

A YOLO-based Method for Object Contour Detection and Recognition in Video Sequences

Mariia Nazarkevych¹, Maryna Kostiak¹, Nazar Oleksiv¹, Victoria Vysotska¹, and Andrii-Taras Shvahuliak²

¹ Lviv Polytechnic National University, 12 Stepan Bandera str., Lviv, 79013, Ukraine

² Lviv Ivan Franko National University, 1 Universytetska str., Lviv, 79000, Ukraine

Abstract

A method for recognizing the contours of objects in a video data stream is proposed. The data will be uploaded using the video camera. Objects will be recognized in real-time. We will use YOLO—a method of identification and recognition of objects in real-time. Recognized objects will be recorded in a video sequence showing the contours of the objects. The approach proposed in the project reasonably synthesizes methods of artificial intelligence, theories of computer vision on the one hand, and pattern recognition on the other; it makes it possible to obtain control influences and mathematical functions for decision-making at every moment with the possibility of analyzing the influence of external factors and forecasting the flow of processes and refers to the fundamental problems of mathematical modeling of real processes. The installation of the neural network is shown in detail. The characteristics of the neural network and its capabilities are shown. Approaches to computer vision for object extraction are shown. Well-known methods are methods of expanding areas, methods based on clustering, contour selection, and methods using a histogram. The work envisages building a system for rapid identification of combat vehicles based on the latest image filtering methods developed using deep learning methods. The time spent on machine identification will be 10–20% shorter, thanks to the developed new information technology for detecting objects in conditions of rapidly changing information.

Keywords

Artificial intelligence, tracking, selection of objects, image recognition, YOLO, segmentation.

1. Introduction

Video surveillance is a common means of solving problems related to security and event monitoring [1–3]. One of the main tasks arising from video surveillance is detection [4], tracking [5] and identification [6] of moving objects. Video cameras are near us and record data about us. Therefore, there is a need to recognize data and objects. To recognize the data, you need to go through the stage of pre-processing, that is, improving the visual quality—increasing the contrast, distinguishing the boundaries, removing blurring, and filtering. Then there is an operation for the

preparation of graphic images—selection of objects, segmentation, and selection of contours.

Tracking is determining the location of a moving object [7] or several objects over time using a video camera (Fig. 1). The algorithm analyzes video frames and outputs the position of moving objects relative to the frame.

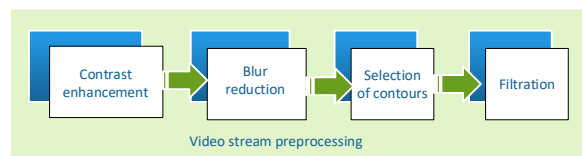
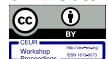


Figure 1: Scheme of preprocessing of a video sequence with object capture

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL : mariia.a.nazarkevych@lpnu.ua (M. Nazarkevych); maryna.y.kostiak@lpnu.ua (M. Kostiak); nazar.oleksiv.mnsa.2020@lpnu.ua (N. Oleksiv); victoria.a.vysotska@lpnu.ua (V. Vysotska); andrii-taras.shvahuliak@lnu.edu.ua (A.-T. Shvahuliak)
ORCID: 0000-0002-6528-9867 (M. Nazarkevych); 0000-0002-6667-7693 (M. Kostiak); 0000-0001-7821-3522 (N. Oleksiv); 0000-0001-6417-3689 (V. Vysotska); 0009-0002-0319-1909 (A.-T. Shvahuliak)



© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

The main tracking problem is matching the positions of the target object in successive frames, especially if the object is moving fast compared to the frame rate. Thus, tracking systems usually use a movement model [8], how the image of the target object can change during various movements (Fig. 2).

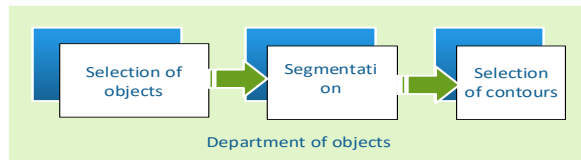


Figure 2: Scheme of separation of objects

Examples of such simple movement patterns are flat object tracking—affine transformation or object image homography [9].

The target can be a rigid three-dimensional object, the motion model determines the appearance depending on its position in space and orientation.

For video compression, keyframes are divided into macroblocks. A motion model is a burst of keyframes where each macroblock is transformed using a motion vector.

The image of a deformable object can be covered with a grid and the movement of the object is determined by the position of the vertices of this grid.

When an object is to be searched and matched against a given one, a new set of key points is extracted into the test image the two sets are matched and a similarity score is calculated.

2. Review of Literature

2.1. Object Selection

Before selecting an object from a video stream, there are pixel-by-pixel, block-by-block, and methods based on energy functional minimization [10] (Fig. 3).

Pixel-by-pixel methods of object selection process all points of the image. These methods are highly accurate, but they are sensitive to noise.

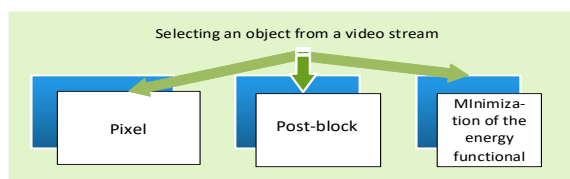


Figure 3: Methods of selecting an object from a video stream

Grayscale methods perform segmentation—dividing a digital image into several sets of pixels [11]. Image segmentation is commonly used to highlight objects and boundaries. More precisely, image segmentation is the process of assigning such labels to each pixel of an image so that pixels with the same labels share visual characteristics.

Block-based methods do not process individual pixels [12], but groups of pixels combined into blocks. If the block contains a boundary, then in such areas the boundary of the object is determined inaccurately [13, 14].

The disadvantage of methods [15] based on the energy function is the low speed of operation.

2.2. Methods of Expanding Regions

The methods of this group are based on the use of local features of the image [16]. The idea of the region expansion method is to analyze first the starting point, then its neighboring points according to the criterion of homogeneity of the analyzed points into one or another group. In more effective variants of the method, the starting point is not individual pixels, but the division of the image into several small areas. Each region is then checked for homogeneity, and if the result of the test is negative, the corresponding area is divided into smaller sections.

Threshold segmentation and segmentation [17] according to the homogeneity criterion based on average brightness (Fig. 4) often do not give the desired results.

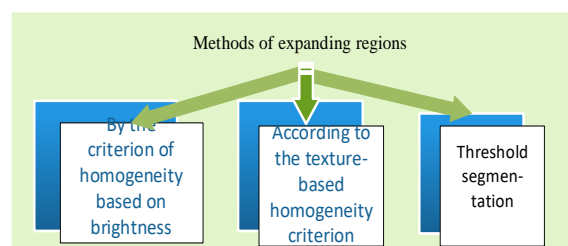


Figure 4: Methods of expanding regions in a video stream

Such segmentation usually results in a large number of small regions. The most effective results are given by the segmentation based on the homogeneity criterion based on the texture [18].

2.3. Selection of Contours

In the video, heterogeneous objects are often observed, so you have to face the task of finding perimeters, curvature, form factors, specific surface area of objects, etc. All these tasks are related to the analysis of contour elements of objects.

Methods for highlighting contours in an image can be divided into three main classes:

1. High-frequency filtering methods [19].
2. Methods of spatial differentiation [20].
3. Methods of functional approximation [21] (Fig. 5).

Common to all these methods is the development of the boundary as a region of a sharp drop in the image brightness function (i), which is distinguished by the introduced mathematical contour model.

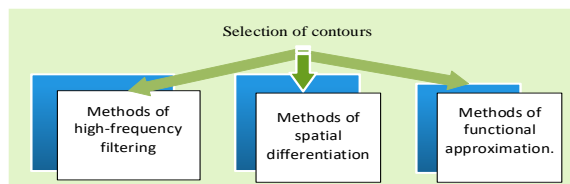


Figure 5: Methods of highlighting contours in a video stream

By the tasks, contour selection algorithms are subject to requirements: the selected contours must be thin, without gaps, and closed. The process of selecting contours is complicated due to the need to apply algorithms for thinning and eliminating gaps. However, the contours are not closed and unsuitable for analysis procedures.

2.4. Methods Based on Clustering

The K -means method is an iterative method used to divide an image into K clusters. The basic algorithm is given below [22]:

Step 1. Choose K cluster centers, randomly or based on some heuristics.

Step 2. Place each image pixel in a cluster whose center is closest to that pixel.

Step 3. Recalculate the cluster centers by averaging all the pixels in the cluster.

Step 4. Repeat steps 2 and 3 until convergence (for example, when the pixels remain in the same cluster).

The distance is usually taken as the sum of squares or absolute values of the differences between the pixel and the center of the cluster.

The difference is most often based on color, brightness, texture, and pixel location, or a balanced combination of these factors.

2.5. Methods Using a Histogram

Histogram-based methods [23] are very efficient when compared to other image segmentation methods because they require only a one-pixel pass.

A histogram is calculated over all pixels in the image and its minima and maxima are used to find clusters in the image. Color or brightness can be used when comparing.

An improvement on this is to recursively apply it to the clusters in the image to divide them into smaller clusters. The process is repeated gradually with smaller and smaller clusters until the moment when new clusters stop appearing altogether.

Approaches based on the use of histograms can also be quickly adapted for multiple frames while retaining their single-pass speed advantage.

2.6. YOLO—Object Detection

You-Only-Look-Once (YOLO) [24] is an independent video object detection system that can operate in real-time at very high frame rates—the common limit is 45 frames per second, with a claimed useful frame rate of up to 155 frames per second. YOLO was originally released in 2015 with Facebook research.

YOLO consists of two main parts: a class detector and a framework detector. The class detector determines which objects are present in the image. The frame detector determines the location of objects in the image. The class detector works by using a regression neural network that learns to predict the value of a variable. It learns to predict the probability that a certain object is present in the image.

The YOLO class detector is a regression neural network with 24 deep layers. The input layer of the network receives a 448×448 pixel image. The output layer of the network contains 84 values. Each value corresponds to the probability that a certain object is present in the image.

2.7. Features of Video Tracking

Digital IP cameras are increasingly used in modern video surveillance systems. Connecting an IP camera to an already existing local network can guarantee minimal installation costs.

Let's consider the characteristics that must be taken into account when choosing computer technologies for a digital video surveillance system.

The first characteristic is the number of physical ports to which other devices can be connected. Will this parameter determine the maximum number of IP cameras that can be connected? For a home video surveillance system, a switch that has 4 ports is often used. Equipment with 8–16–24 ports is used for professional systems [25].

The second characteristic is bandwidth. At the same time, the bandwidth of each port is taken into account. The most common values are 10/100 Mbps and 1 Gbps. It should be taken into account that often the total bandwidth of the switch can be lower than the total value of all ports. When choosing the bandwidth of a cable, you need to determine what data transfer rate your network can handle.

The third characteristic is the speed of data transmission, which will limit the possibility of receiving and transmitting information.

The fourth feature of PoE is the function that allows you to power other devices through the same cable that transmits data. This is very important for the organization of video surveillance, as it allows you to get rid of unnecessary wires, and also simplifies the process of installation and organization of the power supply of connected devices.

The fifth characteristic is management protocols. Yes, PoE switches are divided into managed and unmanaged. Managed switches are devices that support several protocols (functions) of network management and data transmission.

To build simple and small IP surveillance systems, physically isolated from networks in which other critical data is transmitted (telemetry data, banking and financial data, video conferences, etc.), it is possible to dispense with the use of unmanaged PoE switches.

3. Problem Statement

Let's install object tracking in the video stream and examine the speed of object detection.

To do this, we need to write the following command in the terminal:

```
pip install ultralytics  
And then import it into the code:  
from ultralytics import YOLO
```

Now everything is ready to create a neural network model:

```
model = YOLO("yolov8m.pt")
```

As mentioned earlier, YOLOv8 is a group of neural network models. These models were built and trained using PyTorch and exported as .pt files.

The first time you run this code, it will download the yolov8m.pt file from the Ultralytics server to the current folder. It will then construct a model object. You can now train this model, detect objects, and export them for use. There are convenient methods for all these tasks:

train({dataset descriptor file path})—used to train the model on the image dataset.

predict({image})—Used to predict the specified image, for example, to detect the bounding boxes of all objects that the model can find in the image.

export({format})—used to export the model from the default PyTorch format to the specified format.

All YOLOv8 object detection models are already pre-trained on the COCO dataset, which is a huge collection of images of 80 different types.

The prediction method accepts many different types of input data, including a path to a single image, an array of image paths, an Image object from Python's well-known PIL library, and others [26].

After running the input data through the model, it returns an array of results for each input image. Since we only provided one image, it returns an array with one element, which you can extract like this:

```
result = results[0]
```

The result contains detected objects (Fig. 6) and convenient properties for working with them. The most important is the boxes array with information about the detected bounding boxes on the image (Fig. 7).



Figure 6: Video capture of an object and “person” recognition

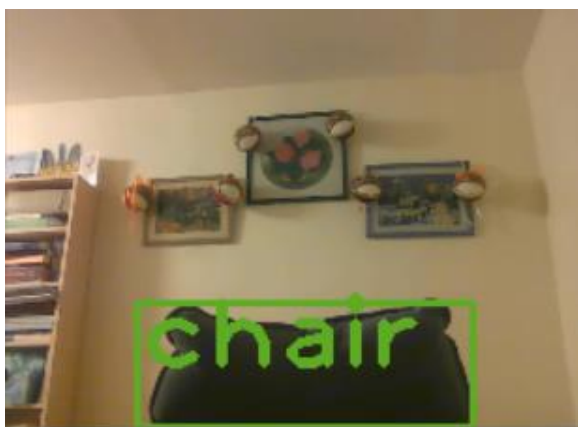


Figure 7: Capturing the object on video and recognizing the “chair”

You can determine how many objects are found by running the len function:

```
len(result.boxes)
```

After launch, “2” was received, which means that two boxes were detected: one for a mobile phone, and the other for a person (Fig. 8).

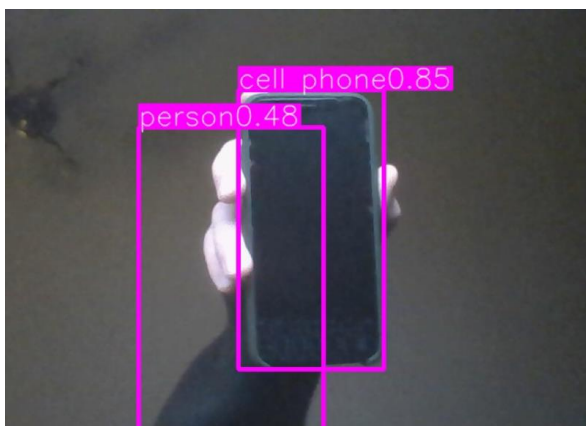


Figure 8: Video capture of two objects and recognition of “person” and “cell phone”

You can analyze each box either in a loop or manually. Let’s take the first object:

```
box = result.boxes[0]
```

The box object contains bounding box properties, including:

xyxy—coordinates of the box in the form of an array [x1,y1,x2,y2]

cls—object type identifier

conf—confidence level of the model regarding this object. If it’s very low, like <0.5, you can just ignore the field.

Let’s display information about the object:

```
print("Object type:", box.cls)
print("Coordinates:", box.xyxy)
print("Probability:", box.conf)
```

For the first object, you will receive information:

```
Object type: tensor([16.])
Coordinates: tensor([[261.1901,
Probability: tensor([0.9528])
```

Since YOLOv8 contains PyTorch models, the outputs from PyTorch models are encoded as an array of PyTorch Tensor objects, so you need to extract the first element from each of these arrays:

```
print("Object type:", box.cls[0])
print("Coordinates:", box.xyxy[0])
print("Probability:", box.conf[0])
```

```
Object type: tensor(16.)
Coordinates: tensor([261.1901,
Probability: tensor(0.9528)
```

Now we see the data as Tensor objects. To extract the actual values from a Tensor, you need to use the .tolist() method for tensors with an array inside, and the .item() method for tensors with scalar values.

Let’s load the data into the corresponding variables:

```
cords = box.xyxy[0].tolist()
class_id = box.cls[0].item()
conf = box.conf[0].item()
print("Object type:", class_id)
print("Coordinates:", cords)
print("Probability:", conf)
```

```
Object type: 16.0
Coordinates: [261.1900634765625,
Probability: 0.9528293609619141
```

Now we see the actual data. Coordinates can be rounded, and probability can also be rounded to two decimal places.

All objects that a neural network can detect have digital identifiers. In the case of the pre-trained YOLOv8 model, there are 80 feature types with IDs from 0 to 79. The COCO feature classes are public. Additionally, the YOLOv8 result object contains a convenient names property to retrieve these classes.

4. Data to Proposed Model

The web application consists of 2 main files:

Flaskapp.py is a file responsible for the project itself, its appearance, and its structure.

YOLO_Video.py is a file that is responsible for the YOLO algorithm, namely for the implementation of object recognition in the video stream.

Implementation of the Flaskapp.py file:

Configuring the Flask application:

A web application is created using the Flask class.

Configuration parameters such as the secret key and the file download folder are set.

Defines UploadFileForm class using Flask-WTF to handle file uploads.

Video processing functions:

Generate_frames and generate_frames_web functions are defined to generate frames based on output from YOLO detection.

These functions use the video_detection function from the YOLO_Video.py file to perform object detection on video frames.

Routes are defined for the home page (/ and /home), the webcam page (/webcam), and the video download page (/FrontPage).

The /video and /webapp routes are responsible for broadcasting video frames with object detection results.

The webcam and front routes render HTML templates for webcam pages and video uploads.

The UploadFileForm class is used to handle uploads of video files.

The application runs on the development server if the script is executed directly.

5. Implementation

The video_detection function takes a video path as input and performs object detection using the YOLO model.

The YOLO model from Ultralytics is loaded from the specified checkpoint file (yolov8n.pt).

The frames of the detected objects on each frame are determined, and the processed frames are returned.

Class names corresponding to detected objects are defined in the classNames list.

Video capture and processing:

OpenCV is used to capture video frames from the specified path.

On each frame, detected objects are drawn along with class labels and confidence levels.

The processed frames are returned for streaming in the user's browser.

The general course of work:

- The user uploads a video file through the interface.
- The file is saved and its path is stored in the Flask session.
- The object detector is called from the received video path.
- The processed video frames are transmitted for real-time viewing through the user's browser.

This project uses Flask for the web application and integrates YOLO for real-time video processing. The YOLO_Video.py file isolates functionality related to YOLO, making it modular and reusable.

When entering the web application, we are greeted by the "title" page:

There are two buttons on this page:

The first Video button sends us to a page where we can upload a video, press the Submit button, and receive the processed video [27].

The second LiveWebcam button sends us to a page where the webcam is connected automatically and displayed on the screen in a processed format.

In the images Figs. 9–16 we can see that the YOLOv8 algorithm is running on the webcam.

Our model is based on pre-trained OSFA and is built on top of PyTorch. The training image size was up to 256×128. A batch size of 64 randomly selected data was then fed to the network. During the test, the test images are also resized to 256×128. Our model is trained on 100 epochs. The values of α_1 , α_2 and the learning rate are the

same as those set by OSFA. α_1 , α_2 and learning rate are set to 1, 0.0007, 3.5×10^{-5} , respectively. In SAM, the number of horizontal parts is 4. All experiments are performed with a hardware environment of 11th Gen Intel(R) Core(TM) i7-11800H at 2.30 GHz and NVIDIA GeForce RTX 3060.

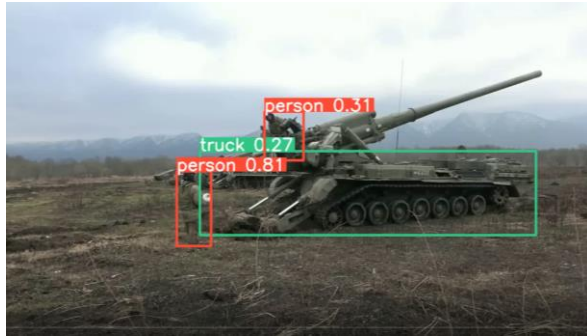


Figure 9: Results of recognition and identification of two people and a truck in the Yolo video sequence

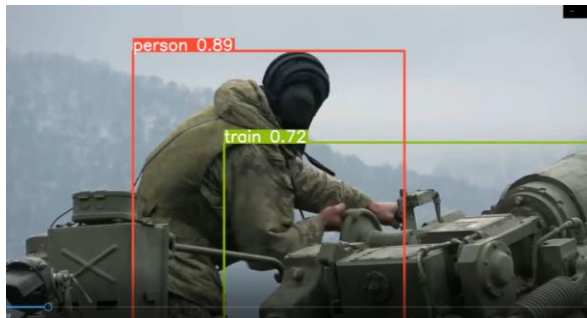


Figure 10: Results of recognition and identification of a person and a Yolo video sequence



Figure 11: Results of object recognition and identification of the Yolo video sequence



Figure 12: Human and tank recognition results in the Yolo video sequence



Figure 13: Recognition results of two people, a tank in the Yolo video sequence



Figure 14: The results of object recognition and identification of the Yolo video sequence



Figure 15: Results of recognition and identification of three objects of the Yolo video sequence



Figure 16: Results of recognition and identification of three objects of the Yolo video sequence

The data were taken from [28] to form the dataset. The protected technology was developed in [29], from where the protected communication channels are taken. The structural diagram of the model was taken in [30], and the use of methods in [31]. Also, the approaches used for image preprocessing were taken in [32] and [33]. Data processing was formed thanks to [34].

6. Conclusions

This study analyzed the YOLOv8 object recognition algorithm and its differences from other machine learning algorithms. A web application for object recognition in a video stream was also created, analyzed, and tested. A clear overview of the development tools for a web application using YOLOv8 has been provided. The PyCharm programming environment, the Flask framework, and its advantages compared to other frameworks were studied. Also explored is Ultralytics, which helps in the development and testing of web applications for video streaming.

As a result of the work performed, a high level of understanding of the YOLOv8 algorithm, its features, and its capabilities in the field of object detection in the video stream was achieved. The developed web application is not only a practical application of thialgorithm but can also serve as a basis for further developments and improvements in this direction.

References

[1] P. Anakhov, et al., Protecting Objects of Critical Information Infrastructure from Wartime Cyber Attacks by Decentralizing the Telecommunications

Network, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3550 (2023) 240–245.

[2] H. Hulak, et al., Dynamic Model of Guarantee Capacity and Cyber Security Management in the Critical Automated System, in: 2nd International Conference on Conflict Management in Global Information Networks, vol. 3530 (2023) 102–111.

[3] V. Grechaninov, et al., Decentralized Access Demarcation System Construction in Situational Center Network, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3188, no. 2 (2022) 197–206.

[4] J. Liu, et al., Deep Industrial Image Anomaly Detection: A Survey, *Mach. Intell. Res.* 21(1) (2024) 104–135. doi: 10.1007/s11633-023-1459-z.

[5] E. Kruger-Marais, Subtitling for Language Acquisition: Eye Tracking as Predictor of Attention Allocation in Education, *Int. J. Lang. Stud.* 18(2) (2024) 129–150. doi: 10.5281/zenodo.10475319.

[6] P. Li, et al., Efficient Long-Short Temporal Attention Network for Unsupervised Video Object Segmentation, *Pattern Recognit.* 146 (2024) 110078. doi: 10.1016/j.patcog.2023.110078.

[7] E. Kawamura, et al., Ground-Based Vision Tracker for Advanced Air Mobility and Urban Air Mobility, *AIAA SciTech 2024 Forum* (2024). doi: 10.2514/6.2024-2010.

[8] Z. Yin, et al., Numerical Modeling and Experimental Investigation of a Two-Phase Sink Vortex and its Fluid-Solid Vibration Characteristics, *J. Zhejiang University-SCIENCE A* 25(1) (2024) 47–62. doi: 10.1631/jzus.a2200014.

[9] T. Wolf, D. Fridovich-Keil, B. Jones, Mutual Information-Based Trajectory Planning for Cislunar Space Object Tracking using Successive Convexification, *AIAA SCITECH 2024 Forum* (2024). doi: 10.2514/6.2024-0626.

[10] H. Zhang, et al., Quality Assessment for DIBR-synthesized Views based on Wavelet Transform and Gradient

- Magnitude Similarity, *IEEE Transactions on Multimedia* (2024) 1–14. doi: 10.1109/tmm.2024.3356029.
- [11] Q. Qin, Y. Chen, A Review of retinal Vessel Segmentation for Fundus Image Analysis, *Eng. Appl. Artif. Intell.* 128 (2024) 107454. doi: 10.1016/j.engappai.2023.107454.
- [12] M. Li, et al., Ftpe-Bc: Fast Thumbnail-Preserving Image Encryption Using Block-Churning, *SSRN* 4698446 (2024).
- [13] M. Vladymyrenko, et al., Analysis of Implementation Results of the Distributed Access Control System. in: *IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology* (2019). doi: 10.1109/picst47496.2019.9061376.
- [14] V. Buriachok, V. Sokolov, P. Skladannyi, Security Rating Metrics for Distributed Wireless Systems, in: *Workshop of the 8th International Conference on “Mathematics. Information Technologies. Education.” Modern Machine Learning Technologies and Data Science*, vol. 2386 (2019) 222–233.
- [15] D. Clayton-Chubb, et al., Metabolic Dysfunction-Associated Steatotic Liver Disease in Older Adults is Associated with Frailty and Social Disadvantage, *Liver Int.* 44(1) (2024) 39–51. doi: 10.1111/liv.15725.
- [16] E. Mira, et al., Early Diagnosis of Oral Cancer using Image Processing and Artificial Intelligence, *Fusion: Practice Appl.* 14(1) (2024) 293–308. doi: 10.54216/fpa.140122.
- [17] L. Qiao, et al., A Multi-Level Thresholding Image Segmentation Method Using Hybrid Arithmetic Optimization and Harris Hawks Optimizer Algorithms, *Expert Syst. Appl.* 241 (2024) 122316.
- [18] Z. Wang, et al., Fibrous Whey Protein Mediated Homogeneous and Soft-Textured Emulsion Gels for Elderly: Enhancement of Bioaccessibility for Curcumin, *Food Chemistry* 437(1) (2024) 137850. doi: 10.1016/j.foodchem.2023.137850.
- [19] J. Gao, Y. Huang, FP-Net: Frequency-Perception Network with Adversarial Training for Image Manipulation Localization, *Multimed. Tools Appl.* (2024) 1–19. doi: 10.1007/s11042-023-17914-1.
- [20] Y. Su, et al., Enhancing Concealed Object Detection in Active Millimeter Wave Images Using Wavelet Transform, *Signal Process.* 216 (2024) 109303. doi: 10.1016/j.sigpro.2023.109303.
- [21] J. Bhandari, D. Russo, Global Optimality Guarantees for Policy Gradient Methods, *Oper. Res.* (2024). doi: 10.1287/opre.2021.0014.
- [22] K. Qian, H. Duan, Optical Counting Platform of Shrimp Larvae Using Masked K-Means and a Side Window Filter, *Appl. Optics* 63(6) (2024) A7–A15. doi: 10.1364/ao.502868.
- [23] S. Lee, et al., Intensity Histogram-Based Reliable Image Analysis Method for Bead-Based Fluorescence Immunoassay, *BioChip J.* (2024) 1–9. doi: 10.1007/s13206-023-00137-9.
- [24] J. Redmon, et al., You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition* (2016) 779–788. doi: 10.1109/CVPR.2016.91.
- [25] T. Zhang, et al., The Design and Implementation of a Wireless Video Surveillance System, *21st Annual International Conference on Mobile Computing and Networking* (2015) 426–438. doi: 10.1145/2789168.2790123.
- [26] M. Nazarkevych, et al., The Ateb-Gabor Filter for Fingerprinting, *Conference on Computer Science and Information Technologies, AISC 1080* (2019) 247–255. doi: 10.1007/978-3-030-33695-0_18.
- [27] V. Sokolov, P. Skladannyi, A. Platonenko, Video Channel Suppression Method of Unmanned Aerial Vehicles, in: *IEEE 41st International Conf. on Electronics and Nanotechnology* (2022) 473–477. doi: 10.1109/ELNANO54667.2022.9927105.
- [28] M. Nazarkevych, Data Protection Based on Encryption Using Ateb-Functions, *XIth International Scientific and Technical Conference Computer Sciences and Information Technologies* (2016) 30–32. doi: 10.1109/STC-CSIT.2016.7589861.
- [29] M. Medykovskyy, et al., Methods of Protection Document Formed from Latent Element Located by Fractals, *Xth*

- International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT) (2015) 70–72. doi: 10.1109/STC-CSIT.2015.7325434.
- [30] V. Sheketa, et al., Formal Methods for Solving Technological Problems in the Infocommunications Routines of Intelligent Decisions Making for Drilling Control, IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (2019) 29–34. doi: 10.1109/PICST47496.2019.9061299.
- [31] V. Sheketa, et al., Empirical Method of Evaluating the Numerical Values of Metrics in the Process of Medical Software Quality Determination, International Conference on Decision Aid Sciences and Application (2020) 22–26. doi: 10.1109/DASA51403.2020.9317218.
- [32] N. Boyko, N. Tkachuk, Processing of Medical Different Types of Data Using Hadoop and Java MapReduce, in: 3rd International Conference on Informatics & Data-Driven Medicine Vol. 2753 (2020) 405–414.
- [33] N. Boyko, et al., Fractal Distribution of Medical Data in Neural Network, in: IDDM Vol. 2448 (2019) 307–318.
- [34] I. Tsmots, et al., The Method and Simulation Model of Element Base Selection for Protection System Synthesis and Data Transmission, Int. J. Sensors Wireless Commun. Control 11(5) (2021) 518–530. doi: 10.2174/2210327910999201022194630.