CHAPTER 5

Detection of small and camouflaged objects

Dmytro Krytskyi
Elvira Kaidan
Artem Chekhovsky

Abstract

Detecting small and camouflaged objects in aerial images is challenging, especially when data is acquired from drones. In such images, targets often occupy only a very small portion of the frame and can visually blend in with the surrounding terrain. Image quality is also affected by flight altitude, unstable lighting, motion blur, background noise, and intentional camouflage. Therefore, analyzing such data is complex and time-consuming, significantly increasing the likelihood of missing a target object. Therefore, automatic detection is a highly relevant approach to identifying these objects.

This chapter examines the use of deep learning methods for detecting small and camouflaged objects in aerial photos and videos. The study focuses on YOLO-based detectors, as these models combine good detection quality with high processing speed and are suitable for practical applications. Particular attention is given to the comparison of YOLOv8 and YOLOv11. An experimental study was conducted on an annotated dataset created from publicly available video footage. Two model configurations were trained and evaluated. Image resizing by direct stretching was replaced by adding padding to preserve the proportions of objects within the frame. Additionally, the class structure was simplified, reducing ambiguity during training and increasing classification confidence. These changes were tested alongside the transition from YOLOv8n to YOLOv11s.

The results showed that the improved approach provided more stable detection in complex data and significantly reduced training time. The YOLOv11 model demonstrated the best practical results when working with small targets in complex background conditions. The obtained results confirm that modern architectures based on YOLO family models can be effectively applied to automated data analysis

 DOI: 10.21303/978-9908-845-05-0.ch5

and can serve as a foundation for the further development of intelligent decision support systems.

Keywords

Small object detection, camouflaged object detection, aerial imagery, computer vision, deep learning, object detection, YOLOv8, YOLOv11, image preprocessing, padding, dataset annotation, automated recognition systems.

5.1 Introduction

Unmanned aerial vehicles (UAVs) have significantly expanded the capabilities of remote sensing and the rapid collection of geospatial data. Today, they are widely used for environmental monitoring, infrastructure inspection, and military reconnaissance. Their use allows for the acquisition of detailed visual information over large areas in a relatively short time. At the same time, the volume of photo and video data collected by UAV platforms has increased so much that manual analysis is often insufficient, especially in situations requiring rapid decision-making [1, 2].

Detecting small and intentionally camouflaged objects is particularly challenging [3, 4]. Such targets may occupy only a small number of pixels in an image and are difficult to distinguish from the surrounding background. The situation is further complicated by camouflage, variable lighting, weather conditions, and the visual complexity of the terrain itself [3, 5]. As a result, the operator must process large volumes of visually complex data, which increases the likelihood of missing desired objects. In such cases, approaches designed specifically for small-object analysis, including hyper-inference and image slicing, become particularly relevant [4].

These limitations make automated image analysis an important research and practical challenge. In recent years, deep learning methods have demonstrated good results in object detection tasks, especially in scenes where traditional image processing methods are not robust enough. Convolutional neural networks (CNNs) are particularly effective because they can learn informative visual features directly from data rather than relying on hand-crafted descriptors [6]. For the use of such systems onboard UAVs, it is important that such models be able to run in real-time or near-real-time on onboard systems with limited computing resources [7].

Among modern object detection approaches, the YOLO family remains one of the most widely used solutions. Its single-stage detection scheme allows the model to predict object classes in a single, straightforward pass, providing a good balance between speed and accuracy [8]. Because of this, YOLO-based detectors are widely used in video analytics and other applications requiring fast processing [9].

At the same time, model quality depends not only on the detector architecture itself. The final result is also influenced by the composition of the training dataset, the data augmentation strategy, preprocessing methods, and the choice of training parameters. Even relatively small changes in preprocessing can impact how well the system handles noisy data and how reliably it detects objects close to the sensor's resolution limit [10].

The aim of this study is to explore approaches to automatically detecting small and partially camouflaged objects in photos and videos using modern deep learning methods. Particular attention is paid to the performance of YOLO-based models under challenging observation conditions.

5.2 Problem of detecting small and camouflaged objects in aerial imagery

Detecting small and camouflaged targets in aerial imagery data streams is one of the most critical and complex fundamental tasks in computer vision [11]. The specific nature of obtaining visual information from unmanned aerial vehicles (UAVs) gives rise to a number of disruptive factors that significantly reduce the effectiveness of classical segmentation and identification algorithms.

The main challenge in such situations is the gap between the sensor's resolution and the small size of the target object. When capturing a large area, the UAV is flying at a high height, and the objects occupy a very small number of pixels in the frame [12]. As a result, we get a "deficiency": texture, shadows – all the most important criteria for recognition are lost, and the system is unable to distinguish the object from the background.

Camouflage must also be considered. During military or reconnaissance missions, objects are often concealed. During such operations, camouflage nets or patterns are used, or the specific terrain and natural factors are taken into account. In such situations, the spectral characteristics and the surrounding surface become similar. The texture of the target to be identified and the surroundings become so similar that the visual boundary becomes blurred. In such cases, classical gradient detection methods fail to provide reliable results for the model.

An additional challenge is posed by the structural non-uniformity of the background, namely its high entropy. Aerial imagery typically contains a high number of non-textural elements, such as buildings, roads, and so on. These objects create visual noise, leading to the detection of false contours and textures. For a system operating in real time and required to make rapid decisions, such errors are unacceptable.

What also cannot be ignored is that the mission is carried out under dynamic conditions [13]. During operation, the UAV performs a series of maneuvers, resulting in sudden changes in camera angles, lighting fluctuations, and additional atmospheric effects. All of these factors lead to blurring and geometric distortions in the video.

Conventional computer vision methods, which rely on manually created features, prove to be insufficiently robust against the difficulties mentioned earlier. In comparison, deep learning allows for the automatic construction of a feature hierarchy. It gradually adapts to complex spatial structures and nonlinear dependencies in the data.

Given all the aforementioned problems, the task is to create adaptive neural network approaches based on deep learning, tailored to the specifics of aerial images from UAVs. The system must ensure high accuracy with a low false positive rate and robustness to dynamic imaging conditions.

5.3 Deep learning approaches for small object detection

Serious changes have taken place in the field of computer vision over the past ten years. Instead of deterministic algorithms, in which features were specified manually, deep learning methods have increasingly come into use.

The fundamental mechanism for feature extraction in such systems is the convolution operation, which preserves spatial correlations between pixels.

The convolution operation can be formally represented as follows

$$(I*K)(i,j)=\sum_m\sum_n I(i+m,j+n)K(m,n), \tag{5.1}$$

where I – the input image; K – the convolution kernel; i, j – the pixel coordinates in the output feature map.

As data passes sequentially through a cascade of convolutional layers, low-level features (lines, gradients) are transformed into high-level semantic structures.

Activation functions are used to model nonlinear dependencies and improve the network's approximation capability. The standard for modern architectures is the Rectified Linear Unit (ReLU) function, defined as

$$f(x)=\max(0,x). \tag{5.2}$$

The use of ReLU and its variants (such as SiLU in the latest versions of YOLO) effectively addresses the problem of gradient vanishing and accelerates model convergence during training.

In modern object detector architecture design, two conceptual strategies are distinguished [14]:

1. Two-stage detectors: R-CNN family. The detection process is divided into the generation of regional proposals and their subsequent classification. Despite their high accuracy, such systems are computationally intensive, which limits their application in real-time tasks [15].

2. One-stage detectors: YOLO, SSD, RetinaNet. These models treat detection as a single regression problem, predicting the coordinates of the bounding boxes and class probabilities in a single pass through the network. This approach is the dominant one in aerial reconnaissance systems due to its high throughput and the ability to deploy it on UAVs with limited computational resources.

However, the detection of small-scale targets remains a "bottleneck" almost exclusively for deep neural networks. The main reason lies in the loss of spatial information that occurs during the sequential downsampling of feature maps. To mitigate this phenomenon, modern architectures employ multi-scale prediction mechanisms and hierarchical feature combination structures (Feature Pyramid Networks, FPN). This allows the network to aggregate contextual information from deep layers with the high spatial resolution of shallow layers.

The evolution of the YOLO family of models has led to the emergence of architectures that demonstrate high robustness to challenging observation conditions: low contrast, spectral blending of the object with the background, and geometric distortions. Within the scope of this study, special attention is paid to a comparative analysis of the YOLOv8 and YOLOv11 models. The choice of these iterations is due to their technological sophistication: the integration of anchor-free detection methods, improved feature extraction blocks, and optimized loss functions, which together open up new prospects for the automated interpretation of aerial reconnaissance data under conditions of active countermeasures and camouflage [16].

5.4 YOLO-based architectures for aerial object detection

Models from the YOLO family are among the most widely used object detection algorithms in modern computer vision systems. Their popularity stems from their combination of high image processing speed and reasonably high object recognition accuracy. Unlike two-stage detectors, such as Faster R-CNN, which first generate region proposals and then classify them, YOLO models

perform object localization and classification in a single pass through the neural network.

The main idea behind the YOLO algorithm is to divide the input image into a regular grid. Each cell of this grid is responsible for predicting objects, which centers are within the corresponding area. For each cell, the neural network predicts the coordinates of the bounding box, the probability of an object's presence, and the probability that the object belongs to a specific class.

The number of parameters predicted by the model can be described as follows

$$S \times S \times (B \cdot 5 + C), \tag{5.3}$$

where S – the size of the grid into which the image is divided; B – the number of predicted frames for each cell; C – the number of object classes.

Each bounding box is described by five parameters: center coordinates x and y, width w, height h, and a confidence value, which characterizes the probability of an object being present within the bounding box.

The architecture of modern YOLO models consists of three main components: backbone, neck, and head. The backbone is used to extract features from the input image using convolutional layers. In this block, feature maps of various levels of abstraction are formed. The neck is designed to combine features at different scales, allowing for more effective detection of both large and small objects. Structures such as Feature Pyramid Networks (FPN) or Path Aggregation Networks (PAN) are often used for this purpose. The head performs the actual prediction of bounding box coordinates and object classes.

To evaluate the quality of object localization, the Intersection over Union (IoU) metric is used, which determines the degree of overlap between the predicted bounding box and the actual bounding box of the object

$$IoU = \frac{A_{rea}\left(B_{pred} \cup B_{gt}\right)}{A_{rea}\left(B_{pred} \cap B_{gt}\right)}, \tag{5.4}$$

where B_{pred} – specified frame; B_{gt} – ground truth frame.

The basic building block of the architecture is the convolutional layer (conv), which structure is shown in **Fig. 5.1**. It consists of a 2D convolutional layer (Conv2d), a batch normalization layer (BatchNorm2d), and a SiLU activation function. This combination allows for the effective extraction of spatial features from the image and stabilizes the neural network training process.

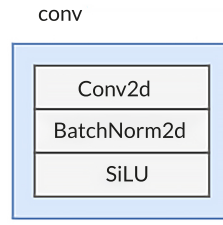

Fig. 5.1 Structure of the convolution block (Conv)
Source: created by the authors

To improve the efficiency of feature processing in the network, bottleneck modules (b_elem) are used. They consist of a sequence of convolutional layers and ensure efficient information transfer between the layers of the neural network. The structure of this block is shown in **Fig. 5.2**.

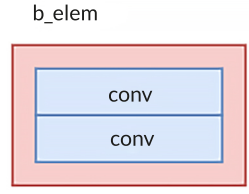

Fig. 5.2 Structure of the bottleneck block (b_elem)
Source: created by the authors

One of the key modules in modern YOLO models is the C2f block, which combines several bottleneck elements with additional convolutional layers. This structure improves feature transfer between layers and enhances the training efficiency of deep neural networks. The architecture of the C2f module is shown in **Fig. 5.3**.

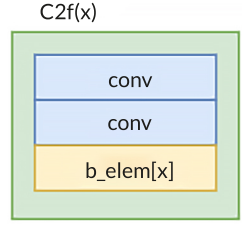

Fig. 5.3 Architecture of the C2f module
Source: created by the authors

The final component of the architecture is the detection head, which predicts the coordinates of bounding boxes and object classes.

This block uses information from the previous network layers to generate the final detection results.

The structure of this component is shown in **Fig. 5.4**.

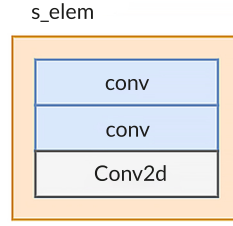

Fig. 5.4 Structure of the detection head element
Source: created by the authors

The combination of these structural elements forms the complete architecture of the YOLO model, which includes feature extraction blocks and an object detection block. The overall structure of the model's architecture is shown in **Fig. 5.5**.

In this study, two state-of-the-art architectures – YOLOv8 and YOLOv11 – were used to detect small and occluded objects in aerial reconnaissance images. The YOLOv8 model is one of the most widely used architectures for object detection tasks and demonstrates a good balance between accuracy and processing speed. However, further development of the YOLO family of models led to the emergence of the newer YOLOv11 architecture, which is characterized by improved computational efficiency and increased detection accuracy [8].

In addition to improvements in the neural network architecture, detection performance depends significantly on image preprocessing methods and the structure of the training dataset.

This study also investigated the impact of various data preparation approaches, including the use of padding to preserve image proportions and the optimization of the number of classes in the dataset.

The analysis conducted allows to evaluate the effectiveness of modern YOLO architectures in automated aerial reconnaissance image analysis tasks and identify the most promising approaches for detecting small-sized and camouflaged objects.

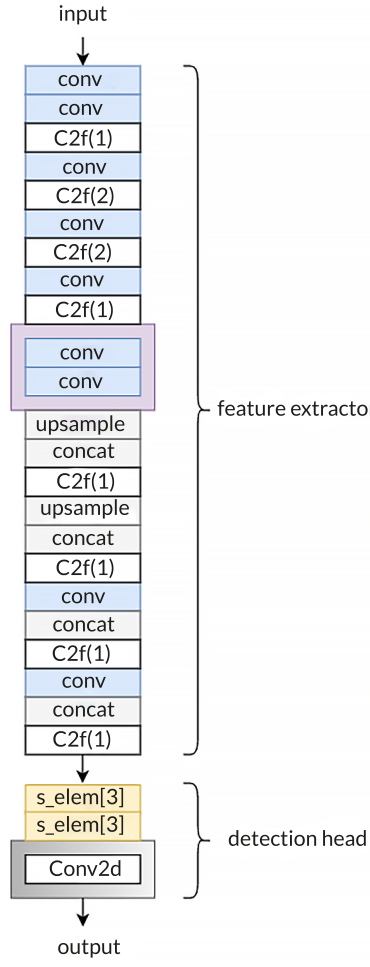

Fig. 5.5 Overall architecture of the YOLO model
Source: created by the authors

5.5 Experiment using YOLOv8 for small object detection

To investigate the effectiveness of modern methods for detecting small and occluded objects, an experiment was conducted using the YOLOv8 model. This model

belongs to the modern generation of single-stage object detectors and is widely used in computer vision tasks due to its combination of high image processing speed and sufficiently high detection accuracy [13].

In the first stage, a training dataset was created based on aerial reconnaissance video footage. The video recordings were divided into individual frames, which were used as images for further training of the neural network. For each image, object annotation was performed, during which the coordinates of the bounding boxes and the corresponding object classes were determined.

Analysis of the dataset structure is a crucial step in experiment preparation, as class distribution can affect the quality of model training. The distribution of objects by class is shown in **Fig. 5.6**.

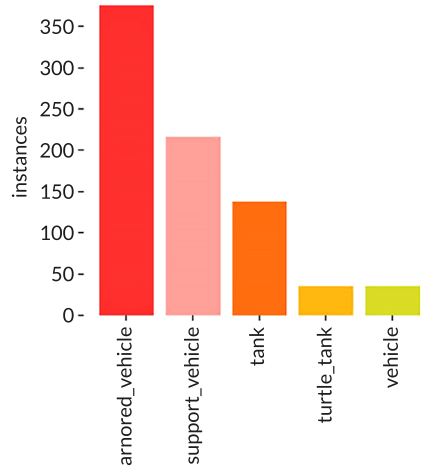

Fig. 5.6 Histogram of class distribution in the training dataset
Source: created by the authors

Further analysis of the positions of objects in the images allows to assess the spatial distribution of objects in the dataset. **Fig. 5.7** shows the distribution of the positions of the frame centers and their sizes.

After preparing the dataset, the YOLOv8 model was trained. During training, the neural network gradually optimizes its parameters by minimizing the loss function and improving object detection accuracy.

The trends in the model's key quality metrics over the training epochs are shown in **Fig. 5.8**.

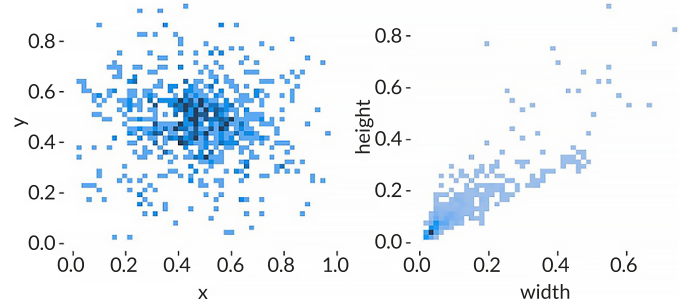

Fig. 5.7 Distribution of positions (left) and dimensions (right) of the limiting frames
Source: created by the authors

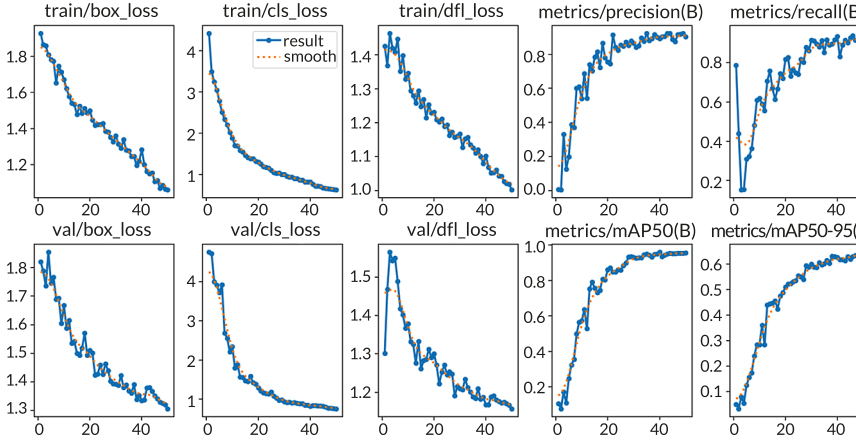

Fig. 5.8 Changes in the model's quality metrics over training epochs
Source: created by the authors

Standard object detection metrics are used to evaluate the model's performance. One of the key metrics is Precision, which measures the proportion of correctly detected objects among all detections

$$Precision = \frac{TP}{TP + FP}, \tag{5.5}$$

where *TP* (True Positive) refers to correctly detected objects, and *FP* (False Positive) refers to false detections.

Another important metric is Recall, which shows what proportion of actual objects the model was able to detect

$$Recall = \frac{TP}{TP + FN}, \tag{5.6}$$

where *FN* (False Negative) refers to objects that the model failed to detect.

To comprehensively evaluate detection quality, the *F*1-score is used, which combines the Precision and Recall metrics

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{5.7}$$

Using this metric provides a comprehensive assessment of the model's performance, taking into account both detection accuracy and recall.

For a more detailed analysis of the model's performance, curves were plotted showing the dependence of key detection metrics on the model's confidence level (**Fig. 5.9**).

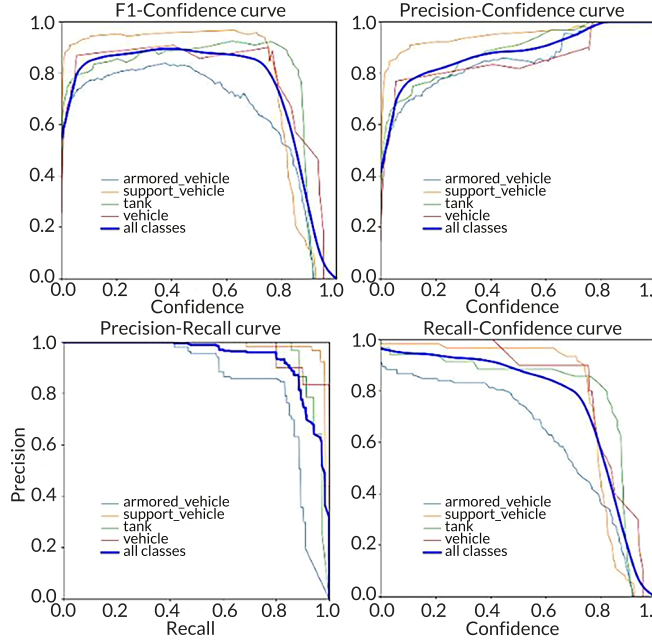

Fig. 5.9 Dependency curves for key detection metrics in the YOLOv8 model
Source: created by the authors

These graphs allow to assess the impact of the confidence threshold on detection accuracy and completeness, as well as to determine the optimal threshold value for the practical application of the model.

As can be seen from the graphs, as the confidence threshold increases, detection accuracy (Precision) improves, but at the same time, the completeness of object detection (Recall) decreases. The Precision-Recall curve illustrates the relationship between these two metrics. The maximum *F*1-score corresponds to the optimal balance between precision and recall. A confusion matrix is used to evaluate the classification quality of different object types. It allows to determine which object classes are most frequently misclassified by the model. The normalized confusion matrix is shown in **Fig. 5.10**.

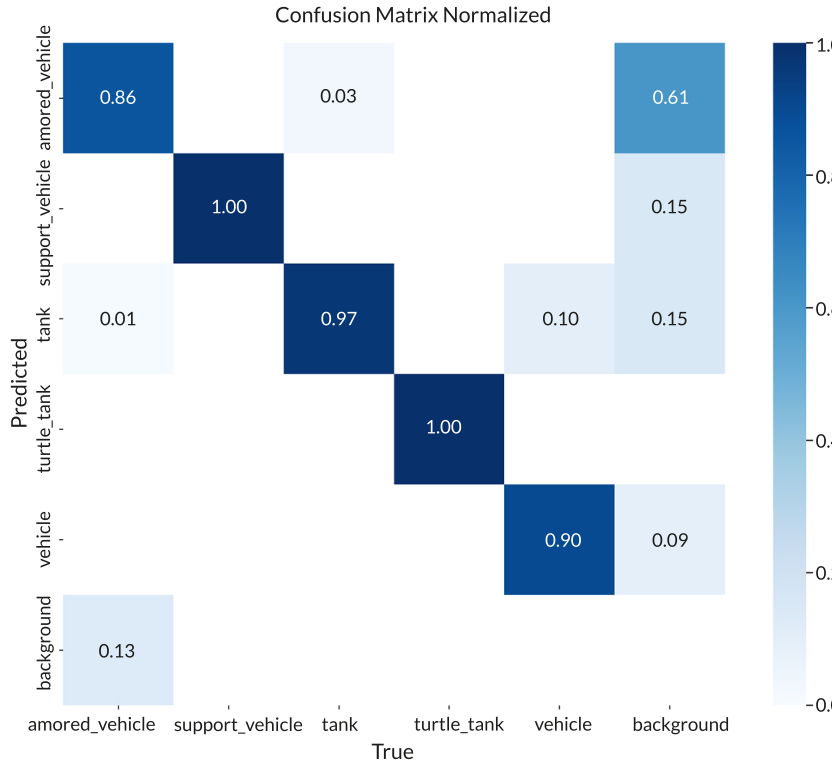

Fig. 5.10 Normalized confusion matrix for the YOLOv8 model
Source: created by the authors

The results showed that the YOLOv8 model is capable of effectively detecting objects in aerial reconnaissance images. At the same time, a number of limitations were identified regarding the detection of small and partially occluded objects, which is attributed to their low pixel count in the image and complex background conditions.

The experimental results were used as a baseline for further comparison with the newer YOLOv11 architecture, which incorporated additional optimizations in image preprocessing and dataset structure.

5.5.1 Experiment using YOLOv11 for small object detection

Following the baseline experiment using the YOLOv8 model, a follow-up study was conducted using the newer YOLOv11 architecture. The primary objective of this experiment was to improve the detection accuracy of small and occluded objects, as well as to optimize the model training process.

The new experiment used the same dataset as the previous study, but several changes were made to the data preparation and model configuration. Specifically, instead of resizing images to a fixed size, the padding method was used, which involves adding blank areas to the image to achieve the required size without altering the aspect ratio. When using the padding method, the image is scaled to a specified size without changing the aspect ratio, after which empty areas are added to the image. Let the original image have a size of $W \times H$, and the required size of the neural network's input image be $S \times S$.

The scaling factor is defined as

$$r = \min\left(\frac{S}{W}, \frac{S}{H}\right). \tag{5.8}$$

After resizing, the new image dimensions are defined as

$$W' = rW, H' = rH. \tag{5.9}$$

The size of the padding area is calculated as follows:

$$p_x = \frac{S - W'}{2}, p_y = \frac{S - H'}{2}, \tag{5.10}$$

where p_x – horizontal padding; p_y – vertical padding.

In addition, the class structure in the dataset was optimized. Reducing the number of classes improves classification confidence, as the neural network focuses on a smaller number of object categories. This also reduces image processing time and increases the speed of the detection system.

During model training, a significant reduction in training time was observed. While in the previous experiment using YOLOv8, model training took more than two hours, in the case of YOLOv11, the full training cycle was completed in approximately 30 minutes.

This indicates the increased efficiency of the new architecture and the optimization of the neural network training process.

The dynamics of changes in the model's key quality metrics over training epochs are shown in **Fig. 5.11**.

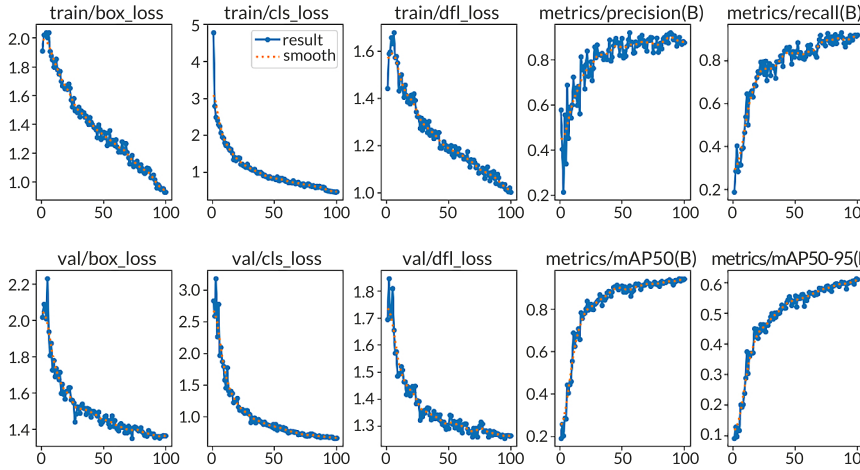

Fig. 5.11 Changes in the key performance metrics of the YOLOv11 model during training
Source: created by the authors

To analyze the model's performance in greater detail, it is possible to plot curves showing how key detection metrics vary with the confidence threshold. These metrics include Precision, Recall, and *F*1-score, which are widely used to evaluate the quality of object detection systems.

The Precision-Confidence, Recall-Confidence, Precision-Recall, and *F*1-Confidence graphs allow to evaluate the model's behavior as the confidence threshold

changes and to determine the optimal balance between detection accuracy and completeness (**Fig. 5.12**).

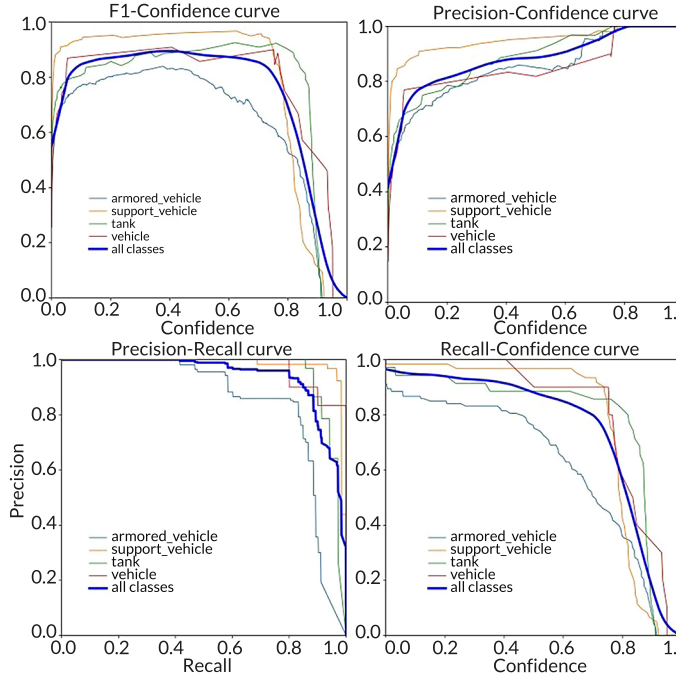

Fig. 5.12 Performance curves for the YOLOv11 model
Source: created by the authors

An analysis of the graphs shows that the model exhibits stable metric values across a wide range of confidence thresholds. As the threshold increases, the Precision metric rises, indicating a decrease in the number of false detections. At the same time, the Recall metric gradually decreases, as some objects may be missed when the threshold is set too high.

For a more detailed analysis of the classification results, a confusion matrix was constructed, which allows for the evaluation of the accuracy of identifying individual object classes. The normalized confusion matrix is shown in **Fig. 5.13**.

As shown in the confusion matrix, most objects are classified correctly, indicating the model's high accuracy. The small number of classification errors is due to the similarity of certain object types and challenging shooting conditions.

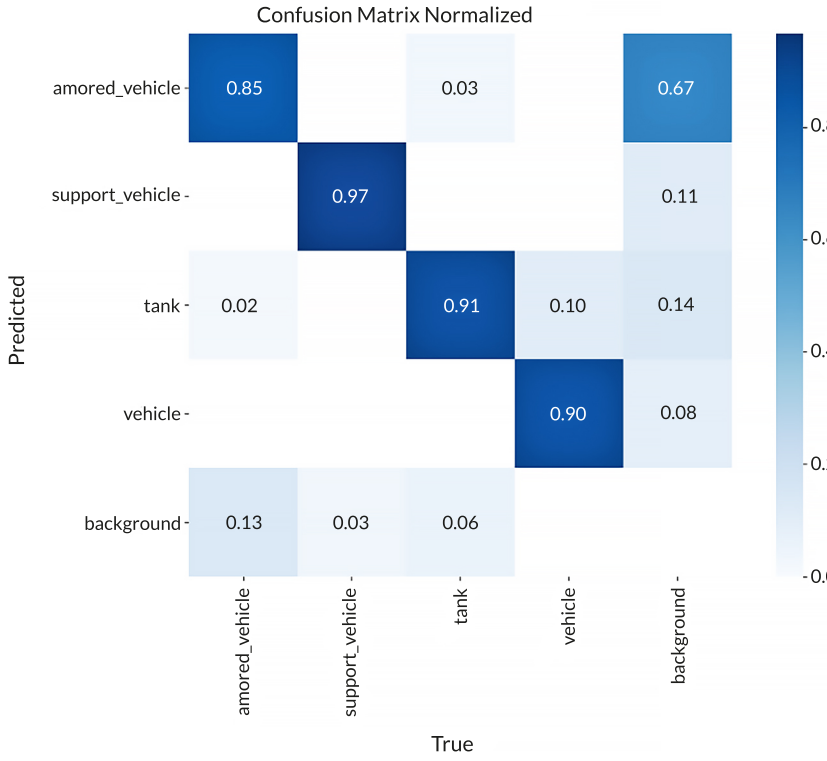

Fig. 5.13 Normalized confusion matrix for the YOLOv11 model
Source: created by the authors

The results demonstrate that using the YOLOv11 model in combination with optimized data preprocessing improves the detection performance of small objects and reduces model training time. This makes this approach promising for use in automated analysis systems for aerial reconnaissance images.

5.6 Comparative analysis of YOLOv8 and YOLOv11

To evaluate the effectiveness of the proposed approach, a comparative analysis was conducted between the YOLOv8n and YOLOv11s models, which were used to detect small and partially hidden objects in photos and videos.

In the second experiment, several changes were made to the data preprocessing and model tuning. First, the image preprocessing method was modified. While in the first experiment, images were resized to the required dimensions, in the second experiment, padding was used, i.e., adding empty areas to the image to achieve the required size without changing its proportions. This approach preserves the geometric characteristics of objects and avoids their deformation, which positively impacts detection quality.

Second, the study utilized the newer YOLOv11s neural network architecture. Version 11's model utilizes computational resources more efficiently, as evidenced by network training speed and data processing speed.

Furthermore, the dataset structure was optimized by reducing the number of classes. This significantly increased the probabilities of class detection, particularly for the "tank" class. The YOLOv8n model took over two hours to train, while the YOLOv11s model was trained in approximately 30 minutes, indicating significant optimization of the training process.

A comparison of the key characteristics of the models is presented in **Table 5.1**.

Table 5.1 Comparison of model specifications

Parameter	YOLOv8n	YOLOv11s
Network architecture	YOLOv8n	YOLOv11s
Image preprocessing	Image resizing	Padding with preserved aspect ratio
Dataset size	> 750 images	> 750 images
Number of classes	initial dataset	optimized dataset
Training time	> 2 hours	< 30 minutes
Precision	≈ 0.90	≈ 0.91–0.93
Recall	≈ 0.88–0.90	≈ 0.90–0.92
mAP@0.5	≈ 0.95	0.939
Detection confidence	moderate	higher

Source: created by the authors

For a comparative analysis, **Fig. 5.14** shows the results of object detection obtained using the YOLOv8n and YOLOv11s models. The examples provided illustrate how the models perform when detecting small objects in aerial reconnaissance images.

Fig. 5.14 Comparison of detection results: 1 – YOLOv8n; 2 – YOLOv11s
Source: created by the authors

5.7 Conclusions

The problem of detecting small and camouflaged objects is highly relevant. This chapter demonstrates the potential of using YOLO family models for this task, as they offer a combination of high speed and sufficient accuracy.

This study analyzed the architectural features of YOLO models and demonstrated their structural elements. An experimental performance evaluation of the YOLOv8n and YOLOv11s models was also conducted. For training and testing, a dataset was created based on 25 videos totaling over 1 hour, from which over 750 images were extracted and annotated.

A baseline experiment using YOLOv8n confirmed the feasibility of effective object detection in photo and video data. However, the results demonstrated the need for further refinement of the approach, particularly with regard to image preprocessing, class set structure, and model architecture.

In the second experiment, the YOLOv11s model was applied, and the data preparation procedure was improved. Instead of directly resizing images, padding was used, preserving the proportions of objects and reducing geometric distortions. Furthermore, the class structure was optimized, which positively impacted classification confidence and detection speed.

The results showed that the YOLOv11s model provides high detection performance: AP = 0.848 for the armored_vehicle class, 0.983 for support_vehicle, 0.956 for tank, and 0.968 for vehicle, while the overall AmAP@0.5 is 0.939. Model training time was reduced from over two hours for YOLOv8n to approximately 30 minutes for YOLOv11s, indicating a significant increase in computational efficiency.

Thus, the results of this study confirm the promise of using modern YOLO family architectures for automated data analysis. The combination of improved data preprocessing methods, class structure optimization, and the use of new detection models allows for increased reliability in detecting small and camouflaged objects in challenging surveillance environments. The main limitations and assumptions of the proposed approach are summarized in **Table 5.2**.

These limitations and assumptions define the directions for future research and indicate the need for validation on larger and more diverse datasets, as well as testing under real-world operating conditions.

Prospects for further research include expanding the training dataset and further adapting the models to practical conditions in monitoring and reconnaissance systems. It is also proposed to increase data augmentation at subsequent stages of research.

Table 5.2 Main limitations and assumptions of the study

Category	Description
Limitation	The dataset is relatively limited in size and diversity, as it is based on publicly available video data
Limitation	The model was not evaluated under extreme conditions such as severe illumination changes, weather effects, or sensor noise
Limitation	Motion blur, rapid viewpoint changes, and real-time communication delays were not explicitly considered
Limitation	Limited validation under real UAV operational conditions
Assumption	Image acquisition conditions are assumed to be relatively stable
Assumption	Preserving object geometry using padding is considered more important than potential loss of contextual information
Assumption	Reducing the number of classes improves detection performance
Assumption	The selected YOLOv11 architecture is appropriate for the defined task

Source: created by the authors

Conflict of interest

The authors declare that there is no conflict of interest in relation to this paper, as well as the published research results, including the financial aspects of conducting the research, obtaining and using its results, as well as any non-financial personal relationships.

Financing

The research has been funded by National Research Foundation of Ukraine (accessed on 24 June 2025) within Project No. 2025.06/0037 "A system for detecting and recognizing camouflaged and small objects based on the use of modern computer vision technologies" (2025–2026).

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence statements

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

Authors' contributions

Dmytro Krytskyi: Approved the research concept, Supervised the study, Contributed to the research design, Participated in the interpretation of the results, and Critically revised the manuscript.

Elvira Kaidan: Prepared and annotated the dataset, Assisted in conducting the experiments, Contributed to the software development, and Preparation and formatting responsibilities of the manuscript materials.

Artem Chekhovsky: Contributed to the software implementation, Validation of the obtained results, and Manuscript editing.

References

1. Pan, Yu., Li, L., Qin, J., Chen, J.-J., Gardoni, P. (2024). Unmanned aerial vehicle-human collaboration route planning for intelligent infrastructure inspection. Computer-Aided Civil and Infrastructure Engineering, 39 (14), 2074–2104. https://doi.org/10.1111/mice.13176
2. Zeng, B., Gao, S., Xu, Y., Zhang, Z., Li, F., Wang, C. (2024). Detection of Military Targets on Ground and Sea by UAVs with Low-Altitude Oblique Perspective. Remote Sensing, 16 (7), 1288. https://doi.org/10.3390/rs16071288
3. Hwang, K.-S., Ma, J. (2024). Military camouflaged object detection with deep learning using dataset development and combination. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 23 (1), 67–78. https://doi.org/10.1177/15485129241233299
4. Akyon, F. C., Onur Altinuc, S., Temizel, A. (2022). Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. 2022 IEEE International Conference on Image Processing (ICIP), 966–970. https://doi.org/10.1109/icip46576.2022.9897990
5. Li, Y., Li, Q., Pan, J., Zhou, Y., Zhu, H., Wei, H., Liu, C. (2024). SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. Remote Sensing, 16 (16), 3057. https://doi.org/10.3390/rs16163057

6. Han, Y., Guo, J., Yang, H., Guan, R., Zhang, T. (2024). SSMA-YOLO: A Lightweight YOLO Model with Enhanced Feature Extraction and Fusion Capabilities for Drone-Aerial Ship Image Detection. Drones, 8 (4), 145. https://doi.org/10.3390/drones8040145
7. Yang, C., Shen, Y., Wang, L. (2025). EMFE-YOLO: A Lightweight Small Object Detection Model for UAVs. Sensors, 25 (16), 5200. https://doi.org/10.3390/s25165200
8. Kaidan, E. (2026). Improved object recognition methods in UAVS based on YOLO. Propylaea of Law and Security, 8, 218–221. https://doi.org/10.32620/pls.2025.8.55
9. Ali, M. L., Zhang, Z. (2024). The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. Computers, 13 (12), 336. https://doi.org/10.3390/computers13120336
10. Krytskyi, D., Tkachov, I., Pyvovar, M., Karatanov, S., Krikun, A. (2025). Detecting Objects in Photos and Videos Obtained by Aerial Reconnaissance from UAVs. Integrated Computer Technologies in Mechanical Engineering – 2024. Cham: Springer, 270–283. https://doi.org/10.1007/978-3-031-94845-9_23
11. Kritsky, D., Kaidan, E., Tkachov, I., Lukin, V. (2025). Automatic detection of hidden objects using uavs: modern neural network approaches. Herald of Khmelnytskyi National University. Technical Sciences, 359 (6.2), 193–204. https://doi.org/10.31891/2307-5732-2025-359-98
12. Mi, Q., Chao, J., Chen, A., Zhang, K., Lai, J. (2026). YOLO11s-UAV: An Advanced Algorithm for Small Object Detection in UAV Aerial Imagery. Journal of Imaging, 12 (2), 69. https://doi.org/10.3390/jimaging12020069
13. Makarichev, V., Tsekhmystro, R., Lukin, V., Krytskyi, D. (2025). Performance Improvement of Vehicle and Human Localization and Classification by YOLO Family Networks in Noisy UAV Images. Information, 16 (12), 1087. https://doi.org/10.3390/info16121087
14. Chaini, C., Jha, V. K., Rajnish, K. (2026). A comparison of single-stage and two-stage based crater detectors on the lunar surface. Earth Science Informatics, 19 (2). https://doi.org/10.1007/s12145-025-02066-7
15. Gu, H., Wu, J., Huang, H. (2026). CASA-RCNN: A Context-Enhanced and Scale-Adaptive Two-Stage Detector for Dense UAV Aerial Scenes. Drones, 10 (2), 133. https://doi.org/10.3390/drones10020133
16. Stets, S. (2025). Analysis of accuracy and speed of vehicle detection using neural networks YOLOv8 and YOLOv11. Herald of Khmelnytskyi National University. Technical Sciences, 357 (5.2), 123–130. https://doi.org/10.31891/2307-5732-2025-357-74