Journal of Universal Science Research
ISSN: **2181-4570 2024, 2**(3), 7–19

The Canny Algorithm Implementation for Obtaining the Object Contour in a Mobile Robot's Workspace in Real Time

Vladyslav Yevsieiev[1], Svitlana Maksymova[1], Amer Abu-Jassar[2]

[1]Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2]Faculty of Information Technology, Department of Computer Science, Ajloun National University, Ajloun, Jordan

Abstract:

The article is devoted to the Canny algorithm implementation for obtaining the objects contours in mobile robots' workspace in real time. The paper presents the mathematical foundations of the algorithm, including all key stages: from image pre-processing up to the Canny operator application. The article main focus is the algorithm integration into the mobile robot system and its adaptation to the dynamic conditions of the workspace. The developed program in the Python programming language using the PyCharm development environment demonstrates the high performance of the algorithm in real time. A series of experiments has confirmed that the average video stream processing speed fluctuates in a narrow range from 1000.07 to 1002.70 frames per second.

Key words: Industry 5.0, Computer Vision Systems, Mobile Robots, Work zone

Introduction

In today's Industry 5.0 era, where production systems strive for maximum integration and interaction, the implementation of advanced technologies plays a key role in ensuring the efficiency and safety of production processes. One important direction is the use of mobile robots in workspaces, providing the ability to autonomously move and perform tasks in various industrial environments [1]-[11]. In this context, the design and implementation of image processing algorithms [12]-[26], such as the Canny algorithm, become key tools for ensuring smooth functioning of mobile robots in real time. Here you can use various approaches to construct and analyze such systems [27]-[30].

Computer vision systems play an essential role in the efficient navigation of mobile robots within work areas by providing them with the ability to perceive and analyze their environment. In this context, the use of the Canny algorithm to extract the objects contour becomes an important tool that allows mobile robots to accurately determine the boundaries of obstacles and elements of the work area. This not only improves safety by preventing collisions and accidents, but also ensures mobile robots can navigate accurately in dynamic manufacturing environments.

So, research and software implementation of the Canny algorithm to obtain the object contour of an in the mobile robot's workspace in real time will attract the attention of developers and engineers working in the field of Industry 5.0 and robotics. Research in this direction will not only be an important contribution to the development of computer vision systems for mobile robots, but will also contribute to increasing the efficiency and competitiveness of production processes, combining the principles of automation and safety.

Related works

There are a lot of research works devoted to edge detection. Let us analyze only a small part of recent ones.

Let us begin from paper [31]. This work explores and compares the plethora of metrics for the performance evaluation of object-detection algorithms. This work reviews the most used metrics for object detection detaching their differences, applications, and main concepts. It also proposes a standard implementation that can be used as a benchmark among different datasets with minimum adaptation on the annotation files. The article [32] proposes a new real-time small object detection algorithm based on YOLOv3, which improves the small object detection accuracy by using feature maps of a shallower layer containing more fine-grained information for location prediction; fusing local and global features of shallow and deep feature maps in Feature Pyramid Network to enhance the ability to extract more representative features; assigning weights to output features of Feature Pyramid Network and fusing them adaptively; and improving the excitation layer in Squeeze-and-Excitation attention mechanism to adjust the feature responses of each channel more precisely.

The authors in [33] use a depth residual network to replace VGG16 for image feature extraction so we can obtain deeper disease features. In the study [34] for the small objects in the street researchers first conduct the up-sampling in the feature map obtained by 8x down-sampling, and then fused it with the feature map obtained by 4x down-sampling. For the images in the extreme climate, we use the sharpness algorithm based on secondary blur (ReBlur) to characterize the image blurriness, where the blurred images will be restored a dark channel prior algorithm.

Scientists [35] propose an improved Mask R-CNN-based method: the ResNet Group Cascade (RGC) Mask R-CNN because the detection performance of Mask Region-Convolution Neural Network (R-CNN) based methods deteriorates when samples are reduced. Zhang, J., & et al. in [36] propose a small object detection method in UAV images as an improved YOLOv5-based algorithm. Paper [37] carries out experiments on three versions of popular YOLO models such as yolov3, yolov4, and yolov5 (yolov5l, yolov5m, yolov5s, yolov5x. Results showed that the yolov4 model is higher than the yolov3 model in terms of mAP values, but slightly lower in terms of speed, while the yolov5 series model is better than the yolov3 and yolov4 models both in terms of mAP values and speed. In [38] a method of multi-block Single Shot MultiBox Detector (SSD) based on small object detection is proposed to the railway scene of unmanned aerial vehicle surveillance.

For improving accuracy of detecting small objects Lim, J. S. and co-authors [39] propose an object detection method using context. The proposed method uses additional features from different layers as context by concatenating multi-scale features. They also propose object detection with attention mechanism which can focus on the object in image, and it can include contextual information from target layer. The research work by Elhoseny, M. [40] introduces a new multi-object detection and tracking methodology. The proposed method uses an optimal Kalman filtering technique to track the moving objects in video frames.

The Canny algorithm implementation for obtaining the object contour in real time

The Canny algorithm is a method for detecting edges in an image developed by John Canny in 1986. It is based on several key steps. First,

the image is smoothed with a Gaussian filter to reduce noise. The image gradients are then calculated using Sobel operators in the horizontal and vertical directions. The Canny algorithm is an efficient edge extractor that maintains high edge sensitivity and minimizes the impact of noise on the image. This makes it widely used in computer vision for tasks such as object recognition, moving object tracking, and edge detection. A general view of implementations of the Canny algorithm for obtaining the contour of an object in real time from a camera is presented in Figure 1.

For the Canny algorithm software implementation to obtain the object contour in real time, we will carry out a mathematical description of the algorithm basic steps (Fig. 1). Image smoothing is based on the Gaussian filter use to smooth out noise and prepare the image for further processing, a general view of which is presented below:

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/(2\sigma^2)} \quad (1)$$

$G(x,\mathrm{y})$ – Gaussian filter kernel, which is defined by a two-dimensional Gaussian distribution; this kernel is a two-dimensional function that determines the weight of each pixel during convolution;

x,y – pixel coordinates in the image;

σ – a parameter that determines the standard deviation (blur) of the Gaussian Function; the larger σ, the more blurred the image;

$\frac{1}{2\pi\sigma^2}$ – a normalizing factor that is used to normalize the weight values in the kernel; it ensures that the sum of all values in the kernel is equal to 1, which preserves the brightness of the image;

$e^{-(x^2+y^2)/(2\sigma^2)}$ – exponential part of the Gaussian function, where e is the base of the natural logarithmж this part determines the weight of each pixel in the kernel according to the distance from the central pixel; the further a pixel is from the center, the less its weight.

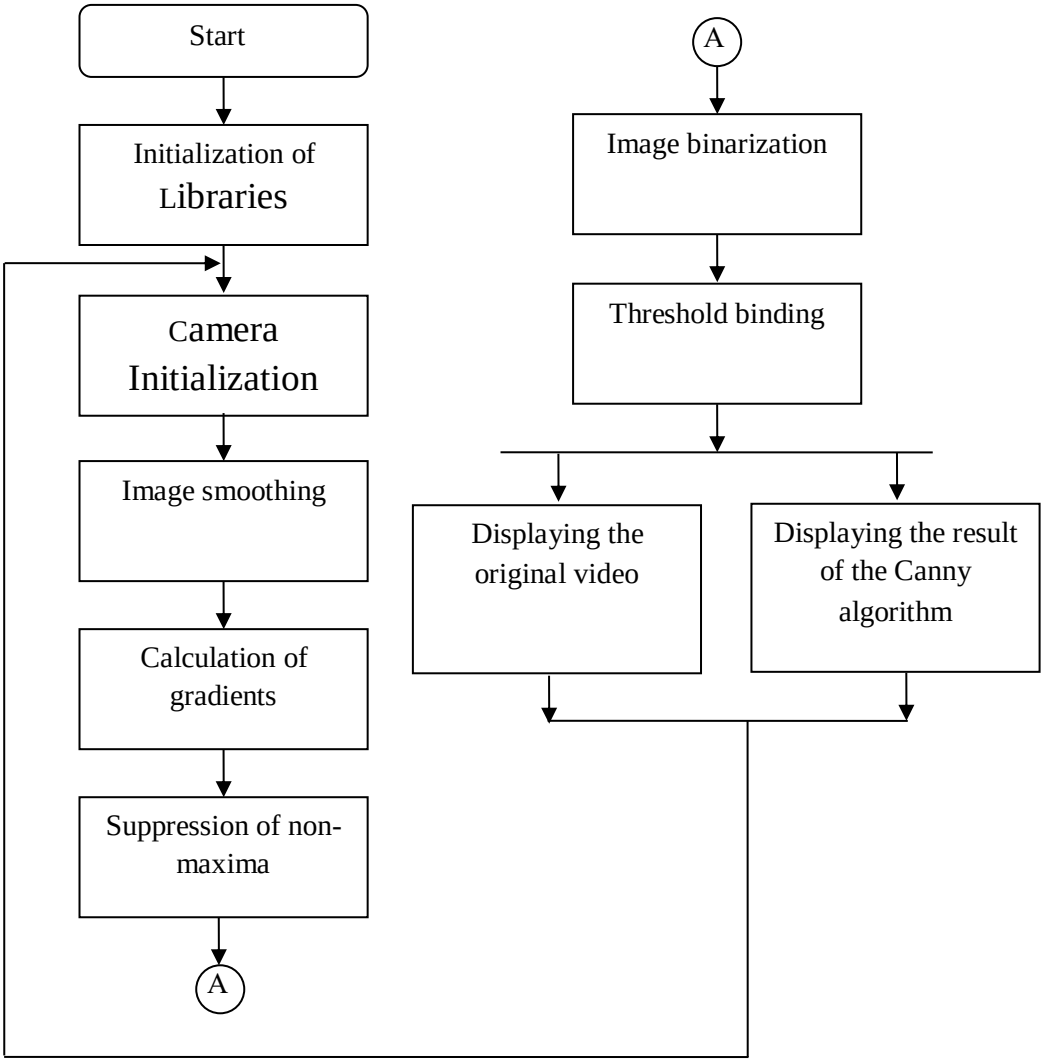


Figure 1: General view of the Canny algorithm implementation for obtaining the object contour in real time from a camera

The next step is to calculate the gradient using the Sobel operator. Let us denote by G_x – the kernel matrix for the horizontal Sobel operator, and by G_y – the kernel matrix for the vertical Sobel operator, where x, y are the pixel coordinates in the image. Hence the Sobel operator for this case can be represented as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2}$$

Sobel operators calculate the gradient of an image in the x and y directions, and their combination allows us to determine the amplitude of the gradient and the direction of change in intensity at each pixel. This ensures that the edges and boundaries of objects in the image are detected.

To determine local maxima, a window (usually 3x3) around each pixel is used. If the gradient amplitude in the current pixel ($M(x,y)$) is greater than in neighboring pixels located in the direction of the gradient ($\theta(x,y)$), then the current pixel is considered a local maximum.

The gradient amplitude (M) in each pixel is calculated by next formula:

$$M(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3}$$

G_x – kernel matrix for the horizontal Sobel operator;

G_y – kernel matrix for the vertical Sobel operator;

x, y – pixel coordinates in the image.

Gradient direction (θ)in each pixel is calculated by next formula:

$$\theta(x,y) = arctg\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \tag{4}$$

Hence the local maximum can be represented as follows:

$$M(x,y) > M(x \pm 1, y \pm 1) \rightarrow \theta(x,y) \tag{5}$$

The non-maxima suppression process helps to preserve only local maxima in the gradient direction, which allows the fine edges of objects in the image to be detected.

The next step is to binarize the resulting image. Binarization is the process of converting an image into a format that represents each pixel as

one of two possible values: black or white. In this case, binarization can be represented as follows:

$$E(x,y)=\begin{cases}1, if M(x,y)>T_{upper}\\ 0, if M(x,y)<T_{lower}\\ \text{Marked for linking, if } T_{lower}<M(x,y)<T_{upper}\end{cases} \quad (6)$$

$M(x,y)$ – gradient amplitude;

T_{upper} – upper threshold;

T_{lower} – lower threshold.

The last step is to link the boundaries, in this case we will use the hysteresis linking method to connect strong and weak boundary pixels and form contour curves.

In the context of mobile robots, the Canny algorithm implementation to camera video streaming allows robots to effectively extract and analyze object boundaries in real time, which is an important component for navigation and safety in the workspace.

Software implementation and experiments

To check the correctness of the reasoning, we will develop a program in Python in the PyCharm 2022.2.3 (Professional Edition) development environment. Let us give an example of software implementation of the above described mathematical expressions.

```
# Open video stream from camera (usually 0 for built-in camera)
cap = cv2.VideoCapture(0)
```

This piece of code is responsible for opening a video stream from a camera using the OpenCV library. Thus, after executing this piece of code, the cap variable becomes associated with the video stream from the camera, and makes it possible to use it for subsequent frame capture, processing and display.

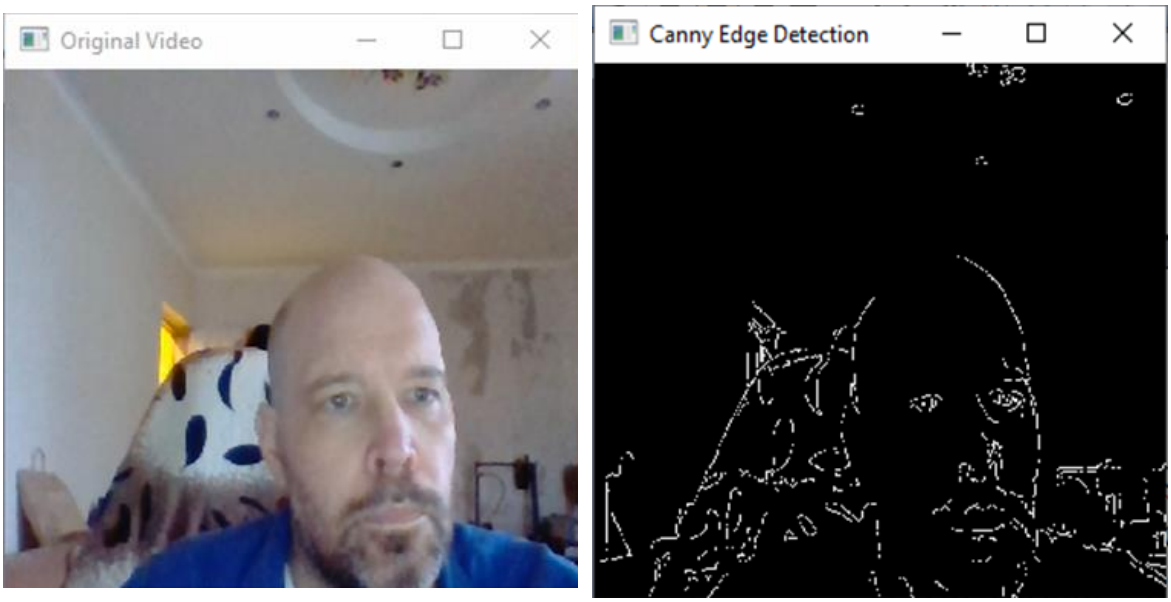
```
# Convert an image to grayscale
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

This piece of code converts a color image in BGR (Blue, Green, Red) format to a grayscale image. The process is carried out using the cvtColor (color conversion) function from the OpenCV library.

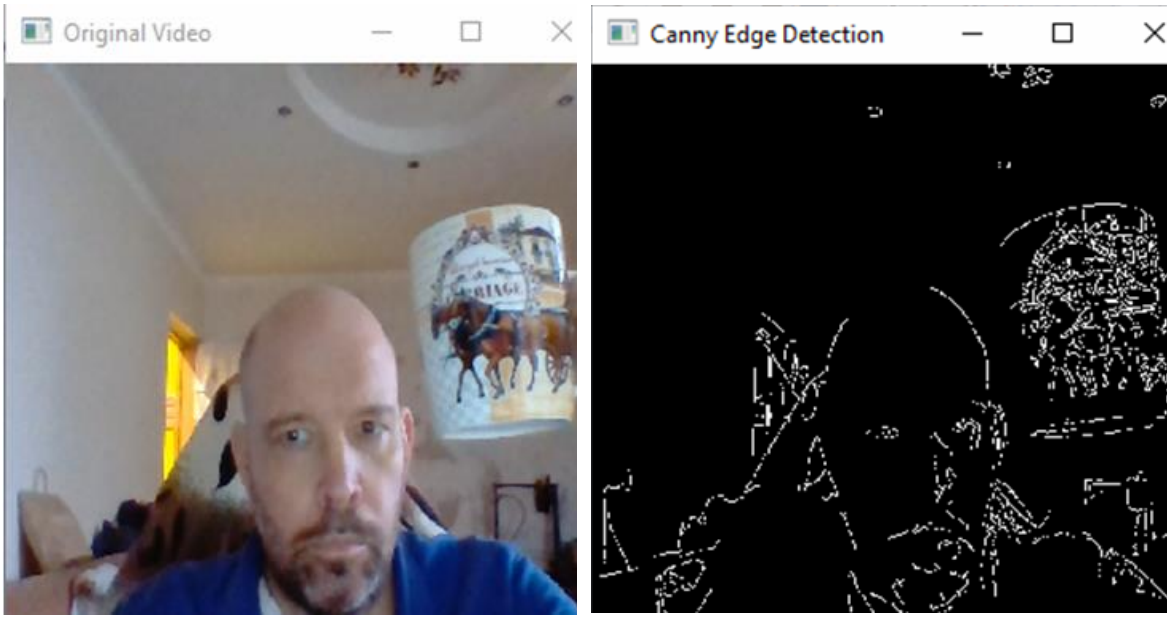
```
# Applying the Canny algorithm
    edges = cv2.Canny(gray, 50, 150)
```

This code snippet applies the Canny algorithm to a grayscale image. The Canny algorithm is used to detect edges in an image. So this code applies the Canny algorithm to a grayscale image and produces a binary image where the boundaries of objects are detected. Selection of threshold parameters allows you to control the sensitivity of the algorithm and determine which boundaries will be detected.

The following hardware was used for research: CPU Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, RAM 16 Gb, GPU NVideo GeForce GTX 1660Ti (Ram 8Gb), Web-camera HD WebCam, OS Windows 10 Pro (Version 22H2). A program for obtaining the contour of an object in real time from a camera was developed in the PyCharm 2022.2.3 (Professional Edition) environment in Python. The results of the program are presented in Figure 2.



a) b)



c) d)

a),c) – Original Video; b), d) – Canny Edge Detection

Figure 2: Results of the program for obtaining the object contour in real time

During the experiment of obtaining the object contour in real time from the camera, the processing speed indicators for all cases were also measured, which are presented in Figure 2; the average processing speed fluctuated in the range of 1000.07 – 1002.70 frames per second.

Conclusion

In a study devoted to the Canny algorithm implementation for extracting the object contour in the mobile robot workspace in real time, the following conclusions were reached.

First, the presented Canny algorithm is successfully integrated into the mobile robot system, providing accurate detection of object boundaries in a dynamic work area. The mathematical apparatus that describes the stages of the algorithm provides an understanding of its fundamentals and facilitates effective implementation.

The program, developed in the Python programming language using the PyCharm development environment, demonstrates the functionality of the algorithm in real time. The experiments carried out confirm the effectiveness of the proposed method, where the video stream processing speed fluctuates in a narrow range from 1000.07 to 1002.70 frames per

second. This indicates the high performance of the algorithm, which is a key indicator for mobile robots in real-time conditions.

Overall, the proposed implementation of the Canny algorithm is an effective tool for image processing in the context of mobile robots, providing accurate extraction of object contours and being implemented in a system with high video processing speed. This research contributes to the development of computer vision systems for mobile robots in Industry 5.0.

References:

1. Borysov, H., & et al. (2023). Parameters for Mobile Robot Kinematic Model Development Determination. Multidisciplinary Journal of Science and Technology, 3(4), 85-91.
2. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.
3. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.
4. Yevsieiev, V., & et al. (2022). A Robotic Prosthetic a Control System and a Structural Diagram Development. In Collection of scientific papers «ΛΌГОΣ», Zurich, 113-114.
5. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. SSRG International Journal of Engineering Trends and Technology, 69(7), 38-44.
6. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural Networks As A Tool For Pattern Recognition of Fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.
7. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.
8. Sotnik, S., & et al.. (2022). Analysis of Existing Infliences in Formation of Mobile Robots Trajectory. International Journal of Academic Information Systems Research, 6(1), 13-20.
9. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free

Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.
10. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 20, e37ije1673.
11. Abu-Jassar, A. T., Attar, H., Lyashenko, V., Amer, A., Sotnik, S., & Solyman, A. (2023). Access control to robotic systems based on biometric: the generalized model and its practical implementation. International Journal of Intelligent Engineering and Systems, 16(5), 313-328.
12. Akopov, M., & et al. (2023). Choosing a Camera for 3D Mapping. Journal of Universal Science Research, 1(11), 28-38.
13. Yevsieiev, V., & et al. (2024). Using Contouring Algorithms to Select Objects in the Robots' Workspace. Technical Science Research In Uzbekistan, 2(2), 32-42.
14. Lyashenko V., & et al. (2023). Automated Monitoring and Visualization System in Production. Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.
15. Yevsieiev, V., & et al. (2024). Active Contours Method Implementation for Objects Selection in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(2), 135-145.
16. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical Science Research In Uzbekistan,2(2), 115-124.
17. Lyashenko, V. V., Lyubchenko, V. A., Ahmad, M. A., Khan, A., & Kobylin, O. A. (2016). The Methodology of Image Processing in the Study of the Properties of Fiber as a Reinforcing Agent in Polymer Compositions. International Journal of Advanced Research in Computer Science, 7(1), 15-18.
18. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО "БизнесИнформ, 112.
19. Lyashenko, V., & et al.. (2021). Wavelet ideology as a universal tool for data processing and analysis: some application examples. International Journal of Academic Information Systems Research (IJAISR), 5(9), 25-30.
20. Deineko, Zh., & et al.. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific-practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, pp. 389-394.

21. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.
22. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.
23. Mousavi, S. M. H., Lyashenko, V., & Prasath, S. (2019). Analysis of a robust edge detection system in different color spaces using color and depth images. Компьютерная оптика, 43(4), 632-646.
24. Uchqun o‘g‘li, B. S., Valentin, L., & Vyacheslav, L. (2023). Pre-processing of digital images to improve the efficiency of liver fat analysis. Multidisciplinary Journal of Science and Technology, 3(1), 107-114.
25. Drugarin, C. V. A., Lyashenko, V. V., Mbunwe, M. J., & Ahmad, M. A. (2018). Pre-processing of Images as a Source of Additional Information for Image of the Natural Polymer Composites. Analele Universitatii'Eftimie Murgu', 25(2).
26. Lyashenko, V. V., & Babker, A. (2017). Using of Color Model and Contrast Variation in Wavelet Ideology for Study Megaloblastic Anemia Cells. Open Journal of Blood Diseases, 7(03), 86-102.
27. Lyashenko, V., Ahmad, M. A., Sotnik, S., Deineko, Z., & Khan, A. (2018). Defects of communication pipes from plastic in modern civil engineering. International Journal of Mechanical and Production Engineering Research and Development, 8(1), 253-262.
28. Sotnik, S., Matarneh, R., & Lyashenko, V. (2017). System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, 8(9), 378-390.
29. Dadkhah, M., Lyashenko, V. V., Deineko, Z. V., Shamshirband, S., & Jazi, M. D. (2019). Methodology of wavelet analysis in research of dynamics of phishing attacks. International Journal of Advanced Intelligence Paradigms, 12(3-4), 220-238.
30. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.
31. Padilla, R., & et al. (2020). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP), IEEE, 237-242.

32. Sun, W., & et al. (2021). RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. Applied Intelligence, 1-16.
33. Zhang, Y., & et al. (2020). Deep learning-based object detection improvement for tomato disease. IEEE access, 8, 56607-56614.
34. Zhu, D., & et al. (2021). Object detection in complex road scenarios: improved YOLOv4-tiny algorithm. In 2021 2nd Information Communication Technologies Conference (ICTC), IEEE, 75-80.
35. Wu, M., & et al. (2020). Object detection based on RGC mask R-CNN. IET Image Processing, 14(8), 1502-1508.
36. Zhang, J., & et al. (2023). Small object detection in UAV image based on improved YOLOv5. Systems Science & Control Engineering, 11(1), 2247082.
37. Liu, K., & et al. (2021). Performance validation of YOLO variants for object detection. In Proceedings of the 2021 International Conference on bioinformatics and intelligent Computing, 239-243).
38. Yundong, L. & et al. (2020). Multi-block SSD based on small object detection for UAV railway scene surveillance. Chinese Journal of Aeronautics, 33(6), 1747-1755.
39. Lim, J. S., & et al. (2021). Small object detection using context and attention. In 2021 international Conference on Artificial intelligence in information and Communication (ICAIIC), IEEE, 181-186.
40. Elhoseny, M. (2020). Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. Circuits, Systems, and Signal Processing, 39, 611-630.