UDC 004.8

N. Bulbotka, O. Polshakova

USING COMPUTER VISION FOR AUTOMATED OBJECT TRACKING SYSTEM

*Abstract:*The article examines the use of computer vision technologies to automate the process of tracking objects in a video stream. The developed system is described, which implements the recognition, tracking and determination of the characteristics of moving objects using the YOLOv8 model. The system includes modules for video display, recognition, tracking, and determination of object characteristics. The process of retraining the YOLOv8 model on specific data sets is described, as well as the application of algorithms for determining the speed of moving objects.

The proposed solution allows analysis of the video stream in real time. The results confirm the compliance of the developed system with the set requirements and its practical suitability in the areas of video surveillance, analysis of the behavior of objects, unmanned aerial vehicles and transport systems.

*Keywords:*automated system, recognition algorithm, object tracking, computer vision, Python, YOLOv8.

Introduction

At this stage of world development, the industrial landscape is constantly changing under the influence of technological innovations. One of such innovations is computer vision systems, which today are widely used in many industries, including security monitoring, human or animal behavior analysis, traffic flow management, etc.

Traditional tracking methods, which are based on manual analysis of a large number of frames, require significant human resources and time to process the video data, which makes it difficult to respond to events in time or to track specific classes of objects. Moreover, the human factor can lead to errors or loss of some data when analyzing videos from different sources. In addition, in the conditions of growing volumes of video data and the need for their effective analysis, there is a demand for systems capable of automating the processes of object recognition and tracking in real time. In turn, this is especially relevant for unmanned vehicles, transport and video surveillance systems, where accuracy and speed of analysis are of great importance.

Object tracking is one of the most important tasks of computer vision, which requires the integration of modern machine learning algorithms and, accordingly, neural networks. In this context, the models of the YOLO series have proven themselves to be among the most effective in image processing tasks due to their ability to work in real time.

 ISSN 1560-8956

The goal of the work is to create an effective system that provides accurate recognition, tracking and analysis of objects, simplifying user interaction with large streams of video data. Implementation of this approach makes it possible to optimize observation processes, increase their reliability and accuracy, as well as demonstrate the prospects of using computer vision technologies to solve current problems.

Problem Statement

As part of the researched problem of automating the process of tracking specific objects, specific tasks have been defined that must be performed to achieve the set goal. This includes categorizing objects in the video, tracking selected objects that may go out of the frame or appear later, and viewing information about the speed of a particular object at a certain point in time. That is, given that one of the main goals of this research is to make the tracking process fast and efficient, the following tasks must be performed:

– to study information about methods of tracking objects based on computer vision, their basis for further design;

– determine the main functional blocks of the automated system and its software architecture;

– prepare a dataset and train a neural model on it to further use it to recognize objects in the selected video;

– implement algorithms for tracking the movement of objects to determine their trajectories and speed.

Accordingly, the results of the work will be of practical importance to people who need tools for the analysis of video streams and to those interested in the field of computer vision. The developed system opens up wide opportunities for the development of the process of tracking objects in unmanned vehicles in order to improve their efficiency and safety in various areas where the use of drones is gaining popularity today, which in turn makes this technology very valuable and promising.

The specifics of the object tracking task

Object tracking requires the ability to link detected objects between video frames. This task is more complex than a simple detection process that only aims to localize the objects of interest in the image and generate a list of output bounding boxes. Tracking algorithms, for their part, provide for the consistent assessment of the position, size and orientation of the object in several frames, and are aimed at preserving the identity of the object in time in order to carry out its continuous monitoring [1].

In addition, there are a number of obstacles that can hinder the effective operation of tracking algorithms. For the most part, all limitations regarding the implementation of such

methods are related to the analysis of very complex scenes. That is, accurate tracking of all objects is difficult in cases where there are many objects in the frame, in addition, the model may lose the tracked object after it becomes partially hidden or temporarily leaves the frame. As a result, if the object was lost or incorrectly marked, incorrect tracking occurs.

Depending on the number of tracked objects and the ability to work at different levels of detail, two main approaches are distinguished: Single Object Tracking (SOT) and Multiple Object Tracking (MOT).

The Single Object Tracking (SOT) algorithm is focused on tracking one selected object in a video stream [2]. The main goal is to precisely find the location of this object in each frame after its initial detection. SOT creates bounding boxes that are given to the tracker based on the first frame of the input image. The tracker is then tasked with finding this unique object in all other frames.

Multiple Object Tracking (MOT) is a technique used to track multiple moving objects similar to our visual system. In general, MOT works on the same principle as SOT. However, as the name implies, multi-object tracking identifies and tracks not one object, but multiple objects in a video. MOT algorithms use large training data sets to understand moving objects. As a result, after training, they can identify and track multiple objects in each frame.

Multiple Object Tracking is called detection tracking, where bounding boxes identifying targets in video frames control the tracking process [3]. These detections are related to maintaining the same identifiers for the same targets on different frames. Many MOT datasets contain predefined notations, allowing algorithms to bypass detection and focus solely on comparing the quality of associations.

In addition, MOT is characterized by using a combination of object detectors (e.g. YOLO, Faster R-CNN) and tracking methods such as DeepSORT, JDE and ByteTrack. However, these algorithms are computationally expensive and take a long time to train.

Tab. 1 highlights a comparison of SOT and MOT approaches, highlighting their features, applications, challenges, and technical aspects.

Table 1. **Comparison of SOT and MOT approaches**

Parameter	**Single Object Tracking**	**Multiple Object Tracking**
Goal	Tracking of one selected object	Tracking multiple objects at the same time
Number of objects	Only one	Can be a lot
Usage scenarios	Video surveillance of a specific person	Counting people, traffic analysis

 ISSN 1560-8956

End of the table

Parameter	Single Object Tracking	Multiple Object Tracking
Complexity of algorithms	Relatively simple	More complex ones require association with personnel
Types of objects	One type of object, often known in advance	Several types of objects change over time
Temporal coherence	Consistency of one object through a sequence of frames	Requirement of coherence of many objects through sequence of frames
Definition of the object	The object is selected at the beginning and tracked	Objects are identified and tracked automatically
Capacity	Higher	Lower

Description of functional modules and system architecture

At the beginning of the development, each module of the future system was defined, each of which performs specific functions and interacts with other modules to ensure integrated operation. Figure 1 shows the interaction of the specified modules of the automated object tracking system. It can be seen that the video display module must receive the video stream and transmit it to the object recognition module. This module, in turn, must analyze the provided data, highlight objects and transfer information to the tracking module, which sends the updated coordinates to the characterization module. The next module processes information from the recognition and tracking modules, determines additional characteristics of objects and passes all this data to the user interface. The user interface displays information about objects, allows you to view the video stream, configure system parameters and interact with all other modules.

Accordingly, each unique module is implemented using the Python programming language and its libraries, including OpenCV and Ultralytics. The first is responsible for the direct processing of images in methods of interaction with the video stream, and the second is for direct communication with the selected YOLOv8 model, based on which the recognition and tracking algorithm was developed.

Dataset preparation and working with the model

In the context of the investigated problem, a properly prepared dataset is almost the most important factor for ensuring the accuracy and reliability of the model. In order to create an effective and representative dataset, several sources were chosen, including open datasets from Roboflow and direct manual data collection, which included taking photos of objects that needed to be further trained to recognize the model, this consisted of static images from different angles and conditions lighting.

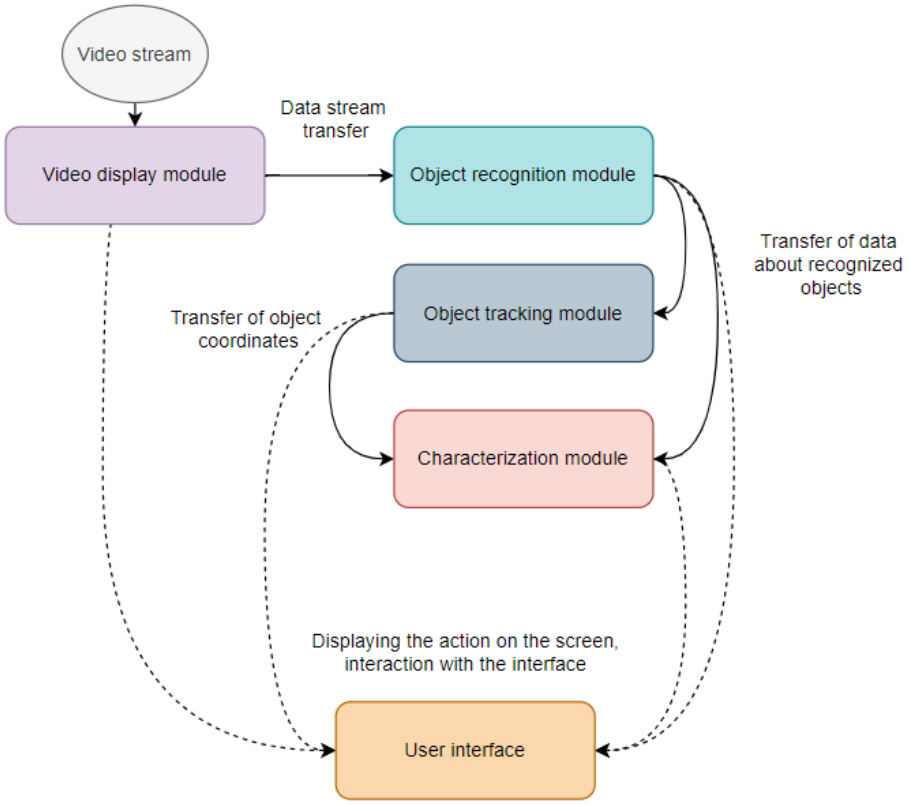

*Figure 1.*Interaction of system modules

Next, an equally important step was data preparation, which involved unifying the size and color profiles of the images, removing low-quality images, and increasing the diversity of the data using augmentation methods such as rotations, scaling, and brightness changes.

As a result, the images were organized into three separate sets: training, validation, and test. Accordingly, the training set is used to train the model, and its size is usually 60-70% of the total dataset [4]. The validation set is for tuning the hyperparameters and evaluating the model during training, it is about 20-15% of the total number of images. The test set is created for the final evaluation of the model’s performance after completion of training, while it is usually 15-10% of the total size of the dataset and is completely independent of other sets.

The distribution of images by sets is presented in Tab. 2.

Table 2. **Number of images in each set**

Set	**Training**	**Validation**	**Test**
Number of images	1937	652	375

Previously, the YOLOv8 model was chosen as the basis for building the main algorithms of the system. To ensure an ideal balance between detection accuracy and data processing time, it

 ISSN 1560-8956

was decided to use its subversion YOLOv8-m [5]. This model has an average value of FPS (Frames Per Second), which provides a sufficient speed of operation and a high indicator of mAP (Mean Average Precision), which indicates an acceptable accuracy of detecting objects in difficult conditions, which is critical in the context of this system, where it is important minimize the number of false positives and missed objects during tracking.

Fig. 2 shows examples of the final markup of each of the classes contained in the created dataset.

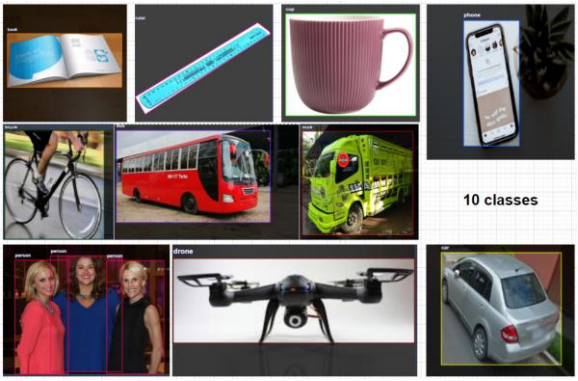

Figure 2. Examples of markings on different classes of the created dataset

In turn, retraining of this model was carried out, where YOLOv8-m kept the results for 100 epochs, including the learned weights and evaluation metrics.

After completing the retraining process, key artifacts stored by the model are analyzed, which contain information about training, model performance, and process visualization. That is, it includes trained weights, training and validation metrics, and various graphs (Fig. 3).

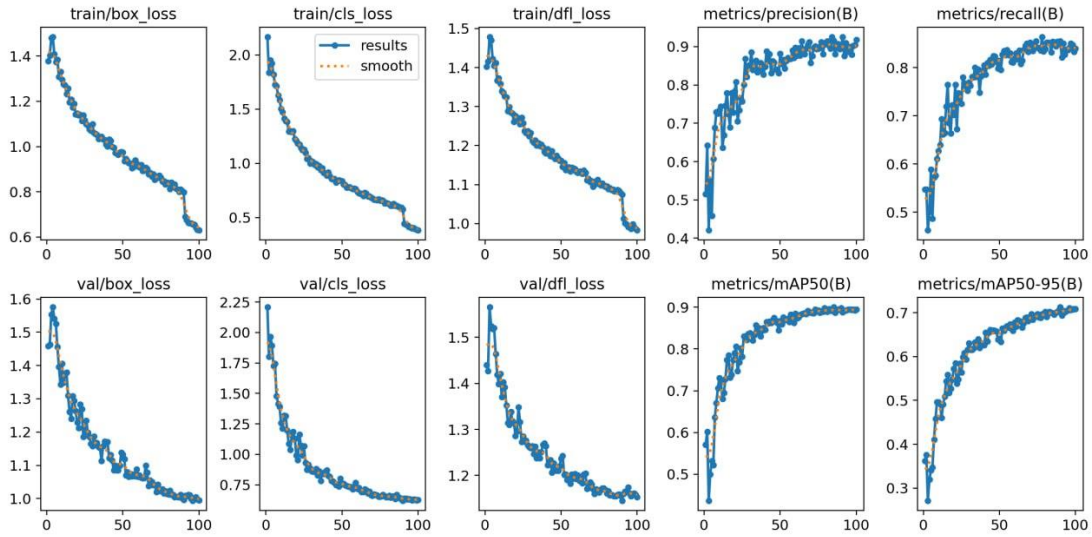

Figure 3. Results of retraining the model

From the obtained results, it can be seen that the precision metric (Precision) reaches a rather high value - 0.92, while the completeness (Recall) is 0.87, and mAP50 and mAP50-90 reached values of 0.9 and 0.71, respectively. Such indicators indicate effective training of the model.

Implementation of the object recognition algorithm

In general, the developed recognition algorithm finds and localizes objects in a video frame, which includes the identification of all 10 classes of objects, for example, cars, people, bicycles, etc. After detection, the algorithm allocates to which class each object belongs and returns the obtained result (Fig. 4).

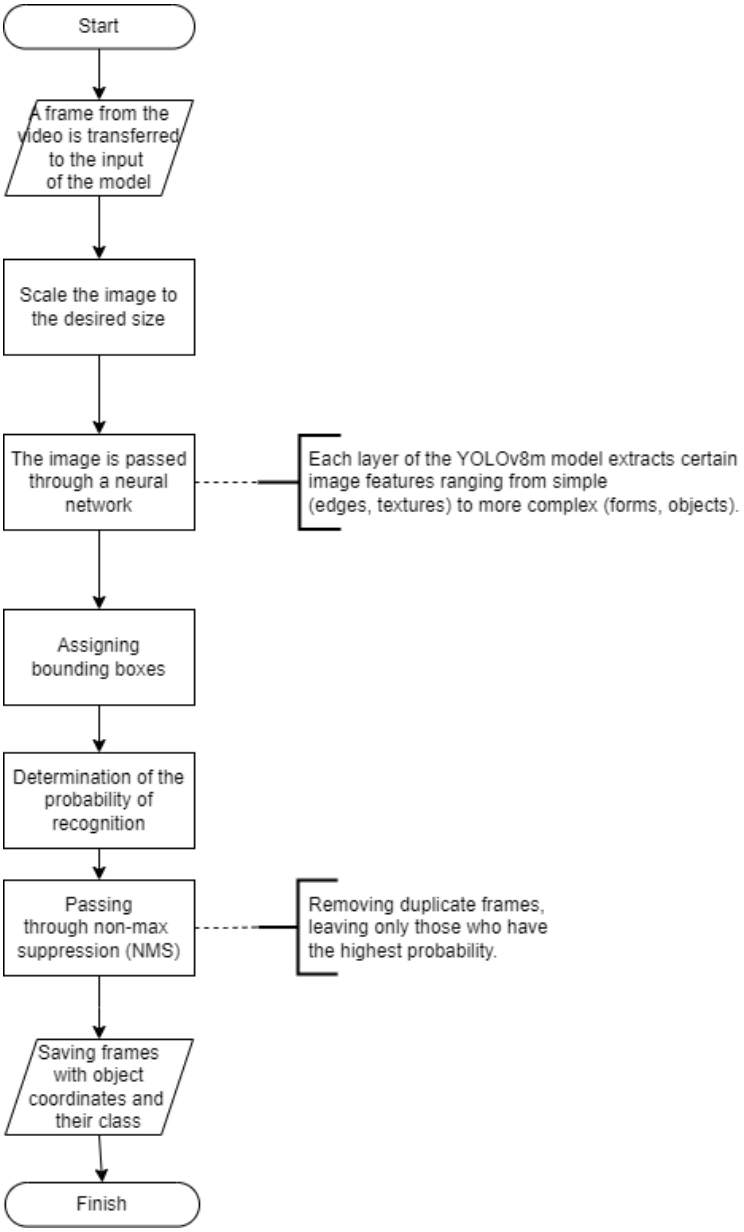

Figure 4. Block diagram of the recognition algorithm

 ISSN 1560-8956

Implementation of the object tracking algorithm

The object tracking algorithm (Fig. 5) is implemented using the track, getSelectedTrackedId, getTracks, getElementID methods and is the basis of the automated system tracking module. Accordingly, the track method is responsible for tracking the selected object in the current video frame. It checks whether the object is tracked (box.is_track) and whether its identifier corresponds to the selected ID (self.selectedID). In addition, it is in this method that a bounding box is drawn on the selected object and its center is calculated, an inscription from the object ID is added, and the tracking process of the target object is logged. The getSelectedTrackedId method returns the ID of the object selected for tracking, and getTracks accordingly returns all object tracks in the current frame.

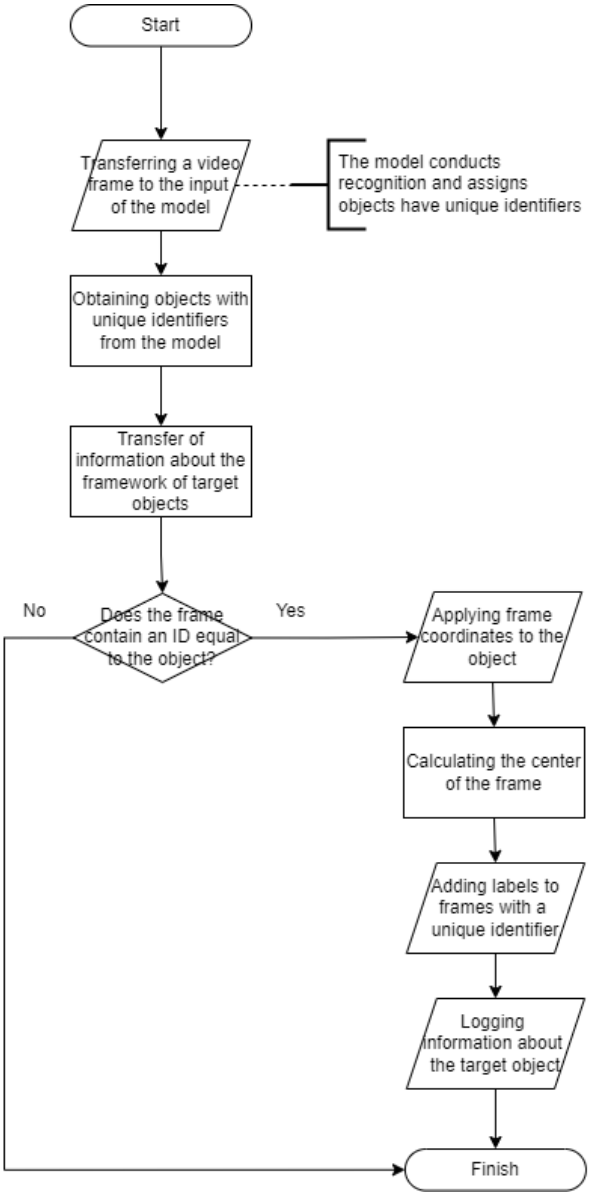

Figure 5. Block diagram of the object tracking algorithm

The created algorithm is based on both approaches to tracking objects: SOT and MOT, however, in this implementation, it allows you to select and track one specific object, while assigning unique identifiers to all detected objects. This way of combining the two approaches provides the possibility to easily expand the algorithm in the future to solve specific tasks, including tracking both one object and many at the same time.

The speed determination algorithm

In this implementation, the calculations are done on a pixel-by-pixel basis, so this can be called a rough estimate, which may vary depending on the speed of the GPU, but still allows you to determine certain results for analysis. This algorithm involves tracking objects using their identifiers, each of which consists of a sequence of coordinates (centers). The YOLOv8m model participates in assigned IDs as previously described [6]. These identifiers are stored in a dictionary where the key is the track identifier and the value is a list of object center coordinates. The distance traveled by the object is calculated as the difference in the coordinates of the object between the current and previous frames. This can be expressed by the formula:

$$\Delta y = y_{cur} - y_{prev} \quad (1)$$

where y_{cur} and y_{prev} – the vertical coordinates of the center of the object in the current and previous frames.

At the same time, the time difference between frames is calculated as:

$$\Delta t = t_{cur} - t_{prev} \quad (2)$$

where t_{cur} and t_{prev} – the time the object was at the current and previous positions, respectively.

As a result, the speed of the object is determined by the traditional formula of the ratio of the obtained traveled distance to the calculated time:

$$v = \frac{\Delta y}{\Delta t} \quad (3)$$

So, the implemented calculate_speed method calculates the speed for each track if the object is in the limiting region and its previous time is not equal to zero. The result is stored in the dictionary with the key as the track ID. This algorithm has been integrated into the characterization module so that the user can, if necessary, find out the approximate speeds of the tracked objects, such as vehicles.

Description of results

This system implements functionality that allows you to choose a source for viewing video data: from the camera or from a file. For tracking "from the camera", the program algorithm sends a request to connect to the device’s webcam, checks for the availability of the webcam and connection to it. After that, it is possible to choose an object for tracking (Fig. 6).

 ISSN 1560-8956

Now consider the situation of selecting a video from local storage, i.e. "from a file". The user can choose any video in .mp4 .avi .mkv formats, while the recognition algorithm instantly activates on the video stream and marks detected objects with frames (Fig. 7).

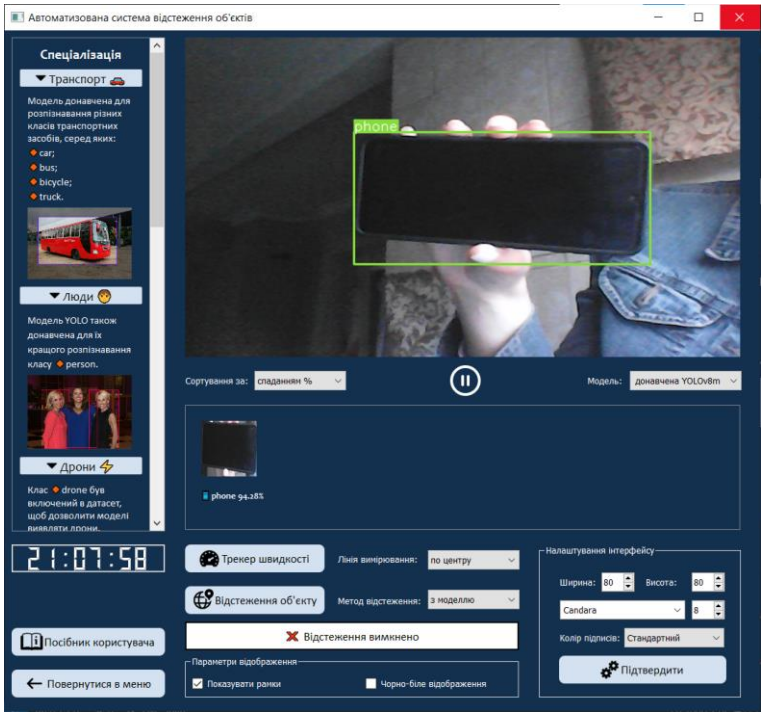

Figure 6. Tracking the "phone" class object from the device’s camera

Figure 7. Tracking objects on local video

Fig. 8 shows the moment of measuring the approximate speed of vehicles. It is worth noting that in this case, with the help of a special drop-down list, the user can change the location of the measurement line relative to which the path traveled by the objects is calculated.

Figure 8. Speed measurement process

It is worth noting that the developed tracking algorithm provides not only the assignment of a unique identifier to objects, but also the ability to remember the target object and re-track it when it appears in the camera’s field of view. That is, if the user chose an object that somehow overlapped with others or disappeared for a while, the system will wait for its appearance until a new target is selected. Such functionality significantly expands the range of possible solved problems, because the standard trackers of the OpenCV library are not able to remember the object and are not effective in conditions of overlapping objects or sharp camera rotation [7].

Accordingly, a study was conducted using the camera, comparing the implemented algorithm and the work of the standard CSRT tracker from the OpenCV library [8].

In this experiment, the object of tracking is "person", to which the algorithm of the automated system assigned the unique identifier 1.0. Tab. 3 presents the results of the conducted experiment.

That is, the standard tracker expected that the target object should appear from the same side where it was lost, so it failed to focus on the studied object, which is a disadvantage of using conventional trackers and an important factor that forces to look for alternatives and implement more effective tracking algorithms, like the algorithm of this system.

 ISSN 1560-8956

Table 3. **The results of comparing the performance of the developed tracking algorithm with the standard CSRT track**

Indicators	Implemented algorithm	OpenCV’s CSRT tracker
Assignment of a unique identifier	The object of the "person" class was assigned the unique identifier 1.0	It is not possible to assign a unique identifier
Imposition of restrictive frameworks	A frame around the target object is present	A frame around the target object is present
Stability of work when the object is partially covered	The algorithm continues the tracking process when the object is partially covered	The tracker loses the object when it is partially overlapped and does not keep focus on it
Work during the temporary disappearance of the object from the field of view	The algorithm waits for the object to appear, saving its unique identifier	The tracker looks for fixation points, expects the object to return to the same place
Work after returning the object to the camera’s field of view	The algorithm performs the correct tracking and reproduces the unique identifier 1.0	The tracker is unable to focus on the target object, tracking does not continue

Conclusions

The introduction of video surveillance systems, the increase in the total volume of video data that needs to be processed and the growth of security needs necessitate the development of an effective tool for automated work with video. At the same time, the question of the object tracking process is acute, because traditional methods, which are based on the manual analysis of a large number of frames, require significant human resources and time.

To solve this problem, a system architecture consisting of five interconnected modules was designed. Using the YOLOv8 model and advanced computer vision tools, a software solution was implemented, the main functionality of which includes object recognition, determination of their approximate speed, and tracking of a specific target object.

The developed system satisfies the set requirements and the main goal of the work — the automation of the object tracking process. It is important that, in comparison with standard trackers, the main algorithm of the system demonstrates high performance and the ability to work in difficult conditions, which is confirmed by the conducted experiments.

REFERENCES

1. *Ghedia, N., Vithalani, C., Kothari, A. M., Thanki, R. M.,* "Moving Objects Detection Using Machine Learning" Springer International Publishing, Switzerland, 2022, pp. 65-68

2. *Ajantha Devi Vairamani, Anand Nayyar, Ashish Kumar, Rachna Jain,* "Object Tracking Technology: Trends, Challenges and Applications", Springer Nature Singapore, Germany, 2023, pp. 1-23

3. *Xu, N., Lin, W., Lu, X., Wei, Y.,* Video Object Tracking: Tasks, Datasets, and Methods, 2024, pp. 75-79

4. Train Test Validation Split. website URL: https://www.v7labs.com/blog/train-validation-test-set

5. YOLOv8: A state-of-the-art object detection model. website URL: https://medium.com/@muradatcorvit23/yolov8-a-state-of-the-art-object-detection-model-2bdc58daa04e

6. Intelligence of Things: Technologies and Applications: The Second International Conference on Intelligence of Things (ICIT 2023), Ho Chi Minh City, Vietnam, October 25-27, 2023, Proceedings, Volume 2. (nd), p, 70

7. *Brahmbhatt, S.* (2013) Practical OpenCV (pp 3-5)

8. *Fazilah Hassan, Mohamad Hafis Izran Ishak, Mohamed Sultan Mohamed Ali, Mohd Ariffanan Mohd Basri, Mohd Saiful Azimi Mahmud, Noorhazirah Sunar* (2023) Methods and Applications for Modeling and Simulation of Complex Systems: 22nd Asia Simulation Conference, AsiaSim 2023, Langkawi, Malaysia , October 25–26, 2023, Proceedings, Part I, p, 24

 ISSN 1560-8956