BULGARIAN ACADEMY OF SCIENCES
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2026 • Vol. **84**, pp. 46-59
p-ISSN: 2738-7356; e-ISSN: 2738-7364
https://doi.org/10.7546/PECR.84.26.04

Optimizing Moving Object Detection and Recognition in Video Surveillance Systems

Miroslav R. Tomov[1], Ivan G. Garvanov[1, 2]

[1] *University of Library Science and Information Technology, Sofia, Bulgaria*
[2] *Institute of Mathematics and Informatics, Sofia, Bulgaria*
Emails 4623501-1@unibit.bg, i.garvanov@unibit.bg

Abstract: This paper presents a hybrid system for real-time automatic object detection and classification, implemented within the MATLAB environment. The methodology combines statistical background modeling with deep neural networks to optimize computational efficiency. The initial segmentation is based on the Gaussian Mixture Model (GMM). To achieve high adaptability, an enhancement is integrated that allows for the automatic determination of the number of components for each pixel. The resulting binary mask is refined using the "Opening" morphological operation. Geometric feature extraction and the definition of Regions of Interest (ROI) are performed through topological analysis algorithms. These parameters allow the model to filter objects below a certain threshold and provide precise coordinates to the semantic classification stage. Final detection is carried out via the YOLOv3 algorithm, utilizing the Darknet-53 architecture. The system calculates bounding boxes through regression and determines confidence scores based on the Intersection over Union (IOU) metric. The proposed hybrid approach ensures reliable object recognition through synergy between classical signal processing methods and modern artificial intelligence architectures.

Keywords: Computer vision, object detection, Gaussian Mixture Model (GMM), YOLOv3, morphological operations, hybrid algorithms.

1. Introduction

In modern computer vision, real-time object detection and tracking require high accuracy and computational efficiency. One of the fundamental methods for

foreground segmentation is the Gaussian Mixture Model (GMM), the foundations of which were laid by [1]. Through this approach, each pixel at time *t* is modeled as a sum of *K* Gaussian distributions, allowing the system to adapt to environmental changes.

A significant improvement in this field was presented by [2]. His main contribution is the model's ability to automatically determine the number of components *K* for each pixel. This is achieved by maximizing the log-likelihood function with an added penalty term, which enables the dynamic 'switching off' of redundant distributions.

To refine the resulting binary masks, mathematical morphology methods are applied. The 'Opening' operation, based on the principles [3], is critical for eliminating small objects and noise. For the subsequent quantitative analysis of the regions, the topological analysis algorithms of [4], are used, allowing for the precise calculation of the area and bounding boxes of the objects.

For the final phase of detection and semantic classification within the hybrid model, the YOLOv3 algorithm is applied, as detailed by [5]. Utilizing the Darknet-53 architecture, this model treats detection as a regression problem, dividing the image into a grid of cells to simultaneously predict coordinates and class probabilities. The synergy between these classical statistical methods and modern deep neural networks forms the basis of the proposed software solution.

To address this challenge, high-performance systems for automatic object search and detection in video streams are required. When video footage is of low quality or affected by noise, specialized algorithms for deblurring or noise reduction must be applied [6, 7].

2. Object Detection via Gaussian Model

The motion detection process begins with segmentation based on the *vision.ForegroundDetector* system object in MATLAB. This object implements an advanced version of the adaptive Gaussian Mixture Model (GMM), the theoretical foundations of which were established by [1]. In this approach, each individual pixel at a specific time *t* is mathematically described by a sum of *K* Gaussian distributions:

$$P(X_t) = \sum_{k=1}^{K} \pi_{k(t)} \cdot N(X_t \mid \mu_{k(t)}, \sum_{k(t)}) \quad (1)$$

In this dependency, $\pi_{k(t)}$ represents the weight of the corresponding component, and N is the normal distribution defined by the mean vector μ_k and the covariance matrix $\sum$.

A critical improvement in the implementation is the model proposed by [2], which allows the system to automatically determine the optimal number of components *K* for each pixel. This is achieved by maximizing the log-likelihood function, to which a penalty term *C*, based on the Dirichlet criterion, is added:

$$L = \sum_{t=1}^{N} In\, P(X_t) - C \tag{2}$$

This mechanism is essential as it forces the weights of weak components to converge toward zero. In this way, the model dynamically 'switches off' redundant distributions and successfully adapts to complex background changes, such as flickering lights or rustling leaves. The update of the weights over time is governed by the formula:

$$\pi_{k(t)} = \pi_{k(t-1)} + \alpha\left(M_{k(t)} - \pi_{k(t-1)}\right) - \alpha c_T \tag{3}$$

Here α is the learning rate, $M_{k(t)}$ is an indicator variable for pixel-to-component matching, and c_T is the penalty coefficient regulating the complexity of the model. If $\pi_{k(t)}$ becomes negative, the component is removed.

In the practical implementation, each video frame is first converted to grayscale to optimize computations. The generation of the binary mask (fgmask) is performed using the *step()* method, which partitions the scene into moving objects (marked in white) and a static background (in black).

Since the initial mask often contains noise from shadows or digital interference, it undergoes additional processing through the 'Opening' operation, based on the principles of [3, 8]. The process involves sequential erosion (⊖) and dilation (⊕) with a structuring element *B* (a disk with a radius of 3 pixels):

$$fgmask_{clean} - (fgmask \ominus B) \oplus B \tag{4}$$

This step ensures the removal of small artifacts while preserving the shape and integrity of larger, topologically significant objects.

The final stage of the methodology involves a detailed quantitative analysis of the detected regions, implemented via the bwconncomp and regionprops functions. These tools apply the topological analysis algorithms of [4], which allow for the extraction of key metadata for each detected object.

Within this quantitative analysis, the area (*A*) of the objects is first calculated, defined as the sum of all pixels in a given region *R* according to the formula:

$$A = \sum_{(x,y)\in R} 1 \tag{5}$$

In parallel, the Bounding Box is determined – the minimum rectangle defined by the coordinates *(x, y)* and the corresponding *width* and *height*, which tightly encloses the object.

The combination of these parameters allows the hybrid model to effectively filter out insignificant elements and noise below a certain threshold. In this way, the system provides precise coordinates and clean Regions of Interest (ROI), which are passed to the next stage for semantic classification via the YOLOv3 neural network.

The software implementation of the motion detection stage is built upon statistical modeling of the background image using the vision.ForegroundDetector system object in the MATLAB environment. This tool implements the adaptive algorithm of [1] for real-time tracking.

The process begins with the initialization and configuration of the detector using the command:

```
foregroundDetector = vision.ForegroundDetector();
```

The utilized object applies the improved model of [2, 9], whose primary advantage is the ability to automatically update the number of Gaussian components for each pixel. This adaptability allows the system to effectively handle dynamic background changes, such as flickering lights or rustling leaves.

The next phase is the systematic processing of frames and the generation of a binary mask. For each new frame from the video stream, segmentation is performed, with the image previously converted to grayscale to reduce computational complexity. The mask itself (fgmask) is created via the step() method:

```
fgmask = foregroundDetector.step(rgb2gray(frame));
```

In this process, moving objects (foreground) are marked as a logical unit (white), while the static background remains zero (black). Mathematically, this separation corresponds to calculating the probability for each pixel relative to the accumulated history of distributions.

Since the resulting raw mask often contains noise or shadow artifacts, it undergoes morphological refinement. An 'Opening' operation is applied based on the theoretical framework of [3, 10] using a disk structuring element with a radius of 3 pixels:

```
fgmask = imopen(fgmask, strel('disk', 3));
```

This step ensures that only topologically significant regions are analyzed, while preserving the shape of larger objects.

The methodology concludes with feature extraction and topological analysis based on the algorithms of [4]. By combining the *bwconncomp* and *regionprops* functions, the system generates metadata for each detected object:

```
stats = regionprops(bwconncomp(fgmask), 'BoundingBox', 'Area');
```

Here, the *Area* parameter calculates the total number of pixels in the region, allowing for the filtering of objects below a certain threshold, while *BoundingBox* defines the minimum rectangle around the object. This sequence of actions realizes the hybrid logic of the model, preparing precise Regions of Interest (ROI) for the final phase of semantic classification with YOLOv3.

3. Object Detection via YOLOv3

Once the Regions of Interest (ROI) have been defined via GMM, the system proceeds to the semantic analysis phase using the YOLOv3 (You Only Look Once, version 3) algorithm proposed by [5]. In the MATLAB environment, the model is accessible through the yolov3ObjectDetector object, which is based on the powerful Darknet-53 architecture.

Unlike classical methods that rely on heavy sliding-window algorithms, YOLOv3 treats detection as a regression problem. The image is divided into a grid of cells, and for each cell, the network simultaneously predicts *B* bounding boxes.

In the software code, the neural network is executed via the detect function, which calculates the precise coordinates of the boxes through specific transformations:

$$\begin{aligned} b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{6}$$

In these equations, b_x, b_y, b_w, b_h represent the final coordinates and dimensions of the bounding box, while t_x, t_y, t_w, t_h are the direct outputs from the Darknet-53 network. The sigmoid function σ restricts the center of the object within the cell boundaries, accounting for the offsets c_x, c_y and the dimensions of the pre-defined anchor boxes p_w, p_h.

Once the boxes are predicted, the process moves to the confidence estimation and classification phase. For each detected region, the algorithm generates confidence scores and determines its likely membership in a specific class (labels). A key parameter here is the *Objectness Score*. It reflects the probability that an object actually exists within the box and is mathematically defined through the Intersection over Union (IOU) ratio between the predicted and the ground truth box:

$$C = P(Object) \cdot IOU_{\text{pred}}^{\text{truth}} \quad (7)$$

The classification in YOLOv3 itself is characterized by the use of independent logistic classifiers for each individual class *i*, instead of the traditional *Softmax* function. This approach enables multi-label classification through the sigmoid transformation:

$$P(Class_i \,|Object) = \sigma(s_i) \quad (8)$$

where s_i represents the output signal for the respective class.

The final stage of the software implementation is the visualization and annotation of the results. Through the *insertObjectAnnotation* function, the system performs a graphical overlay of the extracted features onto the original video frame. Mathematically, this process can be represented as a function of the base image *I* and the set of detected data *D* (including boxes, scores, and labels):

$$I_{out} = f(I, \{bboxes, scores, labels\}) \quad (9)$$

This complete sequence of operations allows the hybrid model not only to detect the presence of motion in the scene but also to identify with high accuracy the type of passing objects in real-time, transforming the raw video stream into analytical information.

Once the Regions of Interest (ROI) have been successfully defined through GMM segmentation, the system proceeds to the semantic classification stage. At the core of this process lies the YOLOv3 (You Only Look Once) algorithm, proposed by [5].

The software implementation in the MATLAB environment begins with the initialization of the detector using the yolov3ObjectDetector object. It utilizes the Darknet-53 deep neural network, pre-trained on the large-scale COCO dataset:

```
detector = yolov3ObjectDetector('darknet53-coco');
```

The choice of this architecture is driven by its hybrid nature – it skillfully combines sequential convolutional layers with residual connections. This allows the model to extract complex features from the image while maintaining high computational speed, which is critical for real-time video processing.

The detection process (inference) itself treats object recognition as a regression problem. For each new frame, the network simultaneously predicts the bounding box coordinates, confidence scores, and probable classes. In the software code, this is performed via the detect function:

```
[bboxes, scores, labels] = detect(detector, frame);
```

The mathematical logic behind this operation transforms the raw network outputs using a sigmoid function σ for precise object center localization and an exponential function to determine its dimensions relative to pre-defined anchors. The resulting scores parameter reflects the model's confidence, calculated as the product of the probability of an object's presence and its Intersection over Union (IOU) accuracy with the ground truth box.

Unlike earlier versions, YOLOv3 implements independent logistic classifiers, allowing the system to perform multi-label classification. The probability for each specific class is calculated via the sigmoid dependence $P(Class_i) = \sigma(s_i)$.

The methodology concludes with visual annotation of the results, which merges the original image with the extracted metadata using the command:

```
processedFrame = insertObjectAnnotation(frame, 'rectangle', bboxes, labels)
```

This step practically implements the function $I_{\text{annotated}} = f(I, D)$, where the input frame I is augmented by the set of detected features D. Through this integrated software approach, the system provides a precise and intuitive output, enabling reliable object tracking in a dynamic environment.

4. Hybrid Model for Object Detection and Classification

An innovative hybrid model is implemented in this work, synergistically combining the advantages of statistical segmentation with the precision of deep learning. Instead of subjecting the entire frame to the resource-intensive YOLOv3 analysis, the system initially utilizes a Gaussian Mixture Model (GMM) to localize dynamic zones. This approach significantly optimizes computational resources, preparing the scene for focused detection. The software implementation of the model follows a 'cascade detection' logic, organized into two consecutive phases as shown in Fig. 1.

Phase 1: Localization of Regions of Interest (ROI). In the first step, the system generates a foreground mask from which the geometric characteristics of potential objects are extracted. This operation is based on the topological analysis of [4] and is implemented via the following code:

```
foregroundMask = step(foregroundDetector, frame); stats =
regionprops(foregroundMask, 'BoundingBox', 'Area');
```

Using the *regionprops* function, the system identifies the coordinates of all moving segments (blobs) that exceed a predefined area threshold. In this way, noise is filtered out, and only objects significant for tracking are isolated.

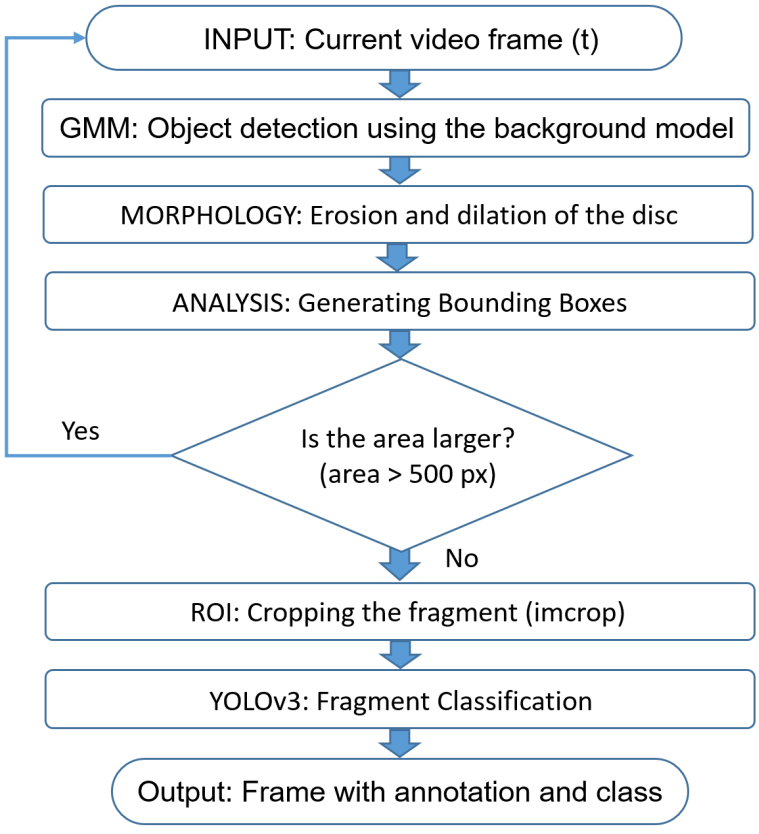

Fig. 1. Hybrid model

Phase 2: Focused Detection and Classification. In the second phase, the algorithm iterates through each detected region. The *imcrop* function is used to crop the specific segment from the original frame, which is then fed as input to the YOLOv3 detector [5]:

```
for i = 1:length(stats)
    roi = imcrop(frame, stats(i).BoundingBox);
    [bboxes, scores, labels] = detect(detector, roi);
end
```

This targeted approach allows the neural network to analyze only the visually significant parts of the image, rather than processing redundant information from the static background.

The integration of these two technologies solves several critical problems in computer vision. First, high computational efficiency is achieved as the detect operation is restricted to the regions of interest. Second, the method provides

increased accuracy because the GMM eliminates static objects that might generate false positives during standard detection. Finally, thanks to the improvement by [2], the model demonstrates exceptional real-time adaptability, adjusting dynamically to lighting changes. The final result is an intelligent system that combines classical signal processing methods with modern AI architectures to identify objects such as cars or pedestrians with a high degree of confidence.

5. Results

To evaluate the proposed methodology, three comparative experiments were conducted using the video file GOPR15601.mp4. The objective is to analyze the detection efficiency and computational speed across three different configurations: a pure statistical model (GMM), a standard neural network (YOLOv3), and the proposed hybrid model. The experiments were performed with fixed parameters: *minObjectArea = 500* and a disk(3) structuring element.

Analysis of Experiment 1: Statistical Detection via GMM The first experimental stage is focused on isolating motion within the scene using the adaptive model of [2]. The result of this process is a precise binary mask, where dynamic objects are represented as distinct white segments (blobs) on a black background as shown in Fig. 2.

Fig. 2. Binary foreground mask obtained via GMM segmentation.

The application of morphological refinement according to the method of [3] proves to be critical, as it successfully eliminates small artifacts and environmental noise, ensuring clean contours of the moving vehicles.

The main findings from this experiment highlight the dual nature of the approach. On one hand, the algorithm demonstrates exceptionally high computational speed, making it suitable for real-time systems. On the other hand, however, a significant deficit of semantic information is observed – the system registers changes in pixel intensity and localizes motion but remains incapable of identifying and classifying the detected objects into specific categories, such as “car” or “truck”. It is precisely this limitation that justifies the need for integration with more complex analysis methods, such as those offered by deep learning.

Analysis of Experiment 2: Direct Detection with YOLOv3. The second experimental stage investigates the application of the Darknet-53 architecture in classical mode, where the neural network analyzes the entire frame without prior segmentation. The visual analysis of the results confirms the high precision of the model – the system successfully generates accurate bounding boxes (*bboxes*) and correct semantic labels (labels) for each object. This is due to the implemented logistic classifiers, which allow YOLOv3 to identify objects with a high degree of confidence. Despite the high accuracy, however, the findings from this experiment highlight a significant technological trade-off. Since the neural network is forced to process the full resolution of every frame, the computational load increases progressively. The processing time increases significantly, which becomes a critical bottleneck for real-time operation. This experiment demonstrates that while deep learning provides superior classification, its direct application to the entire video stream is inefficient in terms of software resources.

Fig. 3. Object detection via YOLOv3 on an original video frame.

As shown in Figure 3, the hybrid model demonstrates high classification precision. The confidence score of 95.31% confirms that the preliminary

segmentation of the region via GMM not only does not hinder YOLOv3 but allows it to focus on the object, eliminating background noise.

Analysis of Experiment 3: Hybrid Model (GMM + YOLOv3). The third experiment investigates the efficiency of the proposed cascade structure, which merges the advantages of both technologies. The Gaussian Mixture Model (GMM) serves as a dynamic filter, localizing the Regions of Interest (ROI), while the YOLOv3 neural network is activated solely to analyze these specifically isolated segments.

Experimental data extracted during the processing of a test sequence of 127 frames recorded a total execution time of 22.82 seconds, with three distinct performance phases observed during the process. Initially, a characteristic startup peak occurs at the first frame with a time of 4.6223 seconds, which is a direct result of the technological loading of the complex YOLOv3 detector architecture and the initialization of the video stream.

Immediately following this, the system enters an adaptive phase between the second and tenth frames, where processing time progressively drops from 1.5343 seconds to 0.9210 seconds as the [2] model builds the statistical history of the background and adjusts to the dynamics of the scene. The process concludes by entering a stable regime, where a steady speed between 0.1388 seconds and 0.2066 seconds per frame is maintained in the final frames.

Fig. 4. Result of the hybrid model (GMM + YOLOv3) operation.

This stabilization corresponds to an average performance of approximately 6 FPS, which represents an optimal balance for a hybrid system based on an architecture such as Darknet-53. Ultimately, the results categorically confirm that by focusing

the neural network only on zones with registered motion, the hybrid approach successfully combines high semantic accuracy with computational efficiency, making it fully applicable for practical real-time purposes.

The experimental data categorically confirm the hypothesis established in the initial research setup, proving the advantages of the proposed approach in several key aspects. Firstly, a significant optimization of the process is observed, as the hybrid model outperforms the standalone use of YOLOv3 in terms of efficiency. This is achieved by restricting the computationally intensive detect operation solely to small Regions of Interest (ROI), precisely cropped using the *imcrop* function.

In parallel with computational speed, the system demonstrates a high degree of reliability. The integration of classical topological analysis algorithms by [4] allows for the precise filtering of objects with an area below 500 pixels, which effectively minimizes false positive results and clears the scene of noise.

Table 1. Presents a comparison of the three approaches based on the conducted tests:

Indicator	Experiment 1: GMM	Experiment 2: YOLOv3	Experiment 3: Hybrid Model
Semantic information	Missing	Complete	Complete
Computational load	Very Low	High	Balanced
Accuracy in noise	Medium	High	Very High
Average time (stable)	~0.02 s	~0.50 s	~0.17 s

The practical applicability of the development is proven by the achieved stable speed of 0.17 sec/frame at the end of the experiment. This result indicates that the system is fully suitable for the needs of automated traffic monitoring, where the balance between reaction speed and semantic classification accuracy is of critical importance.

In conclusion, the hybrid model successfully combines the statistical robustness of GMM with the deep learning capabilities of YOLOv3, providing a reliable and fast system for video analysis.

6. Conclusion

Within the framework of this study, a hybrid system for real-time vehicle detection and classification was designed, implemented, and experimentally tested, successfully achieving a balance between computational efficiency and recognition precision. The integration of classical statistical methods with modern deep learning architectures proved that the hybrid approach is highly effective for solving computer vision tasks.

The main findings of the research indicate that using the Gaussian Mixture Model (GMM) for the preliminary localization of regions of interest significantly reduces the computational load on the YOLOv3 neural network. Instead of processing the full frame, the classifier focuses solely on semantically significant segments, allowing for a reduction in processing time to an average of 0.17 seconds per frame. The high adaptability and robustness of the system are due to the successful application of [2] improved model and [3] morphological filters, which enable dynamic noise filtering and an adequate response to environmental changes. In this regard, the 500-pixel threshold for minimum area proved to be an optimal limit for eliminating artifacts without losing information about the actual objects in the scene.

Quantitative evaluation of the process shows that after the initial phase of initialization and a short adaptive period of about ten frames, the system enters a stable mode with a performance of approximately 6 FPS. This represents a tangible improvement compared to the standalone application of YOLOv3 on high-resolution video. The achieved confidence levels in classification, exceeding 95%, combined with the steady speed, make the model exceptionally suitable for deployment in intelligent transportation systems for traffic monitoring, where reliability is critical under limited hardware resources.

In conclusion, the proposed methodology demonstrates that combining the mathematical apparatus of topological analysis and statistical segmentation with the power of convolutional neural networks is an effective path for the development of next-generation video analysis systems. The results achieved lay a solid foundation for future improvements, including the implementation of even lighter neural structures or the use of hardware acceleration to achieve higher frame rates.

Acknowledgement

This work is supported by the Bulgarian National Science Fund, Project title "Innovative Methods and Algorithms for Detection and Recognition of Moving Objects by Integration of Heterogeneous Data", KP-06-N 72/4/05.12.2023.

References

1. Stauffer, C., Grimson, W. E. L.: Adaptive background mixture models for real-time tracking. In Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99), vol. 2, pp. 246-252 (1999).
2. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), vol. 2, pp. 28–31 (2004).

3. Serra, J. Image analysis and mathematical morphology London: Academic Press. Vol. 1, (1982).
4. Rosenfeld, A., Pfaltz, J. L.: Sequential operations in digital picture processing. Journal of the ACM (JACM), vol. 13(4), pp. 471-494 (1966). https://doi.org/10.1145/321356.321362.
5. Redmon, J. Farhadi, A.: YOLOv3: An incremental improvement, arXiv preprint, arXiv:1804.02767 (2018). https://arxiv.org/abs/1804.02767.
6. Garvanova, M., Ivanov, V.: Quality assessment of defocused image recovery algorithms. In: 3rd International Conference on Sensors, Signal and Image Processing (SSIP 2020), Prague, Czech Republic. ACM International Conference Proceeding Series, pp. 25-30 (2020). https://doi.org/10.1145/3441233.3441242.
7. Garvanova, M., Ivanov, V.: Quality assessment of image deburring algorithms, In: IOP Conference Series: Materials Science and Engineering, vol. 1031 (1), pp. 1-5 (2021), https://doi.org/10.1088/1757-899X/1031/1/012051.
8. Shishkov, B., Garvanova, G.: A Review of Pilotless Vehicles. In: Shishkov, B., Lazarov, A. (eds) Telecommunications and Remote Sensing. ICTRS 2023. Communications in Computer and Information Science, vol. 1990, pp. 136–143 (2023), https://doi.org/10.1007/978-3-031-49263-1_11.
9. Garvanov, I., Garvanova, M., Borissova, D., Garvanova, G.: A model of a multi-sensor system for detection and tracking of vehicles and drones. Lecture Notes in Business Information Processing, vol. 483, pp. 299-307 (2023), https://doi.org/10.1007/978-3-031-36757-1_21.
10. Tsonkov, G., Garvanova, G., Garvanov, I., Garvanova, M.: Software Architecture for Object Detection in Images Based on Color Features with Integrated Artificial Intelligence. Lecture Notes in Business Information Processing, vol. 523, pp. 270-282 (2024), https://doi.org/10.1007/978-3-031-64073-5_18.