

O. Koval¹, N. Bulbotka²

¹V. M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Ukraine
40, Academician Glushkov ave., Kyiv, 03187

^{1,2}National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine
37 Beresteysky Avenue, Kyiv, 03056

¹t70000@gmail.com

²Bulbotka.Nadiia@iit.kpi.ua

¹<https://orcid.org/0000-0002-9265-2748>

²<https://orcid.org/0009-0009-3750-3270>

INTELLIGENT VIDEO ANALYTICS MODULE FOR THE REGIONAL SITUATIONAL CENTERS

Abstract. The article explores the theoretical and practical aspects of designing a video analytics complex, the purpose of which is to continuously monitor and track visual targets in dynamic scenes. A thorough critical review of software solutions and open algorithmic paradigms available on the market has been carried out, which has allowed us to argue the need for the development of optimized tools. The foundation of the developed system is the YOLOv8m convolutional architecture, adapted for localization of entities in complex environmental conditions, including changes in illumination, scaling of the target and its short-term overlap with other objects (occlusion). The mathematical apparatus underlying the algorithm for frame-by-frame estimation of the approximated speed of movement of detected elements is detailed, and the mechanisms for saving their identifiers are also disclosed. An intuitive graphical user interface has been designed, which provides the possibility of interactive processing of both pre-saved media files and streaming video data from webcams in real time. The results obtained and the software product have potential for integration into security systems, traffic monitoring systems, and can also serve as a basic visual navigation module for unmanned aerial vehicles.

Keywords: computer vision, object tracking, convolutional neural networks, YOLO architecture, video analytics, detection algorithms, unmanned aerial vehicles, mathematical modeling, situation centers.

Introduction

The rapid development of artificial intelligence methods and the increase in available computing power have radically changed approaches to visual information processing. Today, one of the most important and most popular areas in the field of machine vision is the creation of reliable systems capable not only of finding certain classes of objects in an image, but also of continuously tracking their movement in a video stream.

Unlike classical detection, which treats each frame as an independent unit of information, tracking technology requires the establishment of temporal and spatial relationships. The system must "understand" that the object found in the current frame is the same entity that was captured milliseconds earlier. This task is complicated by a variety of real-world factors: changes in perspective, sharp fluctuations in illumination, motion blur during rapid movement, and the overlapping of the target by foreign objects.

The relevance of such developments is difficult to overestimate. They are critically important for the functioning of modern security systems, smart cities, automation of

logistics processes and, which is especially important in modern realities, for autonomous navigation systems of unmanned aerial vehicles (UAVs). In this context, there is a need to create software complexes that would harmoniously combine the high accuracy of neural network detectors with ease of deployment and ease of use.

Problem statement

The fundamental difficulty of automated dynamic scene analysis lies in the high level of entropy of the visual environment. When developing tracking systems, engineers face a number of classic problems:

1) ID Switch: a situation where the system confuses two visually similar objects that intersect in the frame and assigns them different numbers;

2) track fragmentation: the temporary loss of an object due to a change in its pose or lighting, which results in the creation of a new identifier for the same entity upon its re-detection;

3) computational redundancy: using too heavy neural networks makes real-time video processing impossible on portable devices.

Despite the large number of scientific works in this area, there is a significant shortage of ready-to-use software tools that would provide a full cycle of analytics (from detection to velocity calculation) without the need for complex programming on the part of the end user and would not weigh too much. Therefore, the development of an integrated solution that combines a modern mathematical basis and a clear graphical interface is an extremely urgent task.

The purpose of the study

Accordingly, the main goal of this work is the design, mathematical justification and software implementation of an intelligent machine vision system. The created complex should provide three basic functions: accurate classification and localization of given entities, stable assignment and maintenance of unique tracking identifiers, as well as calculation of kinematic parameters (velocity) of target movement. A separate super-task is the development of an ergonomic graphical user environment (GUI), which will allow the operator to flexibly configure analysis parameters and obtain visualized results without interfering with the program code.

Analysis of existing solutions

The rapid evolution of artificial intelligence algorithms has opened up fundamentally new horizons for deep analytics of video streams. One of the most complex and promising areas in this area has been the design of complex machine vision methods focused on continuous tracking of targets. The main task of such systems is to accurately detect certain categories of objects in the input image with the subsequent generation of their exact movement routes.

It is worth noting that the monitoring procedure is significantly different from classical detection. While conventional recognition only localizes the target, creating a bounding box for a specific frame, full-fledged tracking requires establishing logical connections between these frames throughout the entire video recording. Trackers must consistently evaluate the spatial position, dimensions, and orientation of the element, striving to keep its unique identifier

unchanged. There are a number of serious obstacles on the way to creating reliable algorithms. The greatest difficulties arise when processing visually overloaded scenes, where there is a significant number of similar small details. In addition, the mathematical model often loses its connection to the target at the moment of partial visual overlap (occlusion) by other objects or when the target disappears from the field of view of the video camera for a certain time. In such situations, the system is threatened with complete loss of the object or erroneous assignment of its label to another element. It should also be taken into account that the preparation of neural network architectures for solving such complex tasks requires enormous costs of computing power and working time of specialists.

Depending on the number of targets and methods of processing visual information, modern approaches are divided into two main branches: Single Object Tracking (SOT) and Multiple Object Tracking (MOT).

The SOT (Single Object Tracking) strategy focuses on continuous monitoring of only one selected target. This approach is classified as "tracking without prior detection" because it requires manual labeling of the desired element in the starting image (template initialization). After that, the algorithm, using correlation filters or analysis of unique features (color, texture, contours), searches for this template in subsequent "search areas". Such models are relatively computationally simple, provide high throughput, but they are limited to working with only one known target type and require strict time consistency.

Instead, MOT (multi-object tracking) operates on the principle of "tracking through detection". The mechanism is able to simultaneously identify and track a large number of moving objects of different classes. The process is based on a combination of powerful detectors (e.g. YOLO architectures) and specialized trackers that associate the found bounding boxes with each other to maintain stable identifiers over time. Although MOT algorithms are ideally suited for complex scenarios (transport traffic analysis, crowd counting), they are computationally heavy and exhibit lower overall throughput compared to SOT systems.

Analysis of the market for ready-made commercial solutions revealed several popular platforms, each of which has its own specifics and limitations.

Sighthound Video is a powerful video analytics tool that allows security cameras to simulate human perception, recognizing people, vehicles, and animals. The system supports the creation of intelligent rules for sending notifications and 24/7 recording of high-resolution streams. However, full functionality is hidden behind an expensive license, and storing video data arrays locally places critically high demands on computer hardware and storage capacity.

Camlytics is a software package designed to aggregate events from network recorders. Its strengths are building heat maps, counting line intersections, and anonymizing metadata before sending to the cloud. However, the free license significantly limits its capabilities. The most critical drawback for our task is the lack of a mechanism for selecting a specific single target for personal support, and the interface itself is too overloaded for the average user.

SynthEyes is a highly specialized environment for 3D camera tracking and geometry, which is actively used by visual effects (CGI) specialists. The software generates 3D meshes based on the points found, adapting to changes in perspective and scale. Despite its unprecedented accuracy, this tool is too complicated to learn. It requires mandatory special training courses, is expensive, and its trial version completely blocks the tools for tracking objects and estimating their speed.

Kapwing is a cloud-based multimedia platform with basic tracking features. Its capabilities are limited to the banal "attachment" of static elements (text or graphics) to a moving point on the screen. This service does not have complex analytical tools, requires a continuous broadband Internet connection, and the basic tariff limits the duration and size of downloaded files.

Methodological basis and software architecture

Taking into account the analyzed shortcomings of existing software products, it

was decided to design our own automated system. The foundation for pattern recognition was the deep neural network YOLOv8m. The process of adaptation (retraining) involves optimizing the weight coefficients, which are subsequently saved in the .pt format for direct integration into the program code.

To adequately understand the performance of the model on training and validation samples, a classic set of metrics is used. All statistics are based on four fundamental indicators: True Positive (*TP*) indicates the number of absolutely correct recognitions; False Positive (*FP*) records the number of false, erroneous detector activations; False Negative (*FN*) indicates those real targets that the mathematical model missed and could not detect; True Negative (*TN*) refers to background areas that the system correctly classified as empty.

The degree of accuracy of the model (its ability not to generate "garbage" hypotheses) is assessed by the *Precision* metric. It is calculated as the mathematical ratio of correctly guessed objects to the total sum of all system activations:

$$Precision = \frac{TP}{TP + FP}$$

Instead, the completeness metric (*Recall*) measures the overall sensitivity of the architecture, demonstrating what fraction of all existing ground truths the algorithm was able to successfully find:

$$Recall = \frac{TP}{TP + FN}$$

To obtain a comprehensive assessment of the detection quality for all predicted object classes at once, the mean average precision (*mAP*) indicator is used:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i$$

where is a variable n represents the total number of categories on which the neural network was trained. An extremely important stage of control during training is the analysis of loss functions. They quantitatively demonstrate the mathematical discrepancy between ideal annotations and real predicted

frames, which allows us to objectively judge the success of the training process.

In general, the object recognition process follows the algorithm shown in Figure 1.

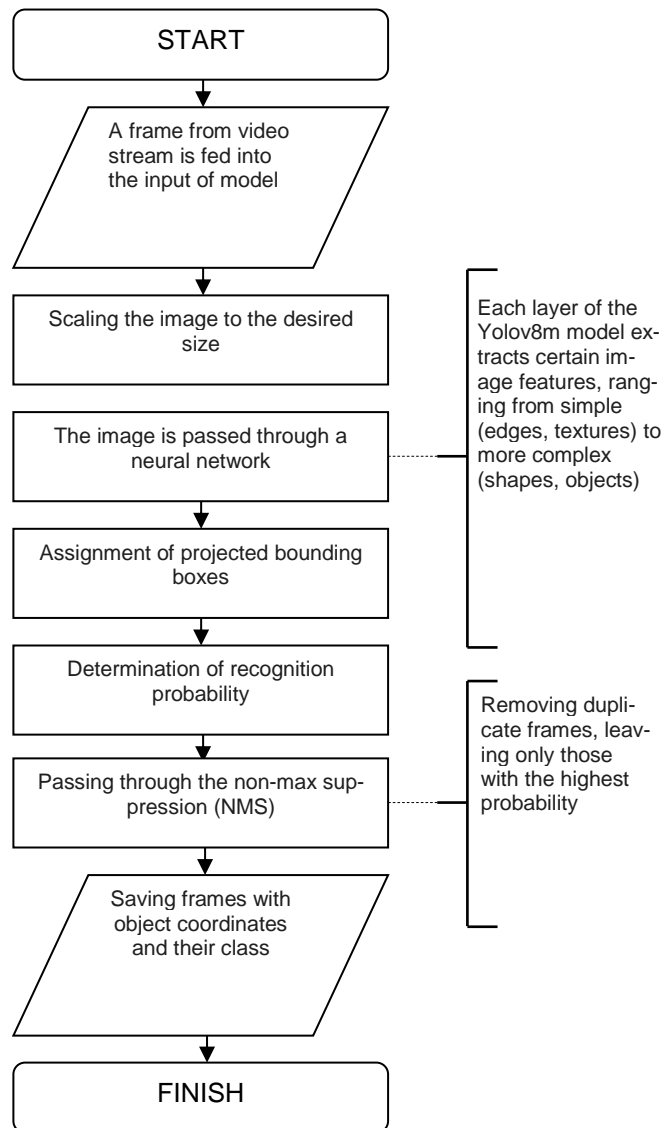


Fig. 1. Flowchart of the recognition algorithm

Tracking and speed calculation algorithms

The developed system operates with five interconnected modules. The main tracking algorithm is responsible for assigning unique identification numbers (ID) to targets, each of which is rigidly tied to the sequence of coordinates of the center of the found element. This data is structured in a special dictionary, where the key is the ID number itself, and the value is an array of spatial labels.

The characterization block is responsible for calculating kinematic parameters (Fig. 2). It should be clearly understood that the speed

calculated when analyzing 2D video is an approximate value, since the system does not have information about the real depth of the three-dimensional world and performs all calculations exclusively in the pixel plane.

The algorithm works as follows: the distance traveled by an object is calculated as the geometric difference of the spatial coordinates of its center between the current and previous frames of the video stream (Fig. 3). Mathematically, this logic is expressed by the Pythagorean theorem for two-dimensional space:

$$d = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2},$$

where x and y – vertical and horizontal coordinates of the element center at the corresponding points in time. The time difference between these frames is calculated by a simple mathematical subtraction operation:

$$\Delta t = t_t - t_{t-1},$$

where t indicators record the exact time an object spends at specific points in space.

Accordingly, the desired speed V of the object is calculated using the classical traditional formula - as the ratio of the pixel distance traveled to the recorded time interval:

$$V = \frac{d}{\Delta t}.$$

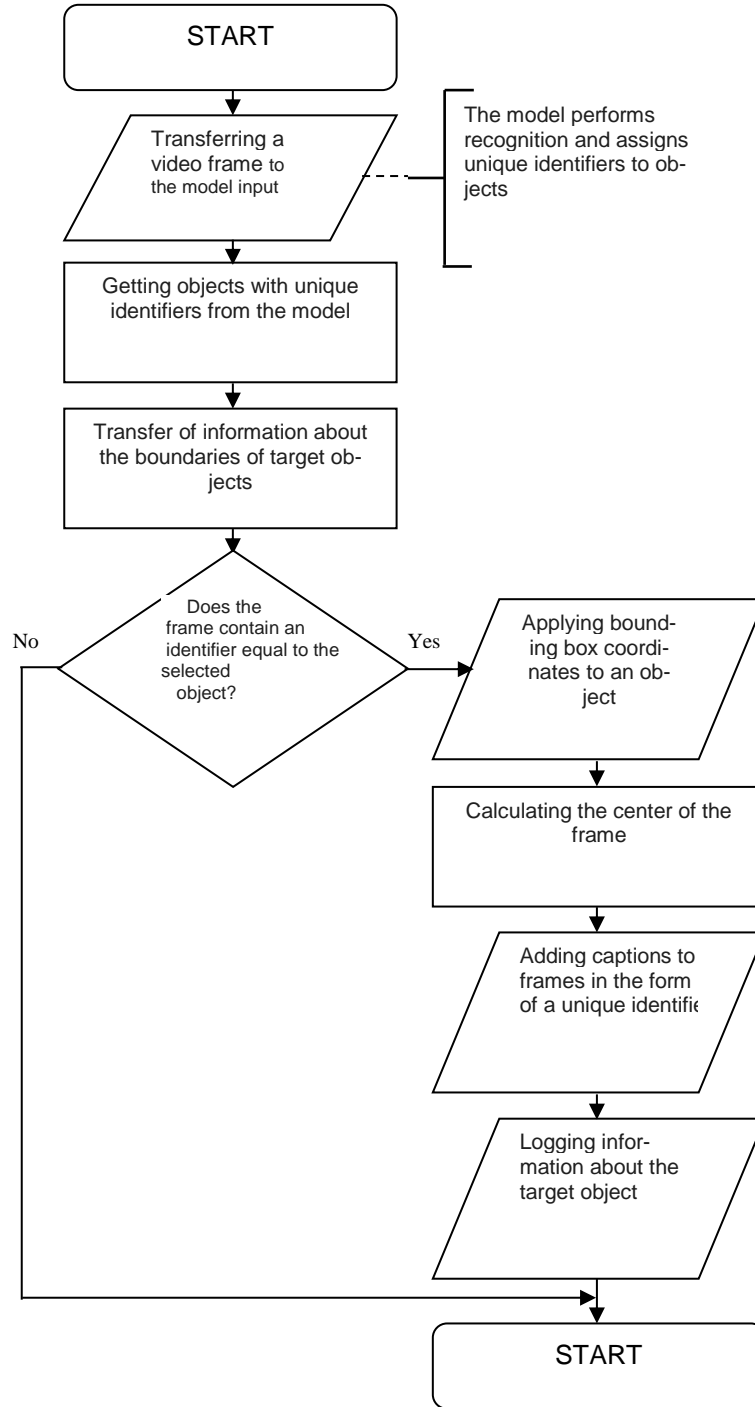


Fig. 2. Flowchart of the object tracking algorithm



Fig. 3. Speed measurement process

User interaction and processing of complex scenes

The user interface was designed with an emphasis on intuitive clarity, which eliminates the need for long-term training of personnel. The software package supports flexible selection of the source of visual data: analysis can be carried out both on the basis of pre-loaded local files and in real time by capturing the stream from a webcam. The main interaction window displays a video sequence, on which auxiliary graphics are superimposed, and a list of detected targets is also formed with the ability to sort them by the percentage of detector confidence

In addition to basic visual customization (changing fonts, colors, frame sizes), the program provides high resistance to stressful shooting conditions. The most difficult situation for any tracker is occlusion - the moment when the tracked element is completely or partially covered by obstacles. Thanks to the implemented predictive analytics algorithms, the system is able to continue virtual tracking of the element even during its overlap, remembering the vector of its previous movement.

When the target temporarily leaves the camera's field of view, the program goes into standby mode. As soon as the object reappears within the frame, the system instantly identifies its features, matches them with the saved templates, and automatically resumes tracking, continuing to record logs for further analysis.

Thus, the developed software package offers a universal environment for multimedia processing, which effectively solves the tasks

of category recognition, dynamic calculation of kinematic properties, and stable target retention under various environmental conditions.

Development of the concept of regional situational centers through the implementation of intelligent video analytics modules

The effective functioning of modern intelligent control systems, in particular the concept of "CONTROL_TEE" (Control of Technical-Ecological Events), described in detail in the works [1, 2, 3, 4, 5], requires the creation of a two-level information processing structure. According to this concept, the organizational basis of the system is based on a network of decentralized regional situational centers (RSCs), which act as coordinators between service customers (municipal services, private structures) and unmanned aerial vehicles (UAVs) as direct performers of monitoring missions.

Petri net

To analyze an intelligent system for managing techno-ecological events using UAVs (and drones of other bases), it can be represented as a discrete-event system. The Petri net in this context describes the logic of interaction between physical objects (drones), data flows and decision-making centers. Below is a formal description of the network elements:

$$N = (P, T, F, M_0),$$

where P – set of states, T – set of transitions, F – arcs (connections): functions of input and

output arcs, M_0 – initial marking $M_0 = (1, 0, 0, \dots, 0)$, what does one token mean in P_1 (for example, the system is ready to launch a mission: drones are ready to operate, the system is in normal mode).

1) Set of positions (States - P).

Positions characterize the conditions or availability of resources in the system nodes:

P_1 : Drone technology park (drones in standby/charging mode).

P_2 : Drones in flight (object monitoring/charging).

P_3 : Original data (raw data collected by sensors).

P_4 : Information repository (database/archive).

P_5 : Regional Situation Center (RSC) – processing stage.

P_6 : Main Situation Center (MSC) – strategic analysis.

P_7 : Package of management commands (decision made).

P_8 : Local incident (an event requiring attention).

2) Set of transitions (Events - T).

Transitions describe active actions or transformation processes:

T_1 : Departure (activation of a group of drones according to a schedule signal).

T_2 : Telemetry/video stream collection and transmission.

T_3 : Returning drones to base (mission completion/charging).

T_4 : Write to storage (data archiving).

T_5 : Expert analysis in the RSC (development of an operational solution).

T_6 : Transfer of critical data to the upper level (to the GSC).

T_7 : Formation of the final directive (GCC→Executors).

3) Logic of functioning (Arcs – F).

Relationships define the cause-and-effect structure:

1. (P_8, T_1) : Incident triggers departure.

2. $(P_1, T_1) \rightarrow P_2$: Drones move from the technology park to flight mode.

3. $P_2 \rightarrow T_2 \rightarrow P_3$: Data is generated during the flight.

4. $P_3 \rightarrow T_4 \rightarrow P_4$: Data goes into storage.

5. $P_3 \rightarrow T_5 \rightarrow P_5$: Data is sent to the RSC for analysis.

6. $P_5 \rightarrow T_6 \rightarrow P_6$: If the risk level is high, the GSC is involved.

7. $P_6 \rightarrow T_7 \rightarrow P_7$: The CSC issues the final management command.

Petri net visualization

On the diagram the positions P are usually indicated by circles, transitions T – are indicated by rectangles (thick lines), and the chips (markers) inside P show the current state of the system (for example, the number of free drones in P_1 or the presence of raw data in P_3). The identity matrix is also given.

Below in the diagram, the main cycle is closed through the return of drones to the technopark (T_3) and the implementation of commands (P_7), which allows the system to be adaptive and self-regulating. For example, the transition T_5 (RSC) (Fig. 4) may have an inhibitory arc from P_6 , if the GSC has already taken control in the event of a large-scale disaster.

Intelligent video analytics module

However, as the analysis of the experience of deploying such centers shows, the critical bottleneck remains the problem of automating the interpretation of large streams of visual data coming from UAV sensors in real time. It is in this context that the automated object tracking system based on computer vision proposed in this article appears as a logical stage in the development and functional content of the RSC.

General mathematical model of drone image recognition.

Below are the components of a generalized mathematical model of a drone pattern recognition system for the situation center technology park. The model is suitable for monitoring man-made, environmental, infrastructure and security events.

1) *Main functions of the system.* The functions of the information and analytical system consist of: receiving a data stream from drones, its pre-processing, feature extraction, image classification, spatio-temporal analysis, decision-making by the situational center.

2) *Incoming data stream.* Let's mark: $D(t) = \{V_t, T_t, L_t, S_t\}$, where V_t – video stream, T_t – thermal imaging data, L_t –

LiDAR/laser scanning, S_t – drone telemetry, t – time.

3) *Signal preprocessing*. For example, noise filtering is performed using Gaussian Filter, Kalman Filter, Median Filter:

$$X_f = F(X),$$

where X – gray data, F – filter operator, X_f – cleared signal.

Next, georeferencing is performed: each pixel is assigned a coordinate:

$$P_i = (x_i, y_i, z_i, t_i).$$

4) *Feature extraction*. A feature vector is formed for each object:

$$\Phi_i = [c_i, g_i, m_i, h_i, \tau_i],$$

where c_i – color characteristics, g_i – geometric parameters, m_i – movement parameters, h_i – thermal characteristics, τ_i – time characteristics.

5) *Neural network recognition model*. A classification is carried out, the function of which is denoted as:

$$Y_i = \mathcal{N}(\Phi_i, \theta),$$

where \mathcal{N} – neural network, θ – model parameters, Y_i – object class.

The next step is to calculate the probability of belonging to a class:

$$P(C_k | \Phi_i) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}},$$

where C_k – class, K – number of classes, z_k – logit neural network.

6) *Space-time model*. To analyze events, a graph is drawn:

$$G = (V, E),$$

where V – set of objects, E – connections between them.

We describe the dynamics of events in the states of the system as:

$$S(t + 1) = AS(t) + BU(t) + W(t),$$

where A – transition matrix, $U(t)$ – external influences, B – influence matrix, $W(t)$ – noise.

7) *Anomaly detection*. To calculate the deviation, we introduce the function:

$$\Delta_i = \|\Phi_i - \bar{\Phi}\|,$$

where $\bar{\Phi}$ – normal profile, Δ_i – degree of abnormality,

and alarm definition criterion:

$$Alarm = \begin{cases} 1, \Delta_i > \lambda \\ 0, \Delta_i \leq \lambda \end{cases}$$

where λ – alarm threshold.

8) *Situational risk model*. We denote the integral risk as:

$$R = \sum_{i=1}^N w_i p_i d_i,$$

where w_i – threat weight, p_i – probability, d_i – potential damage.

9) *Optimal response function*. The purpose of the situation center is to select the optimal response action U^* , which minimizes risk, cost, and response time. That is, the optimization objective function has the following form:

$$U^* = \arg \min_U [\alpha R(U) + \beta C(U) + \gamma T(U)],$$

where U – set of possible response actions, U^* – optimal response strategy, $R(U)$ – risk function after applying the action U , $C(U)$ – response cost, $T(U)$ – response execution time, α, β, γ – system priority weights.

If resources, probability, and efficiency are taken into account, we obtain an extended form of the model:

$$U^* = \arg \min_U [\alpha \sum_{i=1}^N p_i d_i + \beta \sum_{j=1}^M c_j + \gamma \tau + \delta E(U)],$$

where p_i – the probability of the threat developing, d_i – potential damage, c_j – expenditure of resources, τ – event localization time, $E(U)$ – response efficiency, δ – efficiency importance coefficient.

To take into account the system constraints, we denote them as follows: energy

$$E_i \leq E_{max},$$

temporary

$$T(U) \leq T_{crit},$$

resourceful

$$\sum_{j=1}^M c_j \leq C_{max}.$$

Let us give an example for a situation center technopark. For a fire, the set of possible response actions U can be written as:

$U = \{\text{reconnaissance drone, fire module, evacuation, zone blocking}\}.$

The system chooses the following combination of actions for the optimization objective function:

$$U^* = \arg \min [0.5R + 0.3C + 0.2T],$$

i.e.:

50% – risk minimization;

30% – cost minimization;

20% – speed of response.

10) *Total system cycle*. Thus, we obtain the total cycle of the system:

$$D(t) \rightarrow F(D) \rightarrow \Phi \rightarrow \mathcal{N} \rightarrow G(t) \rightarrow R(t) \rightarrow U(t).$$

11) *Typical object classes*. Example of a set of classes:

$C = \{\text{man, car, drone, fire, smoke, leak, destruction, crowd, explosion, anomaly}\}$

12) *Generalized objective function of the system*. Taking into account the above notations, we obtain the generalized objective function of the system:

$$J = \int_0^T (\eta_1 E_r + \eta_2 E_f + \eta_3 T_d + \eta_4 R) dt \rightarrow \min,$$

where E_r – recognition error, E_f – false positives, T_d – detection delay, R – the risk of the situation.

13) *System architecture*. We denote the UAV level as:

$$\text{Drone}_i = \{\text{Camera, Thermal, LiDAR, GPS, AI Edge}\}$$

Primary recognition on board by nodes (periphery) (Edge AI):

$$Y^{edge} = \mathcal{N}_{edge}(X).$$

Thus, we obtain a global model of the main situational center:

$$Y^{global} = \text{Fusion}(Y_1, Y_2, \dots, Y_n)$$

14) *Multi-drone cooperation model*. Adding a group of drones:

$$\Omega = \bigcup_{i=1}^N \Omega_i,$$

where Ω_i – inspection area by the i -th drone.

We denote the coverage optimization as:

$$\max|\Omega|$$

with restrictions

$$E_i < E_{max},$$

where E_i – power consumption of the i -th drone.

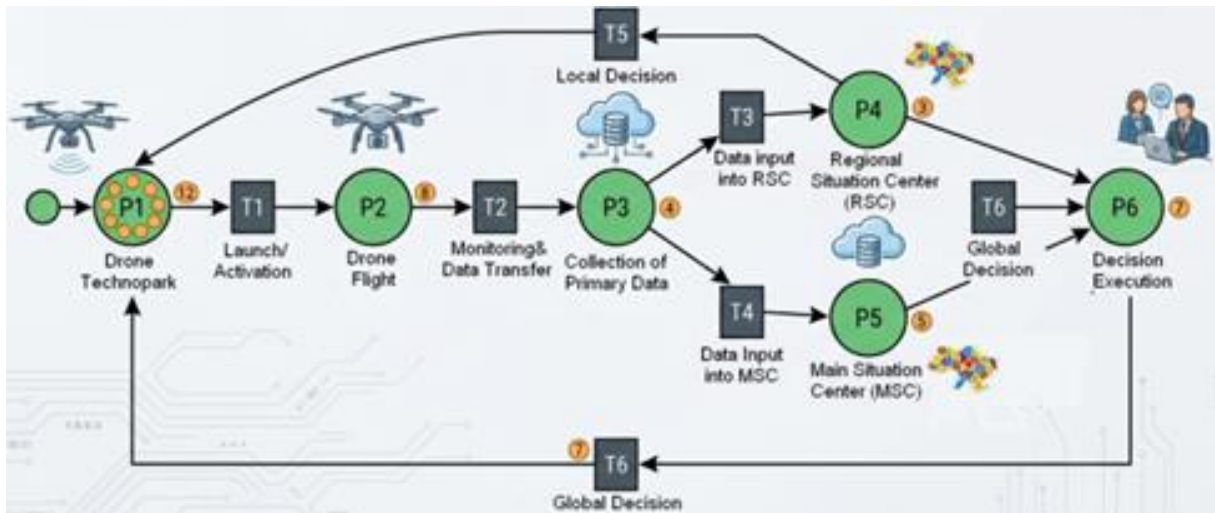


Fig. 4. Petri Net: Intelligent techno-environmental risk management system. Dynamic process «Drones – Situation Center-Decisions»

These components of the model are the basic platform for: AI monitoring of technology parks, city situational centers, civil protection systems, technical surveillance, environmental monitoring, autonomous security systems.

The implementation of YOLOv8 for detection and Deep SORT for stable tracking of targets [6] allows transforming RSCs from passive data collection centers into proactive operational response systems. In particular, the integration of the described video analytics module into the infrastructure of the “CONTROL_TEE” system provides the solution to the following strategic tasks:

1) *Automation of continuous monitoring*: Instead of manually viewing the video stream by the operator, the system independently identifies targets (vehicles, people, man-made objects) and keeps a log of them in the situation center database.

2) *Spatial coordination within a Smart City*: The use of tracking metadata allows for precise linking of dynamic objects to digital maps of urban infrastructure, which is the basis for the interaction of a "smart city" with a fleet of autonomous performers.

3) *Increasing the autonomy of drones*: The proposed algorithms allow implementing the function of intelligent “object tracking”,

which minimizes the need for constant remote control and reduces the load on the RSC communication channels.

The architecture of the pattern recognition subsystem is designed for the collection, transmission, processing, and intelligent analysis of data received from drones of different platforms within the regional situation center technology park. The system supports multimodal information sources, including video, audio, thermal imaging, motion sensors, smoke and gas detectors, temperature sensors, LiDAR, and other sensor types. The subsystem consists of several main layers: Data Acquisition and Transmission — provides collection of data streams from onboard drone systems, preliminary edge processing, encryption, and secure transmission through communication channels such as 4G/5G, SATCOM, Mesh, and LoRa networks. Pattern Recognition Subsystem — performs data normalization and filtering, applies artificial intelligence and machine learning models for object detection, classification, anomaly recognition, and event analysis. The subsystem supports multimodal data fusion from multiple sensor sources. Services and APIs — provide access to streaming data, analytics, event search, reporting, and integration with external systems of the regional situation center. Infrastructure Layer — includes cloud infrastructure, GPU/TPU computing resources, container orchestration, scalable storage systems, and load balancing mechanisms. Security and Governance — implements authentication, access control, event logging, backup, resilience, and cybersecurity compliance mechanisms. The system enables real-time detection of emergencies, technogenic threats, fires, hazardous material leaks, object movement, and other critical events, supporting rapid decision-making and operational response within the regional situation center.

Thus, if the works [1, 2, 3, 4, 5] lay the architectural foundation of situational centers and determine the structure of information storage, then the video analytics system presented in this work is an “intelligent core” that directly implements the function of image recognition and scene analysis. This creates a closed control loop: from setting a task in the

RSC to automatic detection and tracking of the object using computer vision.

Conclusions

As a result of the research, an automated visual object tracking system was successfully designed and software implemented. The use of the YOLOv8m architecture in combination with the mathematical apparatus of interframe association allowed us to create a tool characterized by high localization accuracy and resistance to such destructive factors as changing the scale of the target and its temporary spatial occlusion.

Unlike heavy commercial analogues, the developed complex is distinguished by an intuitive graphical interface, a low entry threshold for the user and minimal requirements for preliminary equipment setup. A detailed formalized mechanism for calculating approximated kinematics opens the way for further analysis of the behavior of objects in the frame.

The prospects for further development of the project are seen in two directions. First, it is the integration of algorithms for perspective transformation of the camera matrix, which will allow converting relative speeds into absolute metric values (km/h). Second, optimization of the program code using tensor computing technologies (for example, TensorRT) for its deployment on microcomputers. This will make it possible to use the developed system as a basic machine vision module “on board” unmanned aerial vehicles, ensuring their autonomous navigation, obstacle recognition at low altitudes, and performing specific tasks for monitoring territories.

References

1. Pysarenko, Y. V., Melkumyan, K. O., Varava, I. V., Koval, O. S., & Chumakova, N. F. (2022). On the organization of regional situational centers of the intellectual system “CONTROL_TEA” using UAVs. *Artificial Intelligence*, (1), PP. 275-281. <https://doi.org/10.15407/jai2022.01.275>
2. Pisarenko, J., & Melkumyan, E. (2019). The structure of the information storage “CONTROL_TEA” for UAV applications. *IEEE 5th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, 274-277. <https://doi.org/10.1109/APUAVD47061.2019.8943938>

3. V. G. Pisarenko, N. V. Nogin, A. S. Kryachok, J. V. Pisarenko, I. A. Varava, A. S. Koval. (2022) About complex intelligent technologies for technological events control in the water area. *Prombles in programming*, 3-4, PP. 437-445.

<https://doi.org/10.15407/pp2022.03-04.437>

4. V. G. Pisarenko, J. V. Pisarenko, O. E. Gulchak, T. I. Chobotok, A. G. Boyko. (2021) Practical experience in the technical systems creating with the artificial intelligence elements. *Artificial Intelligence*, (1), PP. 95-101.

<https://doi.org/10.15407/jai2021.01.095>

5. Pysarenko Yu. V., Karmazin K. V. Vykorystannia ShI dlia obrobky ta analizu velykykh ob'siahiv danykh u realnomu chasi z intehtratsiieiu

multyahentnykh system // *Artificial Intelligence*. – 2025. – № 4. – PP.134-145

<https://doi.org/10.15407/jai2025.04.134>

6. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Uppcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. – PP. 3464-3468.

<https://doi.org/10.1109/ICIP.2016.7533003>

The article has been sent to the editors 09.06.26.

After processing 20.06.26.

Submitted for printing 30.06.26

Copyright under license CCBY-SA4.0.