

Recognition accuracy analysis of military vehicles in aerospace imagery using different neural network architectures

Vita Kashtan^{1,†}, Volodymyr Hnatushenko^{*1,†}

¹ Dnipro University of Technology, Dmytra Yavornytskoho Ave 19, Dnipro, 49005, Ukraine

Abstract

Military vehicle recognition in aerospace imagery is a challenging task due to small target size, complex backgrounds, arbitrary object orientation, heterogeneous acquisition conditions, and limited annotated data. This study investigates the recognition accuracy of four YOLO-based neural network architectures: YOLOv5, YOLOv8, YOLOv9, and YOLOv11—using an author-created dataset of 2,274 aerospace images divided into 1,592 training, 341 validation, and 341 test images. The harmonized dataset contains 11,454 object annotations across four classes: military-vehicle, tank, truck, and jeep. The architectures were evaluated using the same dataset partition, an input resolution of 640×640 pixels, and a common 100-epoch training budget. YOLOv8 achieved the most favorable overall balance, with Precision of approximately 0.65, Recall of approximately 0.35, and mAP@50 of approximately 0.30, while YOLOv9 achieved a comparable mAP@50 of approximately 0.29. YOLOv5 showed the lowest Recall (approximately 0.20) and mAP@50 (approximately 0.23). Class-wise analysis revealed substantially lower recall for tank, truck, and jeep than for the dominant generic military-vehicle class. The results demonstrate that architectural differences within the YOLO family lead to different recognition characteristics under identical data-partition and training-budget conditions, while the pronounced imbalance of class annotations remains an important limitation for fine-grained vehicle recognition.

Keywords

military vehicle recognition, aerospace imagery, deep learning, YOLO, object detection

1. Introduction

The development of military technologies and the emergence of new types of threats [1] impose increased requirements on the efficiency, accuracy, and reliability of intelligence data processing. In this context, deep learning methods have become increasingly important in national security and defense systems [2, 3], enabling effective solutions for detecting and identifying potential threats [4].

A particularly important role in modern intelligence systems is played by Imagery Intelligence (IMINT) [5], which relies on data obtained from optical, infrared, radar, and other sensors mounted on ground-based, airborne, and space platforms. Today, there is a close integration of IMINT with Open-Source Intelligence [6]. A significant portion of such data is formed as high-spatial-resolution aerospace imagery obtained from satellite systems and unmanned aerial vehicles, providing detailed representation of objects and processes on the Earth's surface. However, automated recognition of military vehicles in aerospace imagery is accompanied by a number of substantial scientific and technical challenges. In particular, the complexity of the task is caused by the presence of small-sized objects against highly detailed and heterogeneous background textures; significant variability in the

* ADT-ISMDDC 2026: *Advanced Digital Technologies: Intelligent Systems, Modeling, and Data-Driven Control*, September 22–24, 2026, Odesa, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ vitalionkaa@gmail.com (V. Kashtan); vvgnat@ukr.net (V. Hnatushenko)

ORCID 0000-0002-0395-5895 (V. Kashtan); 0000-0003-3140-3788 (V. Hnatushenko)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

spatial resolution of input data; the influence of atmospheric phenomena, sensor noise, and geometric and radiometric distortions; arbitrary spatial orientation of objects; as well as the use of masking and camouflage techniques that reduce the contrast and distinctiveness of target features.

A separate scientific and applied issue of importance is the formation of training datasets for neural network models. Building representative datasets requires considerable time and computational resources due to the need for high-precision object annotation, including oriented bounding boxes that accurately reflect the geometric characteristics of arbitrarily oriented objects. An additional challenge is the limited availability of high-quality, balanced, and sufficiently diverse data in the military domain, which reduces models' ability to generalize and their robustness to variations in input data. In military-purpose tasks, object detection and recognition systems must ensure high operational efficiency under conditions of a priori uncertainty and limited information. In particular, it is necessary to ensure the identification of small objects, account for variability in image spatial resolution, and ensure invariance to atmospheric distortions. Modern approaches to object recognition, including transformer-based architectures (e.g., DETR) [7] and hybrid feature fusion methods such as Pyramid Converge-and-Assign Fusion [8], demonstrate high potential to improve object recognition accuracy in complex scenes. However, their practical implementation in defense-oriented information-analytical systems is significantly limited by high computational costs, architectural implementation complexity, and difficulties in real-time data processing.

The purpose of this study is to investigate how architectural differences among YOLOv5, YOLOv8, YOLOv9, and YOLOv11 affect military-vehicle recognition under a common data partition and experimental protocol. The work is formulated as an empirical comparative case study rather than as a proposal for a new detector architecture. Attention is paid to overall detection quality, class-wise recall, and the effect of pronounced annotation imbalance on fine-grained recognition of tank, truck, and jeep classes.

2. Related works

Modern approaches to object recognition in high-spatial-resolution imagery fall into several categories: region-based convolutional models, single-stage detectors, transformer-based methods, and hybrid architectures. Early studies relied on discriminative-structural models and classical machine learning techniques, such as support vector machines with feature fusion [9]; however, these approaches showed limited robustness to background variation and poor scalability to high-dimensional imagery.

Region-based convolutional networks (R-CNN, Fast R-CNN, Faster R-CNN) [10] improved detection accuracy through a two-stage region-proposal-and-classification scheme, but their high inference latency limits applicability in real-time aerial reconnaissance. Single-stage detectors such as YOLO and SSD address this limitation by performing bounding-box regression directly from image features; studies [11, 12] confirm the effectiveness of YOLO-based modifications for military vehicle recognition, particularly when combined with attention mechanisms, deformable convolutions, and augmentation strategies, though performance still degrades for extremely small targets in cluttered scenes.

Transformer-based models, such as DETR [13], enable end-to-end recognition with global context modeling but suffer from high computational cost, slow convergence, and reduced effectiveness on small objects in high-resolution imagery. Hybrid architecture combining CNN and transformer strengths, including multi-scale feature fusion methods [14], partially mitigate these issues by improving robustness to blur and low contrast.

Recent studies further demonstrate the practical potential of YOLO-based architectures for aerospace target detection. In [15], a comparative evaluation of YOLO v8–v11 across raw, fine-tuned,

and preprocessing-enhanced settings showed that the proposed preprocessing method reduced the false-negative rate from 19.5% to 3.2% at a 0.75 confidence threshold, thereby improving target visibility under low contrast and background clutter. In [16], a modified YOLOv11-OBB architecture — incorporating Freq-SpaFEM, SPPF, DAGLF, AMSF2, and RAFA modules — achieved an mAP@50 of 96.8%, precision of 99.5%, recall of 96.8%, and OBB IoU of 98.5% for vehicle detection in complex high-resolution scenes, with reported applicability to urban planning and traffic monitoring.

Military vehicle recognition has also been studied as a distinct problem domain, largely centered on the MSTAR SAR dataset, which covers ten classes of ground military vehicles acquired via X-band radar with full aspect-angle coverage [17]. Comparative studies using this dataset report high classification accuracy with standard CNN backbones (VGG16, InceptionV3), while noting that SAR-specific speckle noise remains a key limiting factor [18]. Complementary work has focused on low-altitude optical aerial imagery, emphasizing the inadequacy of general-purpose datasets (PASCAL VOC, ImageNet) for military surveillance tasks [19], while other studies apply transfer learning to aircraft recognition using open-source imagery [20].

Overall, the reviewed literature confirms substantial progress in object detection for aerial and military-domain imagery; however, direct comparison between reported results remains difficult because existing studies employ different datasets, acquisition modalities, preprocessing procedures, and model configurations.

Despite the substantial progress reported in the studies reviewed above, three limitations remain particularly relevant to the present task. First, published results are usually obtained on different datasets, acquisition modalities, preprocessing pipelines, and training configurations, which prevents a direct attribution of performance differences to detector generation alone. Second, the effect of severe class imbalance on fine-grained recognition of individual military-vehicle types is rarely examined under a common protocol. Third, more complex two-stage, transformer-based, and hybrid detectors may provide strong accuracy but introduce additional computational and implementation overhead when low-latency processing is required. These limitations motivate a controlled within-family comparison in which selected YOLO generations are evaluated using the same data partition and fixed training budget, with explicit analysis of class-wise errors and annotation imbalance.

Accordingly, the experimental scope is deliberately restricted to selected YOLO generations because they share a single-stage real-time detection paradigm and a broadly comparable deployment workflow. The study is not intended as an exhaustive benchmark against every detector family, and its conclusions are limited to the models evaluated here.

3. Materials and methods

Figure 1 presents the general pipeline used in this study to compare YOLO-based neural network architectures for military vehicle recognition in aerospace imagery. The pipeline consists of five sequential stages: 1) acquisition of input aerospace imagery; 2) data preprocessing and annotation; 3) dataset partitioning into training, validation, and test subsets; 4) parallel training of four YOLO architectures (YOLOv5, YOLOv8, YOLOv9, and YOLOv11) using the same dataset partition on the identical dataset; and 5) performance evaluation and comparative analysis, resulting in the selection of the most efficient architecture and the corresponding trained weights, which are subsequently applied to the detection of military vehicles in previously unseen aerospace images.

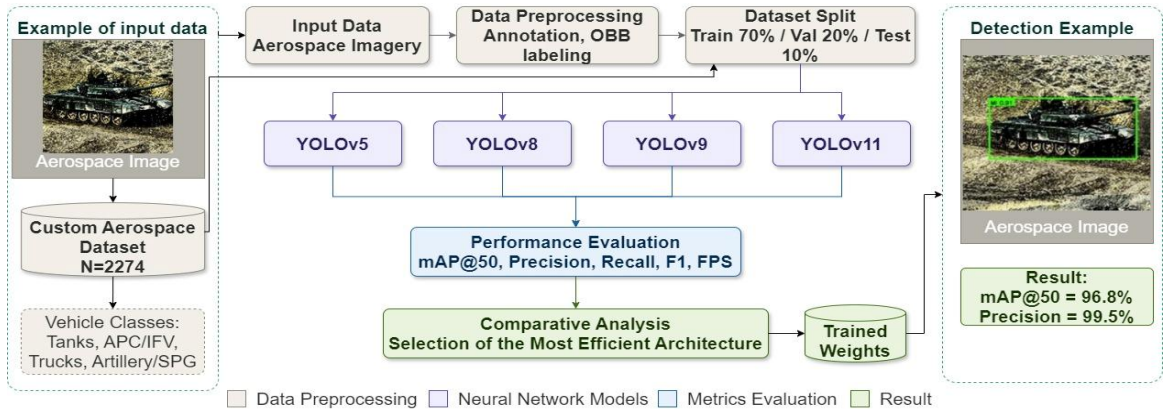


Figure 1: Scheme of the proposed approach

3.1. Input data

The first stage of the pipeline consists of forming an author-created custom dataset of aerospace images. The collection was assembled and annotated by the authors from heterogeneous publicly accessible imagery and aerial-photography materials, including satellite- and unmanned aerial vehicles (UAV)-style views. The final dataset contains 2,274 aerospace images. Because the collection was created from heterogeneous sources rather than from a single controlled acquisition campaign, complete sensor-level metadata such as platform, ground sampling distance, viewing geometry, altitude, and acquisition date are not available uniformly for every image. These variables are therefore not used as experimental factors, and the conclusions are restricted to aggregate image-level recognition performance on the curated dataset.

3.2. Data preprocessing and annotation

The second stage comprises preprocessing, label harmonization, and annotation. The dataset was created by the authors by collecting images from publicly accessible web resources, open-access platforms, and aerial-photography materials. No separate image-enhancement method was introduced as an experimental contribution. Preprocessing consisted of removing out-of-scope labels, harmonizing duplicated class names, converting annotations to the input format required by the detector implementations, and using a common 640×640 input resolution during training and evaluation. Preprocessing was held fixed in the comparison and was not treated as an independent experimental variable.

In this work we performed object annotation in the Computer Vision Annotation Tool (CVAT). Each retained target was assigned a class label and a bounding-box localization (Fig. 2). Annotation quality control included a subsequent label-harmonization pass in which out-of-scope labels were removed, duplicated class names were merged, and obvious class-naming inconsistencies were corrected before the dataset split. The study did not perform a formal inter-annotator agreement experiment; this is reported as a limitation rather than inferred retrospectively.

The original annotation schema included eight class labels. Three labels (base, military, troops) did not represent the target transport-vehicle classes and were excluded, while two labels (Military-Vehicles and military-vehicle) represented duplicate naming of the same generic category and were merged. The harmonized label set therefore contains four relevant classes: military-vehicle, tank, truck, and jeep. This harmonization was performed before the dataset split so that the same label taxonomy was used in training, validation, and testing.

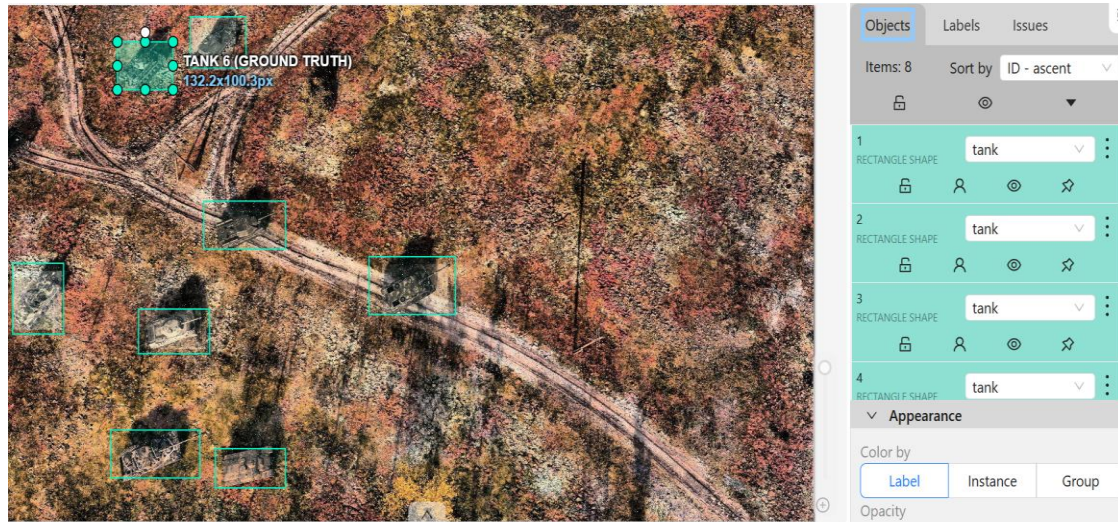


Figure 2: Data annotation process using CVAT

3.3. Dataset partitioning

At the third stage, the 2,274-image dataset was divided into 1,592 training images, 341 validation images, and 341 test images, corresponding approximately to a 70%/15%/15% split. The same partition was retained for all four detector runs so that the compared models were trained and evaluated using identical image subsets.

Table 1 summarizes the resulting class distribution across the dataset splits, where column “Images, n” denotes the number of images in each dataset subset. The class-specific columns denote the number of annotated object instances (bounding-box annotations), not the number of images, and not percentages. Multiple annotated objects may occur in the same image. Thus, for example, the value 76 for the tank class denotes 76 annotated tank instances in the complete dataset.

Table 1

Distribution of images and annotated object instances across dataset splits

Split	Images, n	military-vehicle annotated objects, n	tank, annotated objects, n	truck, annotated objects, n	jeep, annotated objects, n	Total annotated objects, n
Train	1.592	7.918	53	30	17	8.018
Valid	341	1.697	12	6	4	1.719
Test	341	1.697	11	6	3	1.717
Total	2.274	11.312	76	42	24	11.454

As shown in Table 1, the complete dataset contains 2,274 images divided into 1,592 training, 341 validation, and 341 test images, corresponding approximately to a 70%/15%/15% split. The four retained classes contain 11,454 annotated object instances in total. The generic military-vehicle class accounts for 11,312 annotations (98.76%), whereas the tank, truck, and jeep classes account for 76 (0.66%), 42 (0.37%), and 24 (0.21%) annotations, respectively.

3.4. YOLO architecture fundamentals

All four architectures considered in this study (YOLOv5, YOLOv8, YOLOv9, and YOLOv11) belong to the single-stage "You Only Look Once" family of object detectors, which formulates object

detection as a single regression problem: given an input image, the network directly predicts bounding box coordinates and class probabilities in one forward pass, without a separate region-proposal stage. This design choice enables YOLO-based models to achieve inference speeds suitable for near-real-time aerial reconnaissance, compared with two-stage detectors such as Faster R-CNN. For each predicted bounding box, the network outputs an objectness score P_{obj} , indicating the estimated probability that the box contains an object, and a set of conditional class probabilities $P(c_i|obj)$ for each class c_i . The final confidence score assigned to a detection of class c_i is computed as [21]:

$$Conf(c_i) = P_{obj} \cdot P(c_i|obj). \quad (1)$$

The overall training objective is a composite loss function combining localization, objectness, and classification terms [22]:

$$L = \lambda_{box}L_{box} + \lambda_{obj}L_{obj} + \lambda_{cls}L_{cls}, \quad (2)$$

where L_{box} penalizes discrepancies between predicted and ground-truth box coordinates (typically via a CIoU or DFL loss), L_{obj} penalizes incorrect objectness predictions (binary cross-entropy), L_{cls} penalizes incorrect class assignment, and λ_{box} , λ_{obj} , λ_{cls} are the corresponding loss weighting coefficients.

The degree of overlap between a predicted box B_p and a ground-truth box B_{gt} is quantified using Intersection over Union [23]:

$$IoU(B_p, B_{gt}) = \frac{Area(B_p \cap B_{gt})}{Area(B_p \cup B_{gt})}. \quad (3)$$

A predicted box is considered a true positive if $IoU \geq \tau$ for a predefined threshold τ (commonly $\tau=0.5$, denoted mAP@50). While sharing the same single-stage detection paradigm, the four evaluated architectures differ substantially in their network design. YOLOv5 employs a CSP-based convolutional backbone with SPPF and multi-scale feature aggregation. YOLOv8 introduces C2f modules and an anchor-free decoupled detection head. YOLOv9 incorporates Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN) to improve gradient flow and information preservation. YOLOv11 further modifies feature extraction using C3k2 and C2PSA modules while retaining an efficient anchor-free detection design. These architectural differences provide the basis for comparing the recognition characteristics of the four models under a common experimental protocol.

3.5. Evaluation metrics

All four models were trained on the same author-created dataset partition, comprising 1,592 training images, 341 validation images, and 341 test images. A common input resolution of 640×640 pixels and a fixed training budget of 100 epochs were used for all four architectures. The same image subsets were retained across all runs to provide a consistent basis for comparison.

No systematic hyperparameter search or ablation study was performed. The comparison, therefore, represents the models under the fixed experimental protocol used in this study rather than the individually optimized upper bound of each architecture. Implementation-level settings that were not consistently preserved in the archived run records are not reconstructed retrospectively because doing so would introduce unverifiable information. Accordingly, the manuscript reports only the settings that can be verified across all runs data partition, 100 training epochs, and 640×640 input

resolution, and explicitly treats the absence of a hyperparameter-sensitivity analysis as a study limitation.

To quantitatively compare the four architectures, the following standard object detection metrics are computed on the test subset. Precision measures the proportion of correct detections among all positive predictions. Recall measures the proportion of ground-truth objects that were successfully detected. F1-score is the harmonic mean of Precision and Recall, providing a single balanced measure. Mean Average Precision (mAP) summarizes detection accuracy across all classes and confidence thresholds. For a single class, Average Precision (AP) is computed as the area under the Precision–Recall curve [21]. In this study, mAP@50 (computed at $\text{IoU} \geq 0.5$) is reported as the primary accuracy metric, consistent with common practice in the object detection literature. Inference speed (FPS) is measured as the number of images processed per second.

4. Experimental results

4.1. Visual Analysis

Following the training procedure described in Section 3, the four YOLO architectures (YOLOv8, YOLOv9, YOLOv11, YOLOv5) were evaluated on previously unseen aerospace images to assess their practical recognition capability under real-world conditions of complex backgrounds, variable illumination, and arbitrary object orientations. This section presents both qualitative detection outcomes across representative test scenes and, subsequently, the quantitative performance metrics obtained on the held-out test subset. Figures 3–6 present representative qualitative examples of military vehicle detection obtained with the four compared architectures after training on the same author-created dataset described in Section 3, applied to aerospace images spanning diverse background conditions, forested terrain, open grassland, unpaved rural roads, and a mixed settlement/floodplain scene.

Figure 3 (forested area with military vehicles partially occluded by vegetation) illustrates different detection coverage across the four architectures. YOLOv8 produced three detections labeled as truck, YOLOv11 produced two truck detections, and YOLOv5 produced one truck detection, whereas YOLOv9 produced no detections in this scene.



COMPARATIVE DETECTION RESULTS:

■ YOLOv11: TRUCK: 2 (Total: 2)	■ YOLOv8_s: TRUCK: 3 (Total: 3)
■ YOLOv9: No detections	■ YOLOv5: TRUCK: 1 (Total: 1)

Figure 3: Detection results in a forested scene with partially occluded vehicles.

Figure 4 (open grassland scene) contains six target objects belonging to the tank class. YOLOv8 produced 11 detections labeled as tank, whereas YOLOv11 produced seven tank detections. YOLOv9 produced two detections labeled as truck, while YOLOv5 produced no detections. Since the number

of predictions exceeds the number of reference objects, some outputs correspond to duplicate or false-positive detections rather than additional ground-truth tanks.



COMPARATIVE DETECTION RESULTS:

■ YOLOv11: TANK: 7 (Total: 7)	■ YOLOv8_s: TANK: 11 (Total: 11)
■ YOLOv9: TRUCK: 2 (Total: 2)	■ YOLOv5: No detections

Figure 4: Detection results in an open grassland scene with multiple tank-type vehicles.

Figure 5 (unpaved road scene) contains five target objects belonging to the tank class; a civilian minibus is also visible but is not included as a target object. YOLOv11 produced seven detections labeled as tank, whereas YOLOv8 produced five tank detections. YOLOv9 produced one detection labeled as truck, while YOLOv5 produced three detections labeled as truck. In addition, YOLOv5.



COMPARATIVE DETECTION RESULTS:

■ YOLOv11: TANK: 7 (Total: 7)	■ YOLOv8_s: TANK: 5 (Total: 5)
■ YOLOv9: TRUCK: 1 (Total: 1)	■ YOLOv5: TRUCK: 3 (Total: 3)

Figure 5: Detection results for an unpaved road scene, including a false positive.

Figure 5 (unpaved road scene) shows a more balanced spread of detections across architectures: YOLOv11 identified seven tank instances, YOLOv8 identified five, YOLOv5 identified three objects labeled truck, and YOLOv9 identified one truck instance. Notably, YOLOv5 additionally produced a false-positive detection, labeling a civilian minibus as a truck. The scene in Fig.5 contains five target objects belonging to the tank class; a civilian minibus is also visible but is not included as a target

object. YOLOv11 produced seven detections labeled as tank, whereas YOLOv8 produced five tank detections. YOLOv9 produced one detection labeled as truck, while YOLOv5 produced three detections labeled as truck. In addition, YOLOv5 incorrectly classified the civilian minibus as a truck, representing a false-positive detection. Figure 6 (single tank near a riverbank in a mixed settlement/floodplain scene) represents a low-salience detection case: only YOLOv11 and YOLOv8 successfully localized and classified the vehicle as tank, while YOLOv9 and YOLOv5 failed to detect the object entirely. The reference annotation for Figure 6 contains one target object belonging to the tank class. YOLOv8 and YOLOv11 both detected and classified this object as a tank, whereas YOLOv9 and YOLOv5 produced no detection.



COMPARATIVE DETECTION RESULTS:

YOLOv11: TANK: 1 (Total: 1)

YOLOv8_s: TANK: 1 (Total: 1)

YOLOv9: No detections

YOLOv5: No detections

Figure 6: Detection results for a low-salience target near a riverbank.

The detection counts obtained across all four test scenes (Figures 3–6) are summarized in Table 2.

Table 2

Reference object composition and per-scene detection counts for the compared architectures

Scene	Reference objects	YOLOv11	YOLOv8	YOLOv9	YOLOv5
Fig. 3 - Forested area	5 truck-class targets	2 (truck)	3 (truck)	0	1
Fig. 4 - Open grassland	6 tanks	7 (tank)	11 (tank)	2 (truck)	0
Fig. 5 - Unpaved road	5 tanks	7 (tank)	5 (tank)	1 (truck)	3 (truck, incl. 1 FP)
Fig. 6 - Riverside settlement	1 tank	1 (tank)	1 (tank)	0	0

4.2. Quantitative evaluation

To complement the qualitative results presented in Section 4.1, all four architectures were evaluated quantitatively following the metrics defined in Precision, Recall, mAP@50, mAP@50–95, and per-

class recognition accuracy derived from the confusion matrix. To ensure a fair comparison, YOLOv8, YOLOv9, YOLOv11, and YOLOv5 (Figure 7) were all trained under identical conditions (100 training epochs, input resolution 640×640, identical training/validation/test split), eliminating the data- and configuration-related inconsistencies present in earlier evaluation attempts. Precision represents the proportion of accepted detections that correctly correspond to ground-truth objects, whereas Recall represents the proportion of ground-truth objects successfully retrieved by the detector. The F1-score summarizes the balance between Precision and Recall through their harmonic mean. Average Precision (AP) summarizes the precision–recall relationship for an individual class, whereas mean Average Precision (mAP) averages AP across the evaluated classes. In this study, mAP@50 is evaluated at an IoU threshold of 0.50, while mAP@50–95 summarizes performance over progressively stricter IoU thresholds from 0.50 to 0.95. Across all four architectures, the loss components decreased steadily, and the accuracy metrics increased consistently over the training epochs, confirming stable model convergence under the common experimental protocol rather than a trivial or degenerate solution. Table 3 summarizes the final-epoch validation metrics for all four architectures.

Table 3
Overall validation performance metrics of the compared YOLO architectures

Metric	YOLOv8	YOLOv11	YOLOv9	YOLOv5
Precision	≈0.65	≈0.65	≈0.45	≈0.52
Recall	≈0.35	≈0.28	≈0.28	≈0.20
mAP@50	≈0.30	≈0.28	0.29	0.23
mAP@50–95	≈0.14	≈0.13	≈0.15	≈0.14
F1-score (approx.)	≈0.46	≈0.39	≈0.35	≈0.29

To examine recognition accuracy for individual vehicle types, the normalized confusion matrices for all four architectures were analyzed, and the corresponding per-class recall values (the matrix diagonal) are summarized in Table 4. To assess the recognition of individual vehicle classes, the diagonal elements of the normalized confusion matrices were interpreted as class-wise Recall [22]. Thus, for example, the value 0.11 for the jeep class indicates that approximately 11% of ground-truth jeep instances were correctly detected and assigned to the jeep class under the applied evaluation settings. It does not mean that 11% of the images were correctly recognized.

Table 4
Per-class recall derived from the normalized confusion matrices

Class	YOLOv8	YOLOv11	YOLOv9	YOLOv5
military-vehicle	0.66	0.58	0.62	0.71
tank	0.17	0.17	0.17	0.22
jeep	0.11	0.11	0.11	0.00
truck	0.14	0.11	0.4	0.00

The curves in Fig.7 were generated by varying the detection-confidence threshold and recalculating Precision for the predictions retained at each threshold. The horizontal axis represents the confidence threshold, whereas the vertical axis represents Precision. The near-vertical segment of the green truck-class curve is not a manually introduced threshold. It represents a discrete change in Precision when the confidence threshold passes individual truck predictions. The step-like behavior is particularly pronounced for this class because of its limited number of annotated instances.

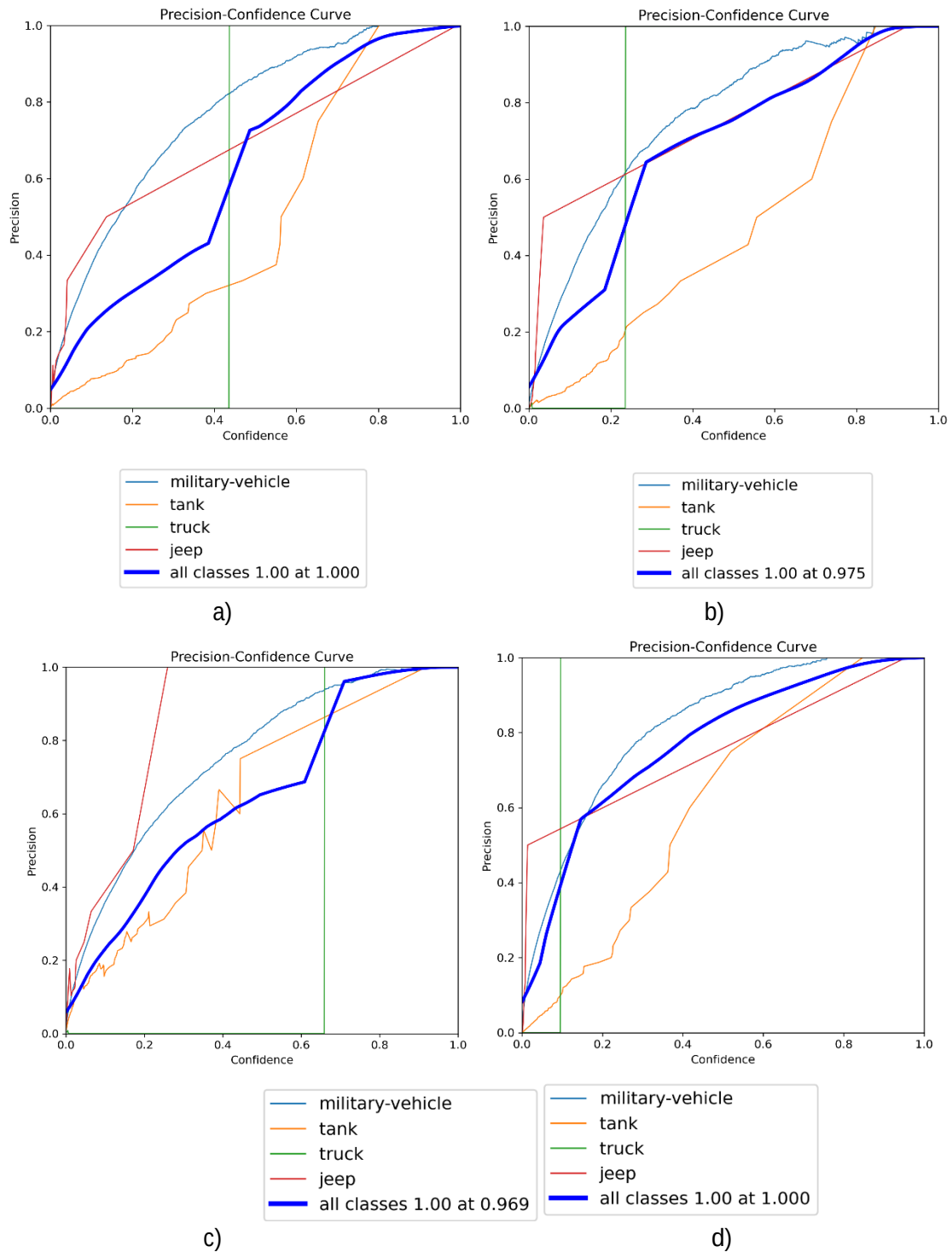


Figure 7: Precision–confidence curves for: a) YOLOv8; b) YOLOv11; c) YOLOv9; d) YOLOv5.

5. Discussion

5.1. Visual analysis

The quantitative results largely corroborate the qualitative observations and indicate that YOLOv8 provides the most favorable overall balance among the four evaluated architectures. As shown in Table 3, YOLOv8 achieved the highest Recall (≈ 0.35) while maintaining Precision of ≈ 0.65 and mAP@50 of ≈ 0.30 . YOLOv9 achieved a comparable mAP@50 of ≈ 0.29 but lower Precision (≈ 0.45), whereas YOLOv11 showed Precision of ≈ 0.65 with a lower Recall of ≈ 0.28 . YOLOv5 showed the lowest Recall (≈ 0.20) and mAP@50 (≈ 0.23), with Precision of ≈ 0.52 . These results indicate that YOLOv8 provides the most favorable balance between detection completeness and precision among the four architectures evaluated in this study. Figure 3 (forested area with military vehicles partially occluded by vegetation) illustrates different detection coverage across the four architectures. YOLOv8 produced three detections labeled as truck, YOLOv11 produced two truck detections, and YOLOv5 produced one truck detection, whereas YOLOv9 produced no detections in this scene. This visual pattern is consistent with the operational challenges outlined in Section 1: the presence of small-sized objects against heterogeneous background textures, partial occlusion by vegetation (Figure 3). Across the four representative scenes, YOLOv8 and YOLOv11 produced the largest numbers of detections, with 20 and 17 detections, respectively, whereas YOLOv5 and YOLOv9 produced 4 and 3 detections, respectively (Table 2). These qualitative examples illustrate differences in detection coverage; however, the overall ranking of the architectures is based on the dataset-level quantitative metrics reported in Section 4.2 rather than on these four selected scenes alone. The false-positive case shown in Figure 5, in which YOLOv5 classified a civilian minibus as a truck, illustrates that aggregate performance metrics do not fully capture scene- and class-specific failure modes, particularly for classes with a limited number of annotated instances. This highlights a practical risk for real-world deployment: aggregate Precision values can mask specific, operationally significant failure modes that are only revealed through visual inspection of individual detections.

5.2. Quantitative analysis

The quantitative results indicate that YOLOv8 provides the most favorable overall balance among the four evaluated architectures. As shown in Table 3, YOLOv8 achieved the highest Recall (≈ 0.35) while maintaining Precision of ≈ 0.65 and mAP@50 of ≈ 0.30 . YOLOv9 achieved a comparable mAP@50 of ≈ 0.29 but lower Precision (≈ 0.45), whereas YOLOv11 showed Precision of ≈ 0.65 with a lower Recall of ≈ 0.28 . YOLOv5 showed the lowest Recall (≈ 0.20) and mAP@50 (≈ 0.23), with Precision of ≈ 0.52 . Thus, YOLOv8 provides the most favorable balance between detection completeness and precision among the architectures evaluated in this study.

Taken together, these results indicate that YOLOv8 offers the most favorable overall balance among the four architectures for military vehicle recognition on aerospace imagery, combining the highest Recall with competitive Precision and mAP@50, and is therefore identified as the recommended architecture for this task among those evaluated. YOLOv9 remains a viable alternative when maximizing aggregate mAP@50 is the primary objective, while YOLOv5's markedly lower Recall makes it the least suitable choice among the four when detection completeness is a priority. The confusion matrices (Table 4) reveal a consistent failure pattern across all four architectures, independent of network design: the dominant source of error is not inter-class confusion (e.g., mistaking a tank for a truck) but rather incomplete detection, with a substantial share of tank, truck, and jeep instances assigned to the background class. Recall for these three minority classes remained considerably lower than for the generic military-vehicle category (58–71% across the four models): tank recall ranged narrowly from 17% to 22%, jeep recall ranged from 0% (YOLOv5) to 11% (YOLOv8, YOLOv11, YOLOv9), and truck recall varied most widely, from 0% (YOLOv5) to 40% (YOLOv9). Notably, even YOLOv8 — the best-performing architecture overall — showed only modest recall on the minority classes (tank: 17%, jeep: 11%, truck: 14%), confirming that its overall advantage stems

primarily from stronger recognition of the well-represented military-vehicle category rather than from a fundamentally better ability to discriminate rare vehicle types. This pattern across architecturally distinct models (single-stage anchor-based YOLOv5, anchor-free YOLOv8/YOLOv11, and PGI/GELAN-based YOLOv9) provides strong evidence that the performance ceiling for minority classes is primarily data-limited rather than architecture-limited. In other words, while YOLOv8's architectural design yields a measurable overall advantage, no architecture alone, including YOLOv8, can achieve military-vehicle-level recognition accuracy for tanks, trucks, or jeeps without a corresponding increase in the number and diversity of annotated training instances for these classes. This confirms that dataset expansion and class rebalancing remain the primary lever for improving fine-grained vehicle-type recognition. The absolute performance values obtained in this study are lower than those reported in several recent YOLO-based aerospace detection studies. For example, the preprocessing-enhanced approach in [15] substantially reduced false-negative detections under low-contrast conditions, whereas the modified YOLOv11-OBb architecture in [16] achieved considerably higher precision, recall, and mAP values on a different high-resolution vehicle dataset. However, direct numerical comparison is not appropriate because these studies differ in target classes, image sources, annotation policies, preprocessing procedures, and model modifications. In contrast, the present study employs a strongly imbalanced author-created dataset, in which the generic military-vehicle class represents 98.76% of the retained annotations. The low recall observed for tank, truck, and jeep is therefore consistent with the limited representation of these fine-grained classes. This observation also agrees with [19], which emphasizes the limitations of general-purpose datasets for military aerial recognition.

6. Conclusions

This study addressed the problem of recognizing military transport vehicles in aerospace imagery by comparing four YOLO-based neural network architectures: YOLOv8, YOLOv9, YOLOv11, and YOLOv5 trained and evaluated under identical conditions on a custom, purpose-built dataset of 2,274 annotated aerospace images. A custom dataset was collected and annotated using the CVAT tool to correctly represent vehicles of arbitrary orientation, a key requirement for aerospace imagery. Preliminary analysis of the original annotation schema revealed substantial class imbalance and several non-vehicle labels, which were subsequently filtered and merged into four relevant categories: a generic military-vehicle class and three specific vehicle types – tank, truck, and jeep. All four architectures were trained on a unified dataset using a common set of hyperparameters, ensuring the comparison reflects genuine architectural differences rather than inconsistencies in the training data or configuration. Under the reported fixed configuration, YOLOv8 achieved the most favorable overall validation performance, with Precision ≈ 0.65 , Recall ≈ 0.35 , and mAP@50 ≈ 0.30 . YOLOv9 achieved a comparable mAP@50 of ≈ 0.29 but lower Precision, YOLOv11 had similar Precision with lower Recall, and YOLOv5 showed the lowest Recall (≈ 0.20) and mAP@50 (≈ 0.23). Class-wise recall remained much lower for tank, truck, and jeep than for the dominant generic class, demonstrating that the selected architecture alone did not overcome the limitations imposed by the highly imbalanced annotation distribution. Based on these findings, YOLOv8 is recommended as the most suitable architecture among those evaluated for military vehicle recognition tasks, prioritizing detection completeness, such as wide-area aerial reconnaissance, while dataset expansion and class rebalancing – through additional data collection, targeted augmentation of underrepresented vehicle types, or synthetic data generation should be prioritized over further architectural experimentation as the primary means of improving fine-grained recognition accuracy. The consistent detection-versus-background failure pattern observed across all architectures further indicates that future work should specifically address the recognition of visually similar or partially occluded objects, rather than focusing solely on inter-class discrimination; it should also be noted that the results reported in this study were obtained under a constrained training budget, and extending the training duration while restoring full input resolution is expected to further improve absolute performance while preserving the relative architectural ranking established here. Future research directions

include extending the dataset with additional annotated instances of tank, truck, and jeep vehicles from diverse geographic and environmental conditions; incorporating additional vehicle categories relevant to defense applications, such as self-propelled artillery and air-defense systems; evaluating hybrid CNN–transformer architectures and multi-scale feature fusion techniques for improved small-object detection; and assessing the real-time inference performance of the compared architectures on embedded and edge-computing platforms suitable for deployment in UAV-based reconnaissance systems.

Acknowledgements

This research is carried out as part of the scientific project no.0126U000995 “Intelligent technologies for analyzing spatiotemporal changes in aerospace imagery to support decision-making under conditions of armed aggression,” funded by the Ministry of Education and Science of Ukraine at the expense of the state budget.

Declaration on Generative AI

The authors used Grammarly to check the grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication’s content.

References

- [1] K. Robinson, B. McKenna, D. Rooney, The relationship of risk to rules, values, virtues, and moral complexity: What we can learn from the moral struggles of military leaders, *J. Bus. Ethics* 179 (2022) 749–766. doi:10.1007/s10551-021-04842-4.
- [2] H. Peng, Y. Zhang, S. Yang, B. Song, Battlefield image situational awareness application based on deep learning, *IEEE Intell. Syst.* 35 (2019) 36–43. doi:10.1109/MIS.2019.2942701.
- [3] C. Mei, Z. Fan, Q. Zhu, P. Yang, Z. Hou, H. Jin, A Novel scene matching navigation system for UAVs based on vision/inertial fusion, *IEEE Sens. J.* 23 (2023) 6192–6203. doi:10.1109/JSEN.2023.3242045.
- [4] L. Zhou, S. Leng, Q. Liu, Q. Wang, Intelligent UAV swarm cooperation for multiple targets tracking, *IEEE Internet Things J.* 9 (2021) 743–754. doi:10.1109/JIOT.2021.3106339.
- [5] Q. Meng, L. Zhang, Z. Sun, F. Meng, L. Wang, Y. Sun, Characterizing spatial and temporal trends of surface urban heat island effect in an urban main built-up area: A 12-year case study in Beijing, China, *Remote Sens. Environ.* 204 (2018) 826–837. doi:10.1016/j.rse.2017.10.027.
- [6] V. Scuro, Open-Source Intelligence (OSINT) for Researchers and Practitioners, 2025. doi:10.1007/978-3-032-02014-7_2.
- [7] N. Carion, F. Massa, G. Synnaeve, et al., End-to-end object detection with transformers, in: *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Springer, 2020, pp. 213–229. doi:10.1007/978-3-030-58452-8_13.
- [8] W. Wang, et al., Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 568–578. doi:10.1109/ICCV48922.2021.00061.
- [9] A. D. W. Sumari, et al., Automatic target recognition and identification for military ground-to-air observation tasks using support vector machine and information fusion, in: *Proc. Int. Conf. ICT Smart Soc. (ICISS)*, 2022, pp. 1–8. doi:10.1109/ICISS55894.2022.9915201.

- [10] H. Su, et al., Object detection and instance segmentation in remote sensing imagery based on precise mask R-CNN, in: Proc. IGARSS, 2019, pp. 1454–1457. doi:10.1109/IGARSS.2019.8898160.
- [11] Z. Guo, X. Mu, C. Chang, et al., An Enhanced Framework for Small Object Detection with MiddleOrder Interaction and Adaptive Cross-Scale Aggregation, Eng. Appl. Artif. Intell. 130 (2025) 107736. doi:10.1016/j.engappai.2024.107736.
- [12] X. Zhao, Y. Xia, W. Zhang, C. Zheng, Z. Zhang, YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection, Remote Sens. 15 (2023) 3778. doi:10.3390/rs15153778.
- [13] F. S. Alrayes, N. Ahmad, A. Alshuhail, et al., Convolutional Transform Learning-Based Fusion for Scale-Invariant Long-Term Target Detection in UAVs, Sci. Rep. 15 (2025). doi:10.1038/s41598-024-51123-x.
- [14] S. Chutke, V. V. Kumar, M. Saravanakumar, AI-Enhanced Image Processing for Target Detection and Threat Recognition in Defence Applications, in: Proc. IEEE Conf. Smart Technol. Virtual Environ., 2025.
- [15] V. Kashtan, Y. Radionov, V. Hnatushenko, Aircraft detection in aerial imagery based on YOLO architectures, in: CEUR Workshop Proc., Vol. 3983, 2025, pp. 196–208. URL: <https://ceur-ws.org/Vol-3983/paper15.pdf>.
- [16] V. Hnatushenko, V. Kashtan, O. Kazymyrenko, Information technology for detecting cars on aerial imaging using a modified YOLO-OBb architecture, in: CEUR Workshop Proc., Vol. 4005, 2025, pp. 293–304. URL: <https://ceur-ws.org/Vol-4005/paper20.pdf>.
- [17] Z. Geng, Y. Xu, B.-N. Wang, X. Yu, D.-Y. Zhu, G. Zhang, Target Recognition in SAR Images by Deep Learning with Training Data Augmentation, Sensors 23 (2023) 941. doi:10.3390/s23020941.
- [18] S. Jacob, J. Wall, S. Sharif, Analysis of Deep Neural Networks for Military Target Classification using Synthetic Aperture Radar Images, in: Proc. 3ICT, 2023, pp. 227–233. doi:10.1109/3ICT60104.2023.10391600.
- [19] F. Kamran, M. Shahzad, F. Shafait, Automated Military Vehicle Detection from Low-Altitude Aerial Images, in: Proc. Digit. Image Comput.: Tech. Appl. (DICTA), 2018, pp. 1–8. doi:10.1109/DICTA.2018.8615865.
- [20] T. Hiippala, Recognizing military vehicles in social media images using deep learning, in: Proc. ISI, 2017. doi:10.1109/ISI.2017.8004875.
- [21] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
- [22] V. Afifah, S. Erniwati, YOLOv8 for Object Detection: A Comprehensive Review of Advances, Techniques, and Applications, Int. J. Adv. Comput. Inf. 2 (2025) 53–61. doi:10.71129/ijaci.v2i1.pp53-61.
- [23] M. Tan, R. Pang, Q. Le, EfficientDet: Scalable and Efficient Object Detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10778–10787. doi:10.1109/CVPR42600.2020.01079.