. F. Gao, M. Wang, Y. Cai, and S. Lu, Pattern Analysis and Applications **22**, 1123 (2018)
31. N.Mamatov, A. Samijonov, K. Yerejepov, I. Narzullaev, and B. Samijonov, The Eurasian Journal of Mathematical Theory and Computer Science **4**, 3 (2024)
32. M. Gramblička and J. Vaský, Journal of Theoretical and Applied Information Technology **94**, 2 (2016)
33. M. Nazarkevych, S. Dmytruk, V. Hrytsyk, O. Vozna, A. Kuza, O. Shevchuk, Y. Voznyi, I. Maslanych, and V. Sheketa, International Journal of Sensors Wireless Communications and Control **11**, 542 (2020)