VEHICLE DETECTION IN UAV ORTHOPHOTOS USING YOLO V7-M DEEP LEARNING ALGORITHM

Đuro Krnić[1]*, Anastasija Božić[1], Marko Marković[1], Zoran Sušić[1], Vladimir Bulatović[1]

[1]Faculty of Technical Sciences, Novi Sad, Serbia
*corresponding author: djuro.geo@uns.ac.rs

Paper type: Original scientific paper
Received: 2025-11-28
Accepted: 2025-12-25
Published: 2026-06-30

UDK: 528.715
DOI: 10.14415/JFCE-929

ABSTRACT:

Automatic vehicle detection in orthophoto images obtained by unmanned aerial vehicles (UAVs) can significantly accelerate information extraction for urban planning, transport and agriculture. This paper presents a case study of vehicle detection implemented in the open-source QGIS environment using the Deep neural remote sensing (Deepness) plugin. A UAV equipped with an RGB camera captured the area of interest (AOI), and Detection was performed using a You Only Look Once (YOLO) v7-m model. The model successfully detected 279 passenger vehicles. High detection accuracy was achieved, as demonstrated by the precision, recall, and F1-score parameters. The results show that the Deepness plugin provides a practical and efficient solution for fast vehicle detection in UAV orthophotos.

KEYWORDS:

UAV, Deepness, Photogrammetry, QGIS, YOLO, vehicle detection

1 INTRODUCTION

The detection of vehicles in aerial images is nowadays widely applied in many applications and is crucial in deriving up-to-date information for traffic monitoring, security-oriented vehicle tracking, parking lot analysis and planning, etc. Therefore, it has attracted growing interest in both the research community and industrial practice [1]. The main challenges of vehicle detection in aerial images, compared to object detection in ground-view images, are the smaller object scale, the often-monotonous appearance, and complex background. Since vehicles are frequently arranged in dense parking configurations, it is difficult to separate them and reliably distinguish them from surrounding structures [2, 3].

Before the emergence of deep learning, vehicle detection in aerial imagery relied mainly on hand-crafted features combined with classifiers. Hand-crafted features lack generalization ability, and the adopted classifier needs to be modified to adapt to these features. Furthermore, these methods localize candidates using a sliding-window search, an approach that is inefficient and results in costly, largely redundant computation [1].

Advances in deep convolutional neural networks have led to the widespread adoption of modern object detectors for vehicle detection. These detectors are typically categorized into two-stage, region-based approaches (e.g., Region-based Convolutional Neural Network (R-CNN) and Faster Region-based Convolutional Neural Network (Faster R-CNN)) and one-stage detectors (e.g., YOLO). The two-stage method first generates region proposals followed by classification, whereas one-stage detector directly predicts object locations and classes in a single forward pass [1, 4]. One-stage YOLO models are most commonly used due to their high speed and strong accuracy in vehicle detection. This can be achieved by leveraging a pre-trained network, trained on other large-scale datasets [5, 6].

Given the previously mentioned challenges, vehicle detection on UAV-derived orthophotos is conducted using the Deepness plugin in QGIS. Deepness is an open-source QGIS plugin that allows running deep-learning models on raster layers, supporting tasks such as segmentation, regression, and object detection [7, 8, 9]. In this study, Deepness serves as an interface between deep-learning prediction results and standard GIS workflows. In this work, the detection model is applied to a UAV-derived orthophoto to identify vehicles. Object detection using Deepness is performed by deploying a pre-trained YOLOv7-m model that localizes and classifies objects by predicting their bounding boxes in the orthophoto [7]. The model is trained on the ITCVD dataset, thereby reducing the need to collect and annotate a large-scale dedicated training dataset. The described approach makes the workflow more time-efficient and more practically applicable in cases where it can be applied.

The aim of this study is to detect and count vehicles by applying the selected model to a UAV-derived orthophoto. In addition, the detection accuracy is assessed, and the model's limitations are examined. In particular, the study provides a quantitative evaluation of a YOLOv7-m-based one-stage detector on high-resolution UAV-derived orthophotos and highlights its principal strengths and systematic shortcomings to be addressed in future research.

2 BACKGROUND AND METHODS

This section presents the methodological background and processing chain adopted in the study, from UAV image acquisition and orthophoto generation to automated vehicle detection and accuracy assessment.

2.1 RELATED WORK

In [5], the authors proposed a YOLOv4-based oriented vehicle detector that predicts oriented bounding boxes, enabling reliable vehicle localization and heading estimation in aerial images. Tested on Dataset for Object Detection in Aerial Images (DOTA) and High-Resolution Ship Collection 2016 (HRSC2016), it achieves strong accuracy with real-time inference speed. The study [10] compared YOLOv3, YOLOv4, and Faster R-CNN on two UAV datasets and found that their accuracy–speed trade-off strongly depends on the dataset. For example, Faster R-CNN performed best on Stanford, while YOLOv4 was superior and faster on Prince Sultan University dataset. An improved YOLOv5-VTO model for detecting tiny and occluded vehicles in aerial images is introduced in [11] by adding a P2 detection branch, using Bidirectional Feature Pyramid Network multi-scale feature fusion, and applying soft non-maximum suppression (Soft-NMS), which yields better performance than YOLOv5s. In addition, numerous other works have addressed YOLO-based vehicle detection in UAV and aerial imagery (e.g., [12, 13, 14]). These studies confirm the applicability of YOLO models, while also highlighting persistent challenges related to small-object detection.

2.2 WORKFLOW

Automated extraction of vehicles from the orthophoto requires a sequence of key processing steps, as illustrated in Figure 1. First, the flight mission was planned, and UAV imagery of the AOI was captured. The acquired UAV imagery was processed through a standard photogrammetric workflow to generate a georeferenced orthophoto used for further analysis. The resulting orthophoto was used as the input for vehicle detection using the Deepness plugin, yielding a prediction mask. This output layer was later used to perform an accuracy assessment in order to reveal the advantages and limitations of this model.

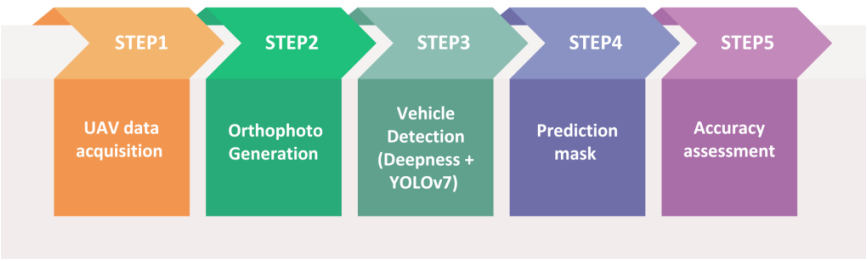

Figure 1: Workflow for automated vehicle extraction from UAV-derived orthophotos using the Deepness (YOLOv7-m) plugin in QGIS

2.3 UAV SPECIFICATION & DATA ACQUISITION

The study was carried out over the University campus area in Novi Sad, covering 3.70 ha. UAV imagery was acquired using a DJI Mavic 3 Pro platform. The specifications of this platform are listed in Table 1.

Table 1: Key specifications of the DJI Mavic 3 Pro [15]

Parameter	Specification
Weight (with propellers)	958 g
Max flight time (no wind)	43 min
Hasselblad Camera	4/3 CMOS, Effective Pixels: 20 MP
Medium Tele Camera	1/1.3" CMOS, Effective Pixels: 48 MP
Tele camera	1/2" CMOS, Effective Pixels: 12 MP
GNSS	GPS + Galileo + BeiDou

The flight mission lasted approximately 6 minutes and was conducted at an altitude of 50 m above ground level. A total of 540 images were acquired, with 85% frontal and side overlap. Each image had a spatial resolution of 5280 × 3956 pixels, with a ground sampling distance (GSD) of 1.35 cm/pixel.

For orthophoto georeferencing, four ground control points (GCPs) were used, evenly distributed across the AOI. The GCPs were materialised as markers, ensuring their clear identification in the UAV imagery. Each GCP was observed using real-time kinematic (RTK) Global Navigation Satellite System (GNSS) measurements. For every GCP, three observation sessions of 30 s were recorded, and the final GCP coordinates were computed as the arithmetic mean of the three epochs. The GCP coordinates were determined in the World Geodetic System 1984 (WGS 84) / Universal Transverse Mercator (UTM) zone 34N coordinate reference system. Prior to their use in the photogrammetric processing workflow, the GCP coordinates were transformed into the Militärgeographisches Institut (MGI) 1901 / Balkans zone 7 projected coordinate reference system (CRS). The mean coordinates of the GCPs, together with their estimated horizontal and vertical precisions, are presented in Table 2.

Table 2: Mean GCP coordinates with estimated horizontal and vertical precision

Point ID	Y [m]	X [m]	H [m]	σ_{hor}[m]	σ_{vert} [m]
1	7410160,050	5011790,781	77,7075	0,006	0,010
2	7410267,303	5011835,760	78,1996	0,010	0,016
3	7410310,843	5011727,563	78,6749	0,024	0,044
4	7410204,719	5011695,613	77,6917	0,011	0,016

The UAV imagery and GCP measurements were processed in Pix4Dmapper software, following a standard Structure-from-Motion/Multi-View Stereo (SfM-MVS) workflow. First, automatic tie-point extraction and image alignment were performed, followed by bundle adjustment with camera self-calibration and the incorporation of GCP observations. A dense point cloud was then generated and used to derive a digital surface model (DSM) and a georeferenced orthophoto with a GSD of 1,35 cm/pixel in the MGI 1901 / Balkans zone 7 coordinate system.

2.4 DEEPNESS PLUGIN AND YOLOV7-M MODEL

Deepness is an open-source QGIS plugin that enables data processing using deep-learning models for object detection, segmentation, and regression. Object detection using Deepness is performed by deploying a pre-trained YOLOv7-m model, which localizes and classifies objects by predicting their bounding boxes in the orthophoto. The model was pretrained on the ITCVD dataset and applied without additional fine-tuning. The raw detections produced by the YOLOv7-m model are first filtered using a confidence threshold to remove low-probability predictions. Non-maximum suppression (NMS) is then applied to eliminate overlapping duplicate boxes, resulting in a single final bounding box per vehicle [7].

Deepness allows running deep-learning models directly on raster layers in QGIS. It supports any framework-agnostic model in Open Neural Network Exchange (ONNX) format by using ONNXRuntime, ensuring broad model and hardware compatibility. The processing concept of this plugin is illustrated in Figure 2, showing how an input orthophoto and an ONNX detection model are integrated within QGIS to produce georeferenced prediction outputs [7, 8].

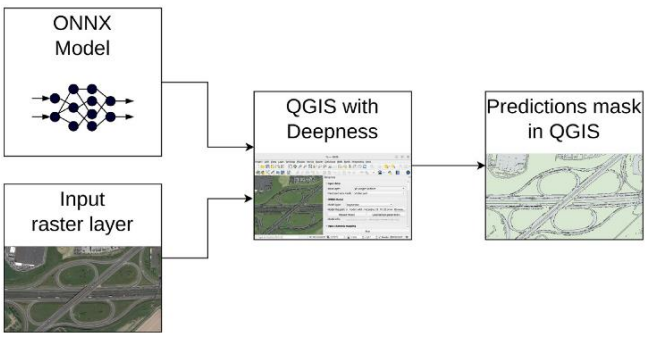

Figure 2: Overview of Deepness plugin workflow for object detection [7]

Within the adopted workflow, the orthophoto is imported into QGIS. The target raster and processing extent are then selected. The detection model is then applied using user-defined parameters, such as confidence threshold. Throughout processing, the orthophoto is divided into small tiles that are processed individually, since the orthophoto exceeds the input size supported by the model. The final output consists of the inference results of these tiles, stitched together into a single layer. After execution, Deepness generates a report with the number of detected vehicles and outputs a prediction mask as a new QGIS layer [7].

2.5 ACCURACY ASSESSMENT

Model performance was quantitatively evaluated using three standard metrics (Equations 1–3). Precision, recall, and F1-score were computed as:

$$Precision = \frac{TP}{TP+FP}, \quad (1)$$

$$Recall = \frac{TP}{TP+FN}, \quad (2)$$

$$F1 - score = \frac{2TP}{2TP+FP+FN}, \quad (3)$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives results detected. TP results denote correctly detected vehicles, i.e., predicted bounding boxes that match a ground-truth vehicle annotation. FP results are predicted boxes that do not correspond to any ground-truth vehicle, while false negatives FN results are ground-truth vehicles that were missed by the detector.

These metrics are widely used in object-detection research as indicators of detection accuracy and robustness.

3 RESULTS AND DISCUSSION

Using the Deepness plugin within the QGIS software package, vehicle detection was performed on an image obtained by UAV photogrammetry. The input image is a georeferenced RGB orthophoto. The Deepness plugin automatically loads the number of bands from the image, which is used to decide which model to use. A pre-trained YOLO v7-m ONNX model of type Detector was used for object detection, with an input size of 640 × 640 pixels.

Before detection, the orthophoto was downscaled and divided into 640 × 640-pixel tiles with 35% overlap in both directions. This overlap was used to reduce missed detections near tile boundaries and to ensure that vehicles crossing tile borders were fully covered in at least one tile. This tiling strategy prevents objects at the edges of tiles from being overlooked and is particularly suitable for small objects, such as passenger vehicles. The 35% overlap was selected as a practical compromise between detection reliability and computational cost. Smaller overlaps increased border-related misses, whereas larger overlaps substantially increased processing time without a noticeable improvement in the results. The predefined model was run with a confidence threshold of 0,25 and an IoU threshold of 0,35. These values were selected based on preliminary results. A confidence threshold of 0,25 provided sufficient sensitivity for smaller or partially occluded vehicles. An IoU threshold of 0,35 was chosen to reduce duplicate detections of the same vehicle. The parameters were determined empirically based on the detection results and the authors' recommendations from reference sources.

The model successfully detected 279 vehicles, all classified as cars. The YOLO detection output is shown in Figure 3, where the predicted bounding boxes are overlaid on the orthophoto. For clarity, Figure 3 displays only a representative subset of the AOI, as displaying the full extent would introduce visual clutter and reduce readability.

The resulting layer was exported to GeoPackage (.gpkg) format for further GIS analysis. For each detected vehicle, the centroid of the corresponding bounding box was calculated, and its X and Y coordinates were stored as additional attributes. The calculated data can be used for distance calculations or for integrating the results into other spatial analyses.

The detection performance was evaluated against a manually created reference (ground-truth) layer. The ground-truth layer was created by visual interpretation of the orthophoto, i.e., by manually digitizing all clearly visible passenger vehicles within the AOI. Annotations

were performed by the first author and verified by the second author. Ambiguous cases were resolved by annotations in the ground-truth layer to clearly visible passenger cars. Using this reference, the numbers of TP, FP, and FN scores were derived. Based on 279 TP, 2 FP, and 5 FN, the resulting metrics were precision of 99,3%, recall of 98,3% and an F1-score of 98,8%. The results indicate robust vehicle detection performance, and the same metrics are often reported in detection studies for objective evaluation and comparison.

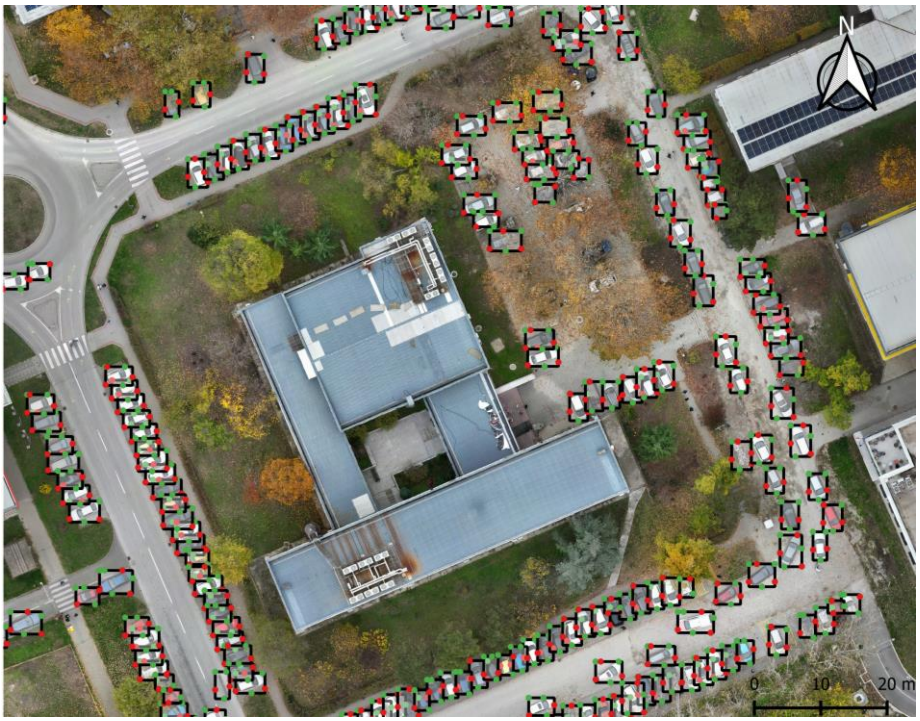

Figure 3: Orthophoto image with detected bounding box of vehicle

The results show high accuracy and reliability, indicating that the model effectively detects vehicles suitable for its purpose. The results demonstrate that the selected Deepness plugin configuration provides an efficient and reliable approach for detection of civilian passenger vehicles in UAV orthophotos. The following image (Figure 4) shows the only two objects that are examples of FP detection in an orthophoto image. In addition, five false negative cases are shown in Figure 5. These examples, the vehicles are partially or completely obscured by dense tree canopy, which reduces their visual contrast and leads to missed detection. These examples highlight that obscuration remains a limitation for reliable vehicle detection in complex scenes. The obtained precision and recall align with previously reported YOLO-based vehicle detection results on UAV/orthophoto data. The observed FP and FN are consistent with common findings in the literature, where false positives typically arise from visually similar objects, while false negatives are mainly driven by occlusion and reduced contrast.

Figure 4: Two FP detections - (left) the roof of a bus stop shelter and (right) stacked construction bricks

Figure 5: Five FN detections - missed vehicle detection

3.1 CONCLUSIONS

In this study, a workflow for automatic vehicle detection in UAV orthophoto images is presented. The implementation was performed entirely within the open source QGIS environment using the Deepness plugin and a pre-trained YOLO ONNX model. The image and model processing parameters used were shown to be sufficient for reliable identification of small passenger cars in an urban environment. The detection results are

provided as a vector layer, containing bounding-box geometries. The associated attributes are centroid coordinates that allow integration into standard GIS analyses.

The results show that under the given conditions, the adopted configuration achieves successful detection of 99,3% of passenger cars in the given environment without a large number of false positives. This case study confirms that standard deep learning models, when carefully parameterized and embedded in GIS software, can significantly reduce manual labour in mapping and tracking. The results show successful detection, provided the detection parameters are carefully selected.

These limitations should be considered when interpreting the results and point to clear directions for further work. Future research directions may focus on expanding the training dataset to include additional classes relevant to other environments and on exploring more advanced artificial intelligence (AI) models. When integrated with other thematic GIS layers, such models have the potential to become a standard tool for rapid and scalable feature analysis on UAV orthophoto.

ACKNOWLEDGEMENTS

This research was supported by the Provincial Secretariat for Sports and Youth of the Autonomous Province of Vojvodina (contract number 000793628 2025 09424 003 000 000 001) in recognition of and financial support provided through the Public Call "Talent Fund" (No. 000254945 2025 09424 003 000 000 001, 3 February 2025).

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. 451-03-137/2025-03/200156) and the Faculty of Technical Sciences, University of Novi Sad through project “Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad 2025” (No. 01-50/295).

FUNDING

This research was financially supported by the Provincial Secretariat for Sports and Youth of the Autonomous Province of Vojvodina (contract number 000793628 2025 09424 003 000 000 001), through the Programme “Fund for Talents”, Public Call (No. 000254945 2025 09424 003 000 000 001, 3 February 2025).

REFERENCES

[1] K. Liu and G. Mattyus, “Fast Multiclass Vehicle Detection on Aerial Images,” IEEE Geoscience and Remote Sensing Letters, 2015. [Online]. Available: https://elib.dlr.de/96765/1/liu_mattyus_jrnl.pdf

[2] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn, “Vehicle Detection in Aerial Images,” 2018. [Online]. Available: https://arxiv.org/abs/1801.07339

[3] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn, "Deep Learning for Vehicle Detection in Aerial Images," in Proc. IEEE ICIP, 2018. [Online]. Available: https://research.utwente.nl/en/publications/deep-learning-for-vehicle-detection-in-aerial-images

[4] K. Pargieła, "Vehicle detection and masking in UAV images using YOLO to improve photogrammetric products," Reports on Geodesy and Geoinformatics, 2022. [Online]. Available: https://sciendo.com/pdf/10.2478/rgg-2022-0006

[5] T.-H. Lin and C.-W. Su, "Oriented Vehicle Detection in Aerial Images Based on YOLOv4," Sensors, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8394

[6] N. Santrač, T. Budimirov, Đ. Krnić, M. Batilović, and V. Bulatović, "Determination of the Position of Solar Modules Using UAV Photogrammetry and Computer Vision," in Proc. SYMOPIS 2025, 2025. [Online]. Available: https://www.symopis2025.fon.bg.ac.rs/download/Programme%20SYMOPIS%202025.pdf

[7] P. Aszkowski et al., "Deepness: Deep neural remote sensing plugin for QGIS," SoftwareX, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711023001917

[8] D. Febrita, "Evaluating AI-Driven Automated Map Digitization in QGIS," arXiv, 26 4 2025. [Online]. Available: https://www.researchgate.net/publication/391246169_Evaluating_AI-Driven_Automated_Map_Digitization_in_QGIS

[9] P. Tresson, P. Le Coz, H. Tulet, A. Malkassian, and M. Réjou Méchain, "IAMAP: Unlocking Deep Learning in QGIS for non-coders and limited computing resources," arXiv, 1 8 2025. [Online]. Available: https://www.researchgate.net/publication/394262411_IAMAP_Unlocking_Deep_Learning_in_QGIS_for_non-coders_and_limited_computing_resources

[10] A. Ammar et al., "Vehicle Detection from Aerial Images Using Deep Learning: A Comparative Study," Electronics, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/7/820

[11] S. Li et al., "Real-Time Vehicle Detection from UAV Aerial Images Based on Improved YOLOv5," Sensors, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/12/5634

[12] M. H. Hamzenejadi and H. Mohseni, "Fine-tuned YOLOv5 for real-time vehicle detection in UAV imagery: Architectural improvements and performance boost," Expert Systems with Applications, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423013477

[13] Z. Chen, L. Cao, and Q. Wang, "YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images," Mobile Information Systems, 2 5 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2022/1828848

[14] S.-B. Wang, Z.-M. Gao, D.-H. Jin, S.-M. Gong, G.-L. Peng, and Z.-J. Yang, "AMEA-YOLO: a lightweight remote sensing vehicle detection algorithm based on attention mechanism and efficient architecture," The Journal of Supercomputing, 22 1 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11227-023-05872-2

[15] DJI, "DJI Mavic 3 Pro – Specifications," DJI, 2023. [Online]. Available: https://www.dji.com/mavic-3-pro/specs