UDC 004.932:004.42]:621.397
DOI: 10.15587/2706-5448.2024.319799

**Oleksandr Laktionov,**
**Alina Yanko,**
**Alina Hlushko**

# DEVELOPMENT OF A HARDWARE-SOFTWARE SOLUTION FOR DETECTION OF COMPLEX-SHAPED OBJECTS IN VIDEO STREAM

*The object of the study is the process of diagnosing complex-shaped objects in a video stream. The paper investigates the applied problem of creating a hardware-software solution for detecting complex-shaped objects in a video stream. Single-board computers Raspberry Pi models 4 and 5 with additional UPS HAT (D) modules and 21700 batteries were used as hardware, ensuring operation in the absence of power supply. Serial Camera Interface cameras and Full HD 1080p webcams were connected to the single-board computers to study effective methods of video processing using several studied video processing architectures. Eight video processing architectures based on the Oriented Features from Accelerated Segment Test and Rotated Binary Robust Independent Elementary Features and Scale-Invariant Feature Transform methods were considered. Each video processing architecture was tested using a one-minute video, where its average performance was determined. The limitations of video processing were a region of interest of 400×300 pixels and the presence of a limited number of reference images. To automate the launch of programs on single-board computers, the systemd initialization system was used.*

*Known video processing algorithms were considered and a modification of the algorithm was proposed by using a double check for the presence of an object in the video stream. A hardware-software solution was implemented, consisting of a single-board computer with external cameras connected to it, and software for detecting complex-shaped objects in the video stream was created. The solution is useful as an auxiliary tool for detecting complex-shaped objects in the video stream on robotic platforms, in industry, everyday life, the educational process, and when repairing electronic modules. The practical significance of the study lies in the fact that the architecture for processing complex-shaped objects has been further developed. They provide for a double check for the presence of an object in the video stream, which increases the processing time of one frame, and on the other hand, increases the efficiency of object detection based on only one reference photo.*

**Keywords:** *computer vision, single-board computer, initialization system, double-check, video processing algorithms.*

*Received: 13.10.2024*
*Received in revised form: 30.11.2024*
*Accepted: 20.12.2024*
*Published: 31.12.2024*



***How to cite***
*Laktionov, O., Yanko, A., Hlushko, A. (2024). Development of a hardware-software solution for detection of complex-shaped objects in video stream. Technology Audit and Production Reserves, 6 (2 (80)), 35–40. https://doi.org/10.15587/2706-5448.2024.319799*

## 1. Introduction

Real-time video processing methods are increasingly in demand in the market. Object recognition in video is carried out at any time of the day and in different weather conditions [1]. In this regard, a number of computing systems are being developed, in particular, based on the principle of fault tolerance [2, 3] and data processing reliability [4].

In practice, computer vision models are used to solve various problems, for example, to diagnose the state of regions of the country [5]. Existing models You Only Look Once [6], Single Shot Multibox Detector [7] involve the use of a large number of images for training, at least 500. Such a number is difficult to prepare very quickly, especially in conditions of uncertainty.

In conditions of rapid technological development, the search for effective solutions that would combine innovation, accessibility and ease of use is becoming increasingly relevant. This is especially important in areas where the speed of implementation and adaptation of technologies depends on their commercial effectiveness. A key factor in this context is the implementation of flexible approaches to technological solutions that allow minimizing costs and increasing efficiency [8]. The relevance of the study is explained by the need for quick setup of models and their implementation on single-board computers for object detection given the existence of a small number of images, 30 or less. Therefore, it is necessary to consider simple solutions for detecting complex objects in a video stream based on a minimum number of input images. This will allow creating easy-to-use, cheap and reliable solutions for object detection that can be maintained by a person with basic computer knowledge.

As noted in [6], the You Only Look Once model assumes the existence of large volumes of input images that need to be prepared in a certain way to build the model. After that, using equipment with limited technical characteristics,

 ISSN-L 2664-9969; E-ISSN 2706-5448

it is difficult to optimize the model for high performance. Therefore, let's consider computer vision algorithms that can detect objects based on a minimum number of input images of complex-shaped objects.

In [9], a method for registering a point cloud based on Scale-Invariant Feature Transform, SIFT, and geometry functions is proposed. The uniqueness of the method lies in the selection of an object using threshold segmentation, where the initial points are obtained by SIFT. The accuracy of the solution proposed in the work was compared by comparative analysis with the existing iterative closest point method, which allowed to prove its superiority.

To increase the performance of the SIFT algorithm in [10], the traditional Scale-Invariant Feature Transform was modified by introducing a two-way Laplace filter for finding extrema.

Data acquisition and analysis systems based on the integrated Scale-Invariant Feature Transform algorithm were proposed in [11]. The uniqueness of the proposed algorithm is the improvement, which consists in reducing the dimensionality of feature points. Testing on five sets of images showed an advantage of point matching by 7.6 % and feature point extraction by 8 %.

In [12], optical image registration is investigated through a new Scale-Invariant Feature Transform structure based on nonlinear diffusion and polar spatial-frequency descriptor. In contrast to [11], a nonlinear scale-invariant feature transformation is proposed. The experimental results show that the proposed method is superior to existing ones.

In contrast to [11], SIFT is considered in [13] for comparing and classifying hyperspectral images. The uniqueness of the study is the feature of the image processing process, where instead of grayscale transformation, spectral transformation is used. This allows to expand the informativeness of the image. The input images were obtained by a hyperspectral camera with wavelengths of 362.05–1002.47 nm. The proposed model is universal and allows to analyze various objects in the images, in particular cars, building materials, etc.

Recognition of drawn electronic components using Oriented Features from Accelerated Segment Test and Rotated Binary Robust Independent Elementary Features, ORB was studied in [14]. The study was conducted on the basis of fifteen electronic components with three different orientations. The user drew an image of the electronic component on a graphical interface, which was fed to the input of the ORB model, which compared it with the FAST method.

An FPGA-based ORB descriptor accelerator is proposed in [15]. The FPGA architecture extracts ORB descriptors with high throughput, which is the uniqueness of the proposed method. According to the experimental results, the architecture achieved a throughput of 94 Mpix/s at a clock frequency of 100 MHz.

The effect of pre-filtering on object detection based on Oriented Fast and Rotating BRIEF (ORB) is proposed in [16]. The input data were 150 images on an Android device and 19 filter configurations. The uniqueness of the study is the approach based on the covariance of the method for selecting Hamming distance thresholds for each of the pre-filtered ORB detectors.

An ORB-based face recognition system for human-robot interaction is proposed in [17]. The scientific novelty of the study is the new ORB-KPCA technique for FR together with threshold-based filtering (TBF). The experiment was conducted on three control datasets (ORL, Faces96 and Grimace), where the effectiveness of the proposed method is proven.

In [18], a high-performance, energy-efficient and functionally accurate hardware accelerator for ORB-SLAM is proposed. It works on the basis of the technique of finding the optimal static template for performing parallel access to banks based on the genetic algorithm. The proposed solution achieves an 8-fold speedup and reduced energy consumption, which is experimentally proven.

An optimized SIFT-OCT algorithm for stitching aerial images of a pine plantation is proposed in [19]. The experiment was conducted on three sets of forest images using existing methods and the proposed method, which demonstrated an advantage over known methods in terms of reducing the time for stitching photographs.

A simple solution for image segmentation is proposed in [20], which is based on the ant algorithm. The scientific novelty of the proposed method is that it takes into account the number of ants in the image, weight, initial amount and rate of pheromone evaporation. This allows increasing the contrast of the image, as shown by practical tests. Similar studies were also conducted by the authors of [21], who investigated the post-processing of compressed noisy images with the BM3D filter.

The method for stabilizing the brightness of the video stream was studied in [22]. According to the results of the studies, it was found that the optimal indicator of the quality of the video stream is the average frame brightness.

In conditions of different lighting, variable video stream quality and other variable factors, it is important to solve the problem of the accuracy of finding (highlighting) the boundaries of complex objects for further recognition [23]. The above factors can lead to a blurring of clear boundaries between objects, which leads to the inefficiency of existing detection methods. An effective direction for solving this problem is the use of fuzzy inference systems. Classical methods of object boundary selection based, for example, on the Canny edge detector, are not always effective under conditions of variable illumination. It has been proven that the use of fuzzy systems allows for the approximation of any mathematical model. And under conditions of uncertainty, incompleteness and variability of operating conditions, such systems are an indispensable auxiliary tool. Thus, the use of fuzzy inference systems with the possibility of retraining has been considered for solving a set of problems under conditions of uncertainty or incompleteness of input data about the state of the system [24, 25]. Therefore, under conditions of possible changes in illumination, fuzzy inference systems can be indispensable for finding the boundaries of complex objects with low efficiency of classical methods. The solution of the problem in this case is reduced to determining the input parameters, compiling the rule base and determining the output variable. For Canny edge detector approximation, for example, gradient values can be used as input variables, and the output variable can be the binary value of the central pixel, the rule base when using the Anfis model will be generated in an automated mode. The direction of further research of the specified approach is to determine the performance for images of different sizes.

Thus, the current direction of improving object detection methods, to which the conducted research is devoted, is to reduce the dimensionality of feature points, and use spectral transformation instead of grayscale transformation.

Thus, *the aim of research* is to implement a hardware-software solution with double checking for the presence of an object in the video stream. This will make it possible to increase the efficiency of detecting complex-shaped objects.

## 2. Materials and Methods

*The object of research* is the process of diagnosing complex-shaped objects in a video stream. The study is based on a given set of reference images with complex-shaped objects. The task is to study algorithms and methods for real-time video processing to select optimal settings configurations. The metrics of the study are the performance and accuracy of detecting complex-shaped objects.

The performance of real-time video processing was measured on one-minute videos. The hardware will be single-board computers Raspberry Pi 4 model B 8 Gb and Raspberry Pi 5 8 Gb with Serial Camera Interface cameras and Full HD 1080p webcams. To build image processing architectures, the Oriented Features from Accelerated Segment Test and Rotated Binary Robust Independent Elementary Features methods from [26] and Scale-Invariant Feature Transform [27] were used. The image processing architectures under study (first-third, fifth-eighth) were built in a limited frame (Region of Interest) with dimensions of 400×300 pixels and with RANdom SAmple Consensus m.distance<0.75x n.distance. For the convenience of visual perception of the interface, the limited Region of Interest frame had a frame with a thickness of 1 pixel. During the testing process, the frame processing performance was recorded to a file, where the average performance value was calculated at the end of video processing.

*Image processing architecture of the first algorithm.* Image preprocessing using several filters, in particular GaussianBlur, Sobel, cv2.magnitude, cv2.normalize. The GaussianBlur filter is used for smoothing with a kernel size of 5×5. Smoothed objects are fed to the input to determine the gradient by the Sobel operator along the $x$, $y$ axes. Next, the gradient is determined by the magnitude function and pixels are normalized in the range [0, 255] using NORM_MINMAX.

The pre-processed images are fed to the input of the ORB detector for point detection. The matching of points with the reference is implemented by Fast Library for Approximate Nearest Neighbors [28] with the parameters algorithm=1, trees=10, checks=50.

*The image processing architecture of the second algorithm* did not involve the use of filters. ORB was used immediately and the matching of points with the reference is implemented by Fast Library for Approximate Nearest Neighbors with the parameters algorithm=1, trees=40, checks=10.

*The image processing architecture of the third algorithm* was an improved version of the second algorithm, where the search for optimal parameter values for Fast Library for Approximate Nearest Neighbors was used, which are described by the expressions:

trees(k)=5 if k<100 else 6 if k<200 else 7 if k<300 else 8 if k<400 else 9 if k<500 else 10,

cheks(k)=20 if k<100 else 30 if k<200 else 40 if k<300 else 50 if k<400 else 60 if k<500 else 70.

*Image processing architecture of the fourth algorithm.* Creation of the Scale-Invariant Feature Transform detector object and the BFMatcher point matching method. Loading the reference image and creating the THRESH_BINARY_INV mask. Finding key points using the detectAndCompute method. Processing frames by converting to grayscale and searching for points and descriptors using detectAndCompute. Determining correspondences between points using BFMatcher.knnMatch and filtering them using the Lowe test [29]. Determining homography using cv2.findHomography() and transforming the coordinates of the corners of the template image using cv2.perspectiveTransform(). Visualizing all points in the frame using drawKeypoints.

The next research tool was the calibration of parameters for the Scale-Invariant Feature Transform. The following parameters were studied here:

nfeatures_values={50, 100, 200, 300, 400, 500, 600},
contrastThreshold_values={0.1, 0.2, 0.3, ..., 1.0},
sigma_values={1.0, 1.1, ..., 1.8},
edgeThreshold_values={5, 10, 15, 20, 25, 30, 40, 50}.

The determined optimal parameter values were used to build SIFT, this is *the fifth image processing architecture.*

*Similar to the fifth image processing architecture were the sixth and seventh*, where attention was focused on setting the parameters SIFT, BFMatcher, FlannBasedMatcher and the main image processing cycle process_video.

*The image processing architecture of the eighth algorithm* assumed the existence of an ORB detector with the parameters nfeatures, scaleFactor, nlevels, edgeThreshold, firstLevel, WTA_K, scoreType=cv2.ORB_HARRIS_SCORE, patchSize. Key points were mapped BFMatcher with cv2.NORM_HAMMING, crossCheck=True.

As indicated in the formal statement of the task, the implementation of the studied image processing algorithms was carried out on single-board computers Raspberry Pi 4 model B and Raspberry Pi 5. The creation of algorithms for processing complex-shaped objects and experimental verification were also carried out using Visual Studio Code [30]. Thus, the preliminary creation and testing of image processing algorithms was carried out in the Visual Studio Code environment, which was the first step of the research.

The second step of the research involved the installation of an electrical circuit of single-board computers, where webcams were connected. Single-board computers were connected to external monitors and used without monitors using the Secure Shell and RealVNC protocols. To conduct the research, a virtual environment myenv was created with the necessary package of installed libraries.

The file with the program text was sent to the single-board computer, where testing was carried out with manual start-up and automatic [31], using an autorun service file with a graphical interface.

## 3. Results and Discussions

The results of previous work [32] allowed to create a theoretical basis for the study of the complex-shaped object recognition scheme in the video stream. The study of image processing was carried out according to the general scheme:

*Stage 1.* Loading reference images onto a single-board computer.

*Stage 2.* Automatic launch of the selected image processing model after turning on the single-board computer.

*Stage 3.* Study of complex-shaped objects.

First of all, a study was conducted on the use of other distance metrics of the Scale-Invariant Feature Transform algorithm, in particular, chi-square, where the desired result was not obtained. In this regard, in accordance with the research methodology, an analysis of different methods of processing complex-shaped objects was carried out.

The first, second and third architectures implemented in the programs 1.ipynb, 2.ipynb, 3.ipynb, respectively. The program launches demonstrated a high performance value fps1=19.07 ms; fps2=19.44 ms; fps3=24.48 ms, respectively, but low image processing accuracy. The third program autonomously changed the optimal parameter values, where trees=7, checks=40 and trees=9, checks=60 were used. The control of the studied points was carried out in the window, Fig. 1.

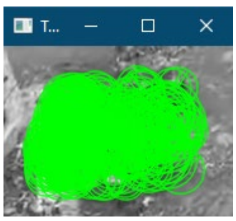

**Fig. 1.** Reference image output window and selection of studied points using the example of the 3.ipynb program

Using the reference image output window allowed to analyze object detection, although it had a slight impact on frame processing performance.

The fourth architecture is implemented in the 4.ipynb file, where the experimental launch of the program demonstrated performance fps4=7.45 ms. But it is worth noting that the studied object was immediately detected and surrounded by a corresponding frame.

The next attempt was to first determine the optimal values of the nfeatures_values, contrastThreshold_values, sigma_values and edgeThreshold_values parameters, the values of which were 200; 0.1; 30; 1, respectively. The determined parameters were used for SIFT, which is implemented in the 5.ipynb file, which demonstrated fps5=8.42 ms.

The sixth and seventh image processing architectures are implemented in files 6.ipynb, 7.ipynb, respectively. Here, attention was focused on the software implementation of the main processing cycle. When increasing the key points from 500 to 1000, there were problems with false positives and low performance values – fps6=7.98 ms; fps7=7.01 ms, respectively.

The eighth architecture is implemented in file 8.ipynb. Based on the ideas of [33, 34], the eighth image processing method had a double check. The first check was the correspondence of the key points being compared, BFMatcher. The second check involved using only points that met the condition min_match_count=7 or more.

The first attempt to launch the eighth image processing method involved the use of the following parameters: nfeatures=500, scaleFactor=1.2, nlevels=8, edgeThreshold=31, firstLevel=0, WTA_K=2, scoreType=cv2.ORB_HARRIS_SCORE, patchSize=31. According to the research results, let's obtain fps8.1=8.5 ms.

The second attempt to launch involved the use of nfeatures=500, scaleFactor=1.4, nlevels=8, edgeThreshold=31, firstLevel=0, WTA_K=2, scoreType=cv2.ORB_FAST_SCORE, patchSize=25. This allowed to increase the performance from 8.5 ms to 15.81 ms, since the optimal values of the ORB parameters were used. Thus, double checking ensures the adequacy of the obtained result.

As can be seen from the research results, increasing the accuracy of object detection in the video stream reduces the performance of video processing. Processing performance also depends on the type of connected video camera and the type of single-board computer. Maximum performance is obtained on a Raspberry Pi 5 single-board computer with a Full HD 1080p webcam.

Basically, effective search for an object in a photo is demonstrated with a performance of up to 8–9 ms. At a performance of 10–25 ms, the detected object is lost due to the presence of noise. Therefore, various usage scenarios are possible. If it is necessary to accurately find an object, image processing algorithms with higher accuracy and lower performance are used. Or vice versa, when it is necessary to find it quickly, and accuracy is not important, then architectures that demonstrate a performance of 10–25 ms and more are used.

The created program texts were launched using Raspberry Pi models 4 and 5. When using Raspberry Pi 4 and 5, with a power supply of 220 volts, there were no problems with the stability of the single-board computers. To ensure energy independence and stable operation during power outages, an additional UPS HAT (D) module with 21700 batteries was used.

In order to facilitate the operation of the created solution, in particular, to quickly enable the software in laboratory and field conditions, automatic launch of the graphical interface using the systemd initialization system was implemented, Fig. 2.

Fig. 2 shows the general principle of creating a systemd initialization system. Thus, when a single-board computer is turned on, the computer vision tool is immediately turned on, which does not require entering additional commands, in particular, activating the myenv environment, etc. From a practical point of view, this reduces the time for preparing the equipment for work and does not require highly qualified specialists.

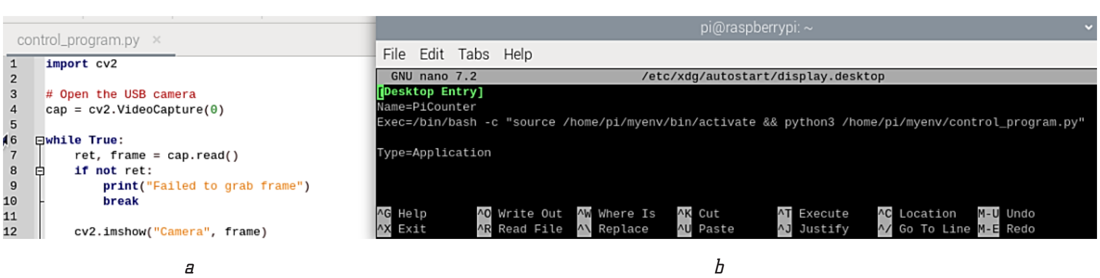


**Fig. 2.** File: *a* – the main, which is enabled automatically; *b* – the window for creating a service file by the systemd initialization system

The developed hardware-software solution was used in laboratory conditions to quickly detect defects in welding metal parts or the quality of soldering electronic modules. This allows to effectively detect poor-quality seams during welding and poorly soldered contacts to electronic modules. The limitations of the study are the use of only cameras without night vision functions.

The proposed hardware-software solution can be improved by using remote methods of controlling the software of a single-board computer. For example, it is considered a method of controlling a single-board computer with a Speedybee F405v3 flight controller connected to it. Control was organized using Radiomaster TX16S equipment. Fig. 3 shows a general diagram of connecting a single-board computer to a flight controller.

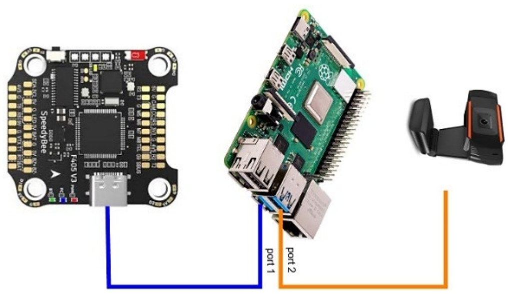


**Fig. 3.** General diagram of connecting a Raspberry Pi single-board computer to the Speedybee F405V3 flight controller

As can be seen from Fig. 3, a webcam is also connected to the single-board computer, which detects complex-shaped objects. The prototype of the circuit was used in combination with a RushFPV video signal transmission device and a monitor. Remote control of the software was implemented using the MultiWii Serial Protocol [35]. The use of remote control expanded the possibilities of inspecting electronic modules for the quality of soldered conductors. This became possible by moving the device to any location for making.

Several aspects are limitations of the research. The behavior of the night vision camera with the proposed software has not been fully studied. The use of other hardware, in particular NVIDIA Jetson and programming languages, has not been sufficiently studied. The performance of the program when using 11 or more template images has not been studied.

The prospect of future research is to find solutions to improve existing algorithms or develop a new algorithm for detecting complex-shaped objects. The algorithm should provide productive operation on single-board computers and minimize noise in input frames.

## 4. Conclusions

During the work, the task of proposing an effective architecture for processing complex-shaped objects based on double checking the presence of an object in the video stream under the condition of a small volume of input images was solved using ORB. In the first iteration of the verification, points were compared with BFMatcher, and in the second, only points that meet the condition min_match_count=7 or more were checked.

The task of proposing hardware and software for processing complex-shaped objects was solved using Raspberry Pi 8 Gb single-board computers with a Serial Camera Interface camera and a Full HD 1080p webcam. Raspberry Pi was equipped with an autonomous power supply system. The systemd initialization system was used to automate the software launch.

Effective object search in the video stream was demonstrated at a performance of up to 8–9 ms. At a performance of 10–25 ms, the detected object was lost due to the existence of noise.

## Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, including financial, personal, authorship, or any other, that could affect the study and its results presented in this article.


## Financing

The research was performed without financial support.


## Data availability

The manuscript has no associated data.

## Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating this work.

### References


1. Guo, J., Shao, M., Chen, X., Yang, Y., Sun, E. (2024). Research on night-time vehicle target detection based on improved KSC-YOLO V5. *Signal, Image and Video Processing, 19 (1).* https://doi.org/10.1007/s11760-024-03576-5
2. Krasnobayev, V., Yanko, A., Fil, I. (2024). Fault-Tolerant Operation of an Integer Data Processing System. *Mathematical Modeling and Simulation of Systems.* Springer, 172–185. https://doi.org/10.1007/978-3-031-67348-1_13
3. Krasnobayev, V., Yanko, A., Kovalchuk, D., Fil, I. (2024). Synthesis of a Mathematical Model of a Fault-Tolerant Real-Time Computer System Operating in Non-positional Arithmetic in Residual Classes. *Mathematical Modeling and Simulation of Systems.* Springer, 186–199. https://doi.org/10.1007/978-3-031-67348-1_14
4. Krasnobayev, V., Yanko, A., Hlushko, A., Kruk, O., Kruk, O., Gakh, V. (2023). Cyberspace protection system based on the data comparison method. *Economic and cyber security.* Kharkiv: PC TECHNOLOGY CENTER, 3–29. https://doi.org/10.15587/978-617-7319-98-5.ch1
5. Pushkarenko, Y., Zaslavskyi, V. (2024). Research on the state of areas in Ukraine affected by military actions based on remote sensing data and deep learning architectures. *Radioelectronic and Computer Systems, 2024 (2),* 5–18. https://doi.org/10.32620/reks.2024.2.01
6. *Ultralytics YOLO Docs.* Available at: https://docs.ultralytics.com
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.; C. Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.) (2016). SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016.* Cham: Springer, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
8. Svistun, L., Glushko, A., Shtepenko, K. (2018). Organizational Aspects of Development Projects Implementation at the Real Estate Market in Ukraine. *International Journal of Engineering & Technology, 7 (3.2),* 447. https://doi.org/10.14419/ijet.v7i3.2.14569
9. Ran, Y., Xu, X. (2020). Point cloud registration method based on SIFT and geometry feature. *Optik, 203,* 163902. https://doi.org/10.1016/j.ijleo.2019.163902

ISSN-L 2664-9969; E-ISSN 2706-5448


10. Singh, P. K., Jana, B., Datta, K. (2022). Robust data hiding scheme through distinct keypoint selection exploiting modified Bilateral-Laplacian SIFT with encoding pipeline. *Displays, 74,* 102268. https://doi.org/10.1016/j.displa.2022.102268
11. Sun, Y. (2024). Exploration on Data Collection and Analysis System Based on Integrated SIFT Algorithm. *Procedia Computer Science, 243,* 388–395. https://doi.org/10.1016/j.procs.2024.09.048
12. Yu, Q., Ni, D., Jiang, Y., Yan, Y., An, J., Sun, T. (2021). Universal SAR and optical image registration via a novel SIFT framework based on nonlinear diffusion and a polar spatial-frequency descriptor. *ISPRS Journal of Photogrammetry and Remote Sensing, 171,* 1–17. https://doi.org/10.1016/j.isprsjprs.2020.10.019
13. Li, Y., Li, Q., Liu, Y., Xie, W. (2019). A spatial-spectral SIFT for hyperspectral image matching and classification. *Pattern Recognition Letters, 127,* 18–26. https://doi.org/10.1016/j.patrec.2018.08.032
14. Pavithra, S., Shreyashwini, N. K., Bhavana, H. S., Nikhitha, G., Kavitha, T. (2023). Hand-Drawn Electronic Component Recognition Using ORB. *Procedia Computer Science, 218,* 504–513. https://doi.org/10.1016/j.procs.2023.01.032
15. Belmessaoud, N. M., Bentoutou, Y., Chikr El-Mezouar, M. (2022). FPGA implementation of feature detection and matching using ORB. *Microprocessors and Microsystems, 94,* 104666. https://doi.org/10.1016/j.micpro.2022.104666
16. Sharif, H., Hölzel, M. (2017). A comparison of prefilters in ORB-based object detection. *Pattern Recognition Letters, 93,* 154–161. https://doi.org/10.1016/j.patrec.2016.11.007
17. A, V., Cholin, A. S., Bhat, A. D., Murthy, K. N. B., Natarajan, S. (2018). An Efficient ORB based Face Recognition framework for Human-Robot Interaction. *Procedia Computer Science, 133,* 913–923. https://doi.org/10.1016/j.procs.2018.07.095
18. Taranco, R., Arnau, J.-M., González, A. (2023). LOCATOR: Low-power ORB accelerator for autonomous cars. *Journal of Parallel and Distributed Computing, 174,* 32–45. https://doi.org/10.1016/j.jpdc.2022.12.005
19. Wu, T., Hung, I.-K., Xu, H., Yang, L., Wang, Y., Fang, L., Lou, X. (2022). An Optimized SIFT-OCT Algorithm for Stitching Aerial Images of a Loblolly Pine Plantation. *Forests, 13 (9),* 1475. https://doi.org/10.3390/f13091475
20. Khudov, H., Hridasov, I., Khizhnyak, I., Yuzova, I., Solomonenko, Y. (2024). Segmentation of image from a first-person-view unmanned aerial vehicle based on a simple ant algorithm. *Information and Controlling System, 4 (9 (130)),* 44–55. https://doi.org/10.15587/1729-4061.2024.310372
21. Rebrov, V., Lukin, V. (2023). Post-processing of compressed noisy images by BM3D filter. *Radioelectronic and Computer Systems, 4,* 100–111. https://doi.org/10.32620/reks.2023.4.09
22. Bilozerskyi, V., Dergachov, K., Krasnov, L., Zymovin, A., Popov, A. (2023). New method for video stream brightness stabilization: algorithms and performance evaluation. *Radioelectronic and Computer Systems, 3,* 125–135. https://doi.org/10.32620/reks.2023.3.10
23. Shefer, O., Laktionov, O., Pents, V., Hlushko, A., Kuchuk, N. (2024). Practical principles of integrating artificial intelligence into the technology of regional security predicting. *Advanced Information Systems, 8 (1),* 86–93. https://doi.org/10.20998/2522-9052.2024.1.11
24. Polshchykov, K., Zdorenko, Y., Masesov, M. (2015). Neuro-fuzzy system for prediction of telecommunication channel load. *2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T).* Kharkiv, 33–34. https://doi.org/10.1109/infocommst.2015.7357261
25. Onyshchenko, S., Haitan, O., Yanko, A., Zdorenko, Yu., Rudenko, O. (2024). Method for detection of the modified DDoS cyber attacks on a web resource of an Information and Telecommunication Network based on the use of intelligent systems. *Proceedings of the Modern Data Science Technologies Workshop (MoDaST 2024).* Lviv, 219–235. Available at: https://ceur-ws.org/Vol-3723/paper12.pdf
26. Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision,* 2564–2571. https://doi.org/10.1109/iccv.2011.6126544
27. Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision,* 1150–1157. https://doi.org/10.1109/iccv.1999.790410
28. Muja, M., Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Application (VISSAPP),* 331–340. https://doi.org/10.5220/0001787803310340
29. Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision, 60 (2),* 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94
30. *Visual Studio Code.* Available at: https://code.visualstudio.com/
31. Männel, W. (2023). *The ultimate guide on using systemd to autostart scripts on the Raspberry Pi.* TheDigitalPictureFrame.com. Available at: https://www.thedigitalpictureframe.com/ultimate-guide-systemd-autostart-scripts-raspberry-pi/
32. Laktionov, O., Yanko, A., Pedchenko, N. (2024). Identification of air targets using a hybrid clustering algorithm. *Eastern-European Journal of Enterprise Technologies, 5 (4 (131)),* 89–95. https://doi.org/10.15587/1729-4061.2024.314289
33. Fedorenko, G., Fesenko, H., Kharchenko, V. (2022). Analysis of methods and development of the concept of guaranteed detection and recognition of explosive objects. *Innovative Technologies and Scientific Solutions for Industries, 4 (22),* 20–31. https://doi.org/10.30837/itssi.2022.22.020
34. Sun, Y., Fesenko, H., Kharchenko, V., Zhong, L., Kliushnikov, I., Illiashenko, O., Morozova, O., Sachenko, A. (2022). UAV and IoT-Based Systems for the Monitoring of Industrial Facilities Using Digital Twins: Methodology, Reliability Models, and Application. *Sensors, 22 (17),* 6444. https://doi.org/10.3390/s22176444
35. *MultiWii. MultiWii flight controller source code and documentation.* Available at: https://github.com/multiwii



***Oleksandr Laktionov,*** *PhD, Associate Professor, Department of Automation, Electronics and Telecommunications, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine, ORCID: https://orcid.org/0000-0002-5230-524X*

***Alina Yanko,*** *PhD, Associate Professor, Department of Computer and Information Technologies and Systems, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine, ORCID: https://orcid.org/0000-0003-2876-9316*

✉***Alina Hlushko,*** *PhD, Associate Professor, Department of Finance, Banking and Taxation, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine, e-mail: glushk.alina@gmail.com, ORCID: https://orcid.org/0000-0002-4086-1513*

✉*Corresponding author*