

UDC 004.932.2:004.85:681.5.033

DOI: <https://doi.org/10.17721/1812-5409.2026/1.33>

Yaroslav SMOLIN, PhD Student
ORCID ID: 0009-0007-9744-6496
e-mail: smolin.yaroslav@iill.kpi.ua
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Viacheslav LISKIN, PhD, Assoc. Prof.
ORCID ID: 0000-0002-9418-0633
e-mail: liskin.viacheslav@iill.kpi.ua
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

META-ANALYSIS OF PARETO-OPTIMAL OBJECT DETECTION MODELS FOR UNMANNED AERIAL VEHICLES ON THE VISDRONE2019-DET DATASET

The subject of this research is modern deep learning models for object detection from unmanned aerial vehicles (UAVs), trained and evaluated on the benchmark VisDrone2019-DET dataset. The problem addressed was the lack of a systematic analysis of existing models based on the criterion of optimal balance between detection accuracy, computational speed, and model size, all of which are critical for real-time operation on UAV onboard computers with limited resources. The core result lies in a comprehensive meta-analysis of over 60 scientific publications and the identification of two groups of Pareto-optimal models: one focused on minimizing model parameters and the other on minimizing computational complexity. Through multi-criteria Pareto analysis and the creation of the most complete performance summary table to date, the study identified architectures that offer the best trade-offs between accuracy, speed, and size. These turned out to be optimized versions of YOLO and transformer-based detectors like DETR, a result explained by the efficiency of their core architectures and further tuning for the specifics of aerial imagery. Using the Hailo 8L accelerator as a case study, the research also demonstrated that a model's computational complexity is directly proportional (with a coefficient of determination up to 0.98) to its execution speed on the typical neural accelerator that is used in a UAV companion computer. This allows engineers to estimate relative runtime based on the number of operations, avoiding time-consuming benchmarking. Overall, the findings can be used by engineers to make informed choices about detector architectures depending on specific UAV hardware constraints, and by researchers as reference points for developing new models for autonomous UAV operation in aerial monitoring, infrastructure inspection, and search-and-rescue missions.

Key words: *unmanned aerial vehicles, VisDrone2019, object detection, deep learning models, Pareto analysis.*

AMS 2020 classification: 68T07, 68T10, 90C29.

Introduction

Unmanned aerial vehicles (UAVs) are increasingly used in surveillance, infrastructure inspection, precision farming, rescue operations, traffic monitoring, security operations, and reconnaissance (Cao et al., 2023; Du et al., 2024; Gutierrez et al., 2025; Jiao, & Abdullah, 2024; Li et al., 2023; Zhu et al., 2019). The unique aerial view, mobility, low cost, and ease of use of UAVs provide advantages over both ground-based systems and manned aviation (An et al., 2024; Cao et al., 2023; Hua, & Chen, 2025; Li et al., 2023; Mirzaei et al., 2023; Wan et al., 2021).

Autonomous or semi-autonomous operation of UAVs is becoming critically important due to radio horizon limitations, possible communication interference in difficult terrain or urban environments, delays in data transmission over long distances, and the need for instant response to dynamic changes in the environment without operator intervention, especially in the presence of electromagnetic interference or electronic countermeasures. In addition, full autonomy eliminates the human factor and allows the operator's work to be automated (Elmokadem, & Savkin, 2021).

A key technology for solving the autonomy problem is object detection, which allows UAVs to identify and locate objects in their environment, understand the scene, and provide operational information (Cao et al., 2023; Jiao, & Abdullah, 2024; Nikouei et al., 2025). Modern detectors, such as those based on transformers, are often too resource-intensive, making them unsuitable for use on board UAVs in real time, as they have significant limitations in terms of battery consumption, payload, computing power, and memory capacity (Chen et al., 2021; Suo et al., 2024). At the same time, faster models, such as those based on EfficientNet (Tan et al., 2020), are not accurate enough for stable detection of small objects in dense scenes. Despite significant research in this area, there is no systematic comparative analysis of existing models based on key parameters, which complicates the selection of the optimal solution for specific UAV hardware constraints. Researchers and engineers are faced with the problem of navigating a vast "zoo" of models, each of which demonstrates improvements in certain metrics, but without a clear understanding of the overall picture of trade-offs between accuracy, speed, and model size.

The practical value of this study lies in creating the most comprehensive comparison of object detection models for UAVs to date, based on the VISDRONE2019-DET dataset (Du et al., 2019) and evaluated according to three key parameters: average accuracy (mAP), number of model parameters (M), and computational complexity (GFLOPs). Using Pareto analysis (Nia et al., 2022), the study allows us to objectively identify models that demonstrate the best balance between these parameters. Although GFLOPs is not traditionally used as a direct measure of detection speed, the study shows that for models with similar architectures, this indicator allows approximating the relative execution time, providing a theoretical basis for the initial selection of models without the need for testing on specific hardware.

The relevance of the study lies in establishing a starting point for both researchers working on new architectures and practitioners interested in directly implementing models on board UAVs. A systematic meta-analysis allows for quick navigation among the multitude of existing solutions and select models that demonstrate the best balance in their "weight class" for specific hardware constraints.

1. Literature review and problem statement

Standardized datasets play an important role in the objective evaluation and comparison of object detection models in aerial photography. One such comprehensive reference dataset for object detection from UAVs is VisDrone2019 (Du et al., 2019), developed by the AISKEYE team at Tianjin University. It consists of 10209 images containing a variety of scenes under different weather and lighting conditions. The key challenges of this dataset are significant variations in viewpoints and scales, a large number of small objects, scenes with high density and overlap, and a complex background (Wan et al., 2021). The complexity of the VisDrone2019 dataset has stimulated the active adaptation and development of various object detection paradigms. However, analysis of existing solutions reveals a number of unresolved issues related to the fundamental trade-off between accuracy and computational efficiency on board UAVs. Detection models can be divided into three categories: two-stage, single-stage, and transformer detectors.

Traditional two-stage detectors, such as Faster R-CNN (Ren et al., 2016) and Cascade R-CNN (Cai, & Vasconcelos, 2018), although demonstrating high accuracy on standard datasets, remain computationally expensive and slow. This significantly complicates their use on UAV edge devices for real-time computing (Song et al., 2024).

Single-stage detectors, in particular the YOLO family (Redmon et al., 2016), SSD (W. Liu et al., 2016), and RetinaNet (Lin et al., 2020), offer significantly better speed, making them a popular choice for real-time systems (Vaddi et al., 2021). However, the standard configurations of these models do not always effectively handle the specifics of aerial images. Anchor-free approaches such as CornerNet (Law, & Deng, 2020), CenterNet (Duan et al., 2019), FCOS (Keles et al., 2022), and YOLOX (Camarlinghi et al., 2024) partially solve the problem of detecting objects with variable aspect ratios, but as noted in (Du et al., 2019; Xiao et al., 2024), they can have difficulty localizing very small objects, which is a critical drawback for VisDrone. Thus, the question of reliable detection of small targets using fast single-stage models remains open.

A new paradigm of transformer detectors (DETR) (Kong et al., 2024; Liu et al., 2025; Zhang et al., 2025; Zhao et al., 2024) offers an end-to-end approach with a global attention mechanism that also eliminates the need for additional Non-Maximum Suppression (NMS) post-processing. Despite this, early DETR models had significant drawbacks: high computational cost and low performance in detecting small objects (Rekavandi et al., 2023). Although recent models such as RT-DETR (Zhao et al., 2024) and specialized versions UAV-DETR (Zhang et al., 2025) and VRF-DETR (Kong et al., 2024) have become more competitive, they do not solve the problem completely. For the most part, achieving high accuracy still requires significant computational resources, and their lighter versions are inferior in accuracy to specialized convolutional architectures. Thus, the question of creating a transformer architecture that simultaneously retains its advantages and achieves parity with the best CNN models in terms of accuracy/efficiency ratio on small objects remains unresolved. To overcome these challenges, researchers have proposed a number of strategies, but each has its limitations.

For example, multiscale feature fusion techniques - Bidirectional and Recursive Feature Pyramid Networks (Xiao et al., 2024), Hierarchical Extended Path Aggregation Networks (Chen, & Zhang, 2025), self-attention mechanism and specialized "heads" (Chen, & Zhang, 2025) improve the detection of small objects, but often at the cost of increasing the number of parameters and computational complexity, which contradicts the requirements for peripheral devices.

The use of lightweight architectures such as YOLO-tiny/n/s (Redmon et al., 2016), EfficientDet (Tan et al., 2020), EdgeYOLO (Liu et al., 2023), RT-DETR-R18 (Zhao et al., 2024), VRF-DETR (Wenbin, 2025) leads to a reduction in computational complexity. The same applies to the use of efficient convolution operations: depthwise separable (X. Tang et al., 2024), partial (Wang et al., 2025) and gated (Wenbin, 2025) convolutions, and spatial-channel separable downsampling (He, & Wang, 2023; Peng et al., 2018) decrease computation complexity and parameter count. However, simplifying the architecture often leads to the loss of important features necessary for identifying the smallest and most complex objects, which in turn affects the accuracy of the model.

To improve detection accuracy and training stability for small objects, it is recommended to use specialized loss functions such as Normalized Wasserstein Distance (Xu et al., 2022), Shape-IoU (Zhang, H., & Zhang, S., 2024), or Wise-IoU (Tong et al., 2023). They are less sensitive to localization errors of small objects, as these functions do not "penalize" minor shifts of a few pixels, which are critical for standard IoU metrics. The danger of this approach lies in the potential increase in computational complexity during training and the risk of model over-specialization. If the loss function is too focused on small objects, it can potentially impair the model's ability to generalize to medium and large objects.

Several approaches are used to address detection on high-resolution images. One of them is to slice images (Akyon, Altınuç, & Temizel, 2022) into smaller, often overlapping blocks, allowing standard models to find small objects. Another approach is super-resolution (Xiao et al., 2024), which attempts to artificially increase the quality of areas with small objects before detection. However, the disadvantage of these methods is a significant increase in processing time, as the detector is run multiple times, and a loss of global context, which can lead to errors at block boundaries or misclassification. QueryDet (Yang, Huang, & Wang, 2022) is an alternative that avoids dense computations on high-resolution feature maps. Although more efficient, this approach complicates the architecture, and its success strongly depends on the quality of the initial, coarse detection stage, where errors cannot be corrected later.

Analysis shows that achieving state-of-the-art results on VisDrone is an evolutionary process of adapting existing models (YOLO, DETR) by integrating specific modules. However, most publications focus on demonstrating the accuracy gains from their proposed modifications, often without conducting an in-depth analysis of the trade-off with computational costs in the context of real UAV hardware platforms. Although there is dynamic competition between CNN and transformer approaches in the industry, there is no single methodology that would allow objective identification of models that offer the optimal balance for different hardware budgets. Researchers and practitioners face the problem of choosing among dozens of models without a clear understanding of their practical suitability.

Thus, an important unresolved problem is the lack of a systematic analysis and identification of models that form the Pareto front in the multidimensional "accuracy-speed-size" space for the task of object detection in aerial images. Existing studies offer local improvements but do not provide a global picture that would allow engineers to make informed choices about the architecture that best suits the hardware constraints of a particular UAV and mission requirements.

Problem statement: It is necessary to identify which of the existing object detection models not only achieve high accuracy on complex aerial images, but are also computationally efficient enough for real-time use on resource-constrained UAVs. This requires a careful analysis of the "accuracy-speed-model size" trade-off and a search for optimal solutions on the Pareto front.

2. Research objectives

The aim of the study is to conduct a critical comparative meta-analysis of modern deep learning models for object detection tested on the VisDrone2019-DET dataset to identify architectures that form the Pareto frontier in the "accuracy-speed-size" space. This will enable us to formulate informed recommendations for engineers and researchers on the selection of optimal solutions for practical applications on UAV onboard computers, taking into account their hardware limitations. To achieve this goal, the following tasks were set:

1. Collect and compare data on accuracy (mAP), number of parameters (M), and computational complexity (FLOPs) for a wide range of models tested on VisDrone2019-DET.
2. Identify models that demonstrate the best balance between accuracy and computational cost (Pareto front) for different levels of resource requirements.
3. Show how GFLOPs correlate with execution speed in ms for different types of models using the example of Hailo 8L NPU

3. Materials and research methods

This section begins with a definition of the key elements of the research.

Research object: modern deep learning models for object detection, optimized for aerial monitoring tasks using unmanned aerial vehicles (UAVs) and tested on the VisDrone2019-DET benchmark dataset.

The main research hypothesis: among the variety of existing architectures, it is possible to identify a distinct set of Pareto-optimal models that offer the best compromise between accuracy, speed, and size. It is assumed that this frontier is predominantly formed by specialized rather than standard detectors, thanks to their targeted adaptation to the challenges of aerial imagery.

Assumptions made in this work: model performance data collected from various scientific publications are comparable because they were obtained on the standardized VisDrone2019-DET benchmark. It is assumed that this dataset adequately represents real-world aerial monitoring scenarios.

Simplifications made in this work: the multifactorial task of optimizing a model for UAVs (accuracy, execution speed, energy consumption, size) is simplified to the analysis of three key metrics: average accuracy (mAP), number of parameters, and computational complexity (FLOPs). It is important to note that although FLOPs is a common metric for computational complexity, the actual speed of the model, measured in frames per second (FPS) or milliseconds of execution (inference time), depends on many other factors (Fernandez et al., 2023). These include the specifics of the hardware platform (e.g., GPU, NPU), software optimization, memory bandwidth, and other implementation aspects. FLOPs is a theoretical, indirect metric, while the actual speed depends on the cost of memory access and platform characteristics (Jayanth, Gupta, & Prasanna, 2024; Ma et al., 2018). To investigate the relationship between theoretical complexity (GFLOPs) and practical runtime on representative edge hardware, we analyzed the correlation using performance data from the Hailo-8L NPU (Hailo-8L Model Zoo, 2025). The Hailo-8L is a specialized AI accelerator designed for edge devices, representative of hardware used in modern UAV companion computers.

The performance data (latency in milliseconds) was sourced directly from the official Hailo Model Zoo benchmarks. These benchmarks are conducted on the Hailo-8L M.2 PCIe module using the HailoRT (Hailo Runtime) environment. The analysis includes approximately 40 object detection models from the YOLO and DETR families, providing a statistically robust dataset for this investigation.

Crucially, the models benchmarked on the Hailo-8L utilize Post-Training Quantization (PTQ) to convert the model weights and activations from 32-bit floating-point (FP32) to 8-bit integer (INT8) format. This quantization is necessary for efficient execution on the NPU. While the meta-analysis (Tab. 1) reports accuracy in FP32 precision—as this is the standard metric reported in the original research papers and allows for fair architectural comparison—it is important to acknowledge that INT8 quantization typically results in a minor accuracy degradation (usually 1–2 % mAP) in practical deployment. The Pareto analysis in this study is based on FP32 accuracy and theoretical GFLOPs to identify the most efficient architectures. The Hailo-8L analysis (Fig. 3) then serves to validate whether this theoretical efficiency (GFLOPs) translates to practical execution speed (latency) on quantized models running on target hardware.

The research methodology was based on a comparative meta-analysis of the collected data. The following key metrics were selected to evaluate the "speed-accuracy" trade-off:

- **Mean Average Precision (mAP):** The mAP@0.50 indicator was mainly used (detection is considered correct if the intersection over union (IOU) between the predicted and reference frames exceeds 0.50). Notably, mAP@0.50 is preferable to the more stringent mAP@0.5:0.95 (which averages precision across IoU thresholds from 0.50 to 0.95 in increments of 0.05) when evaluating small objects, as IoU-based assessments tend to underperform for diminutive targets due to their sensitivity to minor pixel-level discrepancies.
- **Number of parameters (Parameters, M):** This metric is measured in millions (M) and indicates the size of the model, its complexity, and, accordingly, the memory requirements for storing and running the model.
- **Number of floating point operations (FLOPs, G):** Measured in billions and quantifies the computational cost of a single forward pass of the model (inference). This metric is often used to estimate theoretical computational efficiency.

Performance data (mAP, Parameters, FLOPs) were primarily extracted from the original research papers. In cases where the number of parameters or FLOPs was not explicitly reported (marked with "*" in Tab. 1), they were estimated using a standardized methodology. The estimation relied on architectural similarity and theoretical calculation/profiling. If the model is a direct modification of a known baseline (e.g., YOLOv5s with an added module), the baseline's values were used as a starting point, and the cost of the added modules was estimated based on their description. We also reconstructed the model architecture based on the description provided in the paper, utilizing the official source code if available. The theoretical

FLOPs and parameter counts were then calculated using standard PyTorch profiling tools, such as torchinfo or the thop (PyTorch-OpCounter) library. For example, the estimations for the EDNet series (Nos. 5, 7, 9, 11, 30, 47, 48) were performed by profiling the architecture based on the details provided in (Song et al., 2024). All calculations assumed the input image resolution specified in the respective paper; if the resolution was not specified, a standard benchmark resolution (e.g., 640 x 640) was used for the estimation.

Pareto front analysis was used to visualize and analyze the trade-off between these metrics. The model A is considered Pareto-optimal in the set of models S if there is no other model $B \in S$ that is simultaneously inferior to A in all criteria and superior to it in at least one criterion. That is, A is Pareto optimal if $\nexists B \in S : [\forall i, f_i(B) \geq f_i(A)] \wedge [\exists j, f_j(B) > f_j(A)]$, where f_j are criterion functions (mAP, number of parameters, GFLOPs) that need to be maximized or minimized depending on the task. The models that form the Pareto front represent a set of optimal solutions for which it is impossible to improve one indicator (e.g., mAP) without worsening another (e.g., increasing the number of parameters or FLOPs).

4. Research results

We start by benchmarking the surveyed models on VisDrone2019 across three axes – mAP@0.5, parameter count, and GFLOPs-and we summarize the results in Table 1. The literature analysis revealed a number of architectures that were specifically designed or adapted with a focus on real-time performance and computational efficiency for deployment on UAVs. To quantitatively assess the accuracy-efficiency trade-off, the performance data of various models on the VisDrone2019 dataset was analyzed. Table 1 provides summary data for representative models, including those at the Pareto frontier and well-known baseline models.

Table 1

Comparison of the performance of selected object detection models on VisDrone2019-DET.

No	Object detection model	Family	mAP@.50 (%)	Params (M)	FLOPs (G)	Pareto optimal categories
1	Drone-DETR (Kong et al., 2024)	DETR	53.9	28.7	128.3	M, FLOPs
2	VRF-DETR (Wenbin, 2025)	DETR	51.4	13.5	44.3	M, FLOPs
3	UAV-DETR-R50 (Zhang et al., 2025)	DETR	51.1	42.0	170	-
4	RT-DETR-R50 (Wenbin, 2025; Zhao et al., 2024)	DETR	50.7	41.8	133.2	-
5	EDNet-X (Song et al., 2024)	YOLOv10	50.2	48.7	270.4*	-
6	DBYOLOv8s (Tan et al., 2025)	YOLOv8s	49.3	11	119.8	M
7	EDNet-L (Song et al., 2024)	YOLOv10l	49.0	31.7	179.2*	-
8	Modified YOLOv8l (Wang et al., 2025)	YOLOv8l	48.3	31.2	490.9	-
9	EDNet-B (Song et al., 2024)	YOLOv10	48.3	25.5	141.1*	-
10	RT-DETR-R18 (Wenbin, 2025; Zhao et al., 2024)	DETR	47.2	20.2	57.0	-
11	EDNet-M (Song et al., 2024)	YOLOv10	47.1	19.1	88.4*	-
12	SL-YOLO (Chen, & Zhang, 2025)	YOLOv8s	46.9	9.6	36.7*	M, FLOPs
13	YOLOv9e (Song et al., 2024; Wang et al., 2025)	YOLOv9	46.5	57.4	189.0	-
14	LSOD-YOLOv8s (Li, & Wu, 2024)	YOLOv8n	45.2	2.7	23.5	-
15	MSGD-YOLO (Xiao et al., 2024)	YOLOv8n	45.2	2.6	21.5	M, FLOPs
16	YOLOv10x (Song et al., 2024; Hussain, 2024)	YOLOv10	44.8	31.6	160.4	-
17	YOLOv5x (Song et al., 2024; Hussain, 2024)	YOLOv5	44.7	97.2	205.7	-
18	YOLOv8x (Song et al., 2024; Hussain, 2024)	YOLOv8	44.7	68.1	257.8	-
19	YOLOv5s VIOU+PFFN+ADhead (Lu et al., 2023)	YOLOv5	44.6	19.26	112.6*	-
20	HIC-YOLOv5 (Tang et al., 2024; Wenbin, 2025)	YOLOv5	44.3	10.5	31.2	-
21	YOLOv10l (Song et al., 2024; Hussain, 2024)	YOLOv10	44.3	25.7	120.3	-
22	YOLOv8l (Song et al., 2024; Hussain, 2024)	YOLOv8	43.6	43.6	165.2	-
23	YOLOv10-B (Hussain, 2024)	YOLOv10	43.4	20.4	92	-
24	YOLOv5s VIOU+PFFN (Lu et al., 2023)	YOLOv5	43.2	7.4	39.6*	-
25	YOLOv9m (Song et al., 2024; Wang et al., 2025)	YOLOv9	43.1	20.0	76.3	-
26	YOLOv9-C (Wang et al., 2025)	YOLOv9	43.1	25.3	102.1	-
27	YOLOv12-M (Wenbin, 2025)	YOLOv12	43.1	20.2	67.5	-
28	CPAM-YOLO (Wu, Zhu, & Cao, 2024)	YOLOv5l	43.0	27.5	106.4	-
29	YOLOv5l (Song et al., 2024; Hussain, 2024)	YOLOv5	42.9	53.1	109.1	-
30	EDNet-S (Song et al., 2024)	YOLOv10	42.5	9.3	37.1*	-
31	Modified YOLOv8s (Li et al., 2023)	YOLOv8	42.2	9.6	22*	-
32	YOLOv8m (Song et al., 2024; Hussain, 2024)	YOLOv8	41.9	25.9	78.9	-
33	DM-YOLOX (Li et al., 2024)	YOLOX	41.9	9.6	27.9	-
34	LV-YOLOv5 (Wang et al., 2022; Wang et al., 2025)	YOLOv5	41.7	36.6	38.8	-
35	YOLOv10m (Song et al., 2024; Hussain, 2024)	YOLOv10	41.5	16.5	59.1	-
36	PDWT-YOLO (Zhang et al., 2023)	YOLOv7-tiny	41.2	6.4	24.2	-
37	YOLOv5m (Song et al., 2024; Hussain, 2024)	YOLOv5	41.1	21.2*	49	-
38	UN-YOLOv5s (Guo et al., 2023)	YOLOv5	40.5	9.1*	37.4	-
39	LAI-YOLOv5s (Deng et al., 2023)	YOLOv5	40.4	6.3	29.0	-
40	YOLOv9s (Song et al., 2024; Wang et al., 2025)	YOLOv9	39.0	7.2	26.4	-
41	YOLOv8s (Jiao, & Abdullah, 2024)	YOLOv8	38.9	11.1	28.5	-
42	YOLOv5s (Wang et al., 2025)	YOLOv5	38.5	9.1	23.8	-
43	YOLOv10s (Song et al., 2024; Hussain, 2024)	YOLOv10	38.5	8.0	21.6	-
44	C3TB-YOLOv5 (Wu et al., 2024)	YOLOv5	38.3	8.0	19.7	FLOPs
45	MPE-YOLO (Su et al., 2024)	YOLOv8	37.0	4.4	11.9*	FLOPs
46	YOLOv7-tiny (Wang et al., 2025)	YOLOv7	33.9	6.0	13.3	-

47	EDNet-N (Song et al., 2024)	YOLOv10	33.8	2.9	15.2*	-
48	EDNet-Tiny (Song et al., 2024)	YOLOv10	33.3	1.8	14.2*	M
49	YOLOv8n (Wang et al., 2025)	YOLOv8	33.2	3.0	8.1	FLOPs
50	Faster R-CNN (ResNet50-FPN) (Kong et al., 2024)	R-CNN	33.2	41.2	207.1	-
51	DCE-YOLOv8n (An et al., 2024)	YOLOv8	32.7	4.82*	13*	-
52	YOLOv10n (Song et al., 2024; Hussain, 2024)	YOLOv10	32.3	2.7	6.7	FLOPs
53	YOLOv5n (Wang et al., 2025)	YOLOv5	32.2	2.5	7.1	-
54	YOLOv6s (Wang et al., 2025)	YOLOv6	29.1	16.3	44.2	-
55	CenterNet-ResNet-50 (Lu et al., 2023)	CenterNet	29.0	32*	120*	-
56	RetinaNet-ResNet-50-FPN (Kong et al., 2024)	RetinaNet	27.3	19.8	93.7	-
57	SSD-ResNet-50 (Kong et al., 2024)	SSD	23.9	24.5	87.9	-
58	YOLOv3-tiny (Wang et al., 2025)	YOLOv3	23.6	12.1	18.9	-
59	EfficientDet-D1 (Lu et al., 2023)	EfficientDet	21.2	6.6*	6.1*	FLOPs

Values marked with "*" were not directly reported in the original works, but were approximated by the authors using the methodology detailed in Section 3. In the "Pareto optimal" column, "M" denotes Pareto optimal models by the number of parameters, and FLOPs by the number of operations.

Building on these benchmarks, we apply Pareto analysis to reveal the best accuracy-efficiency trade-offs. The scatter plots (Fig. 1: parameters vs. mAP@0.5; Fig. 2: GFLOPs vs. mAP@0.5) make the Pareto front clear along the upper-left edge. The application of Pareto analysis in this task allows not only to systematize existing architectures, but also to objectively evaluate their effectiveness in a multidimensional space of criteria. The models that form the Pareto front are benchmarks in terms of the balance between accuracy, speed, and size, and they should be considered a priority for practical implementation. This approach eliminates subjective model selection based on a single criterion and promotes transparency and reproducibility of research in the field. This approach ensures that all models on the Pareto frontier are the best possible compromises for the given constraints and requirements. The results of the meta-analysis are presented in summary tables and graphs, which are a summary visualization demonstrating the best models for practical application on UAVs, where each of these models represents the optimal trade-off between accuracy and efficiency for different hardware constraints.

It is noteworthy that models such as MSGD-YOLO, SL-YOLO, VRF-DETR, and Drone-DETR appear on both sides of the Pareto front, indicating their exceptional balance in terms of both size and computational complexity relative to the achieved accuracy.

To connect these trade-offs with real device behavior, we relate GFLOPs to latency on a representative UAV companion computer – the Hailo-8L NPU, using YOLO/DETR models as a case study (Fig. 3). FLOPs are a useful indicator for comparing architectures, especially within a single model family or on a single hardware platform, but they are not an absolute measure of real-world performance, and testing on target hardware remains necessary. In Fig. 3, a bubble chart is shown demonstrating models for object detection of the YOLO and DETR families on Hailo 8l for another dataset popular in object detection – COCO (Hailo-8l Model Zoo, 2025). The horizontal axis shows the computational cost in gigaflops per image (GFLOPs, log₁₀ scale), and the vertical axis shows the processing delay of a single image in milliseconds (log₁₀ scale). The area of the bubble is proportional to the number of trained parameters (Params M), which forms the third dimension and reflects the memory requirement. The purple subset, which corresponds to YOLOv11 n/s/m/l/x, is also a Pareto optimal model according to (Hailo-8l Model Zoo, 2025).

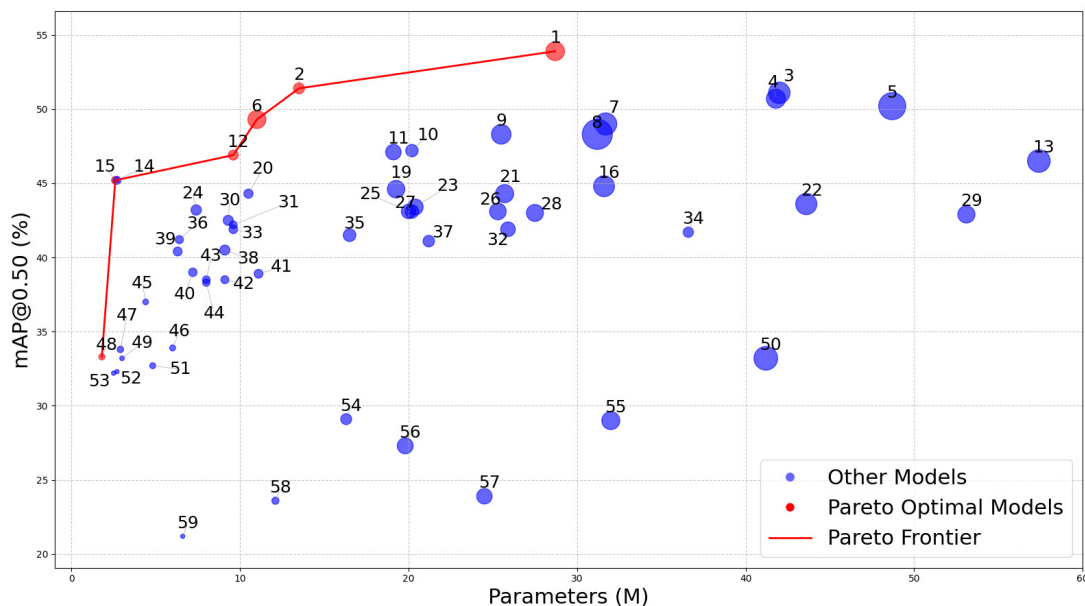


Fig. 1. Number of parameters (M) vsmAP@0.5. Pareto models are shown in red. The size of the bubble is proportional to GFLOPs, the number corresponds to the column "No." in Tab. 1

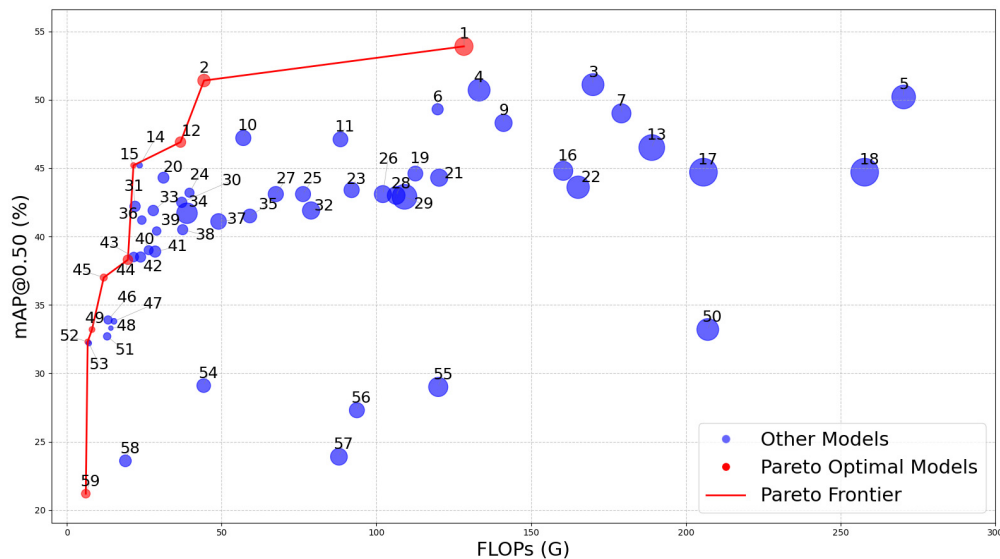


Fig. 2. Flops (G) vs mAP@0.5. Pareto models are shown in red.

The size of the bubble is proportional to the number of parameters, the number corresponds to the column "№" in Tab. 1

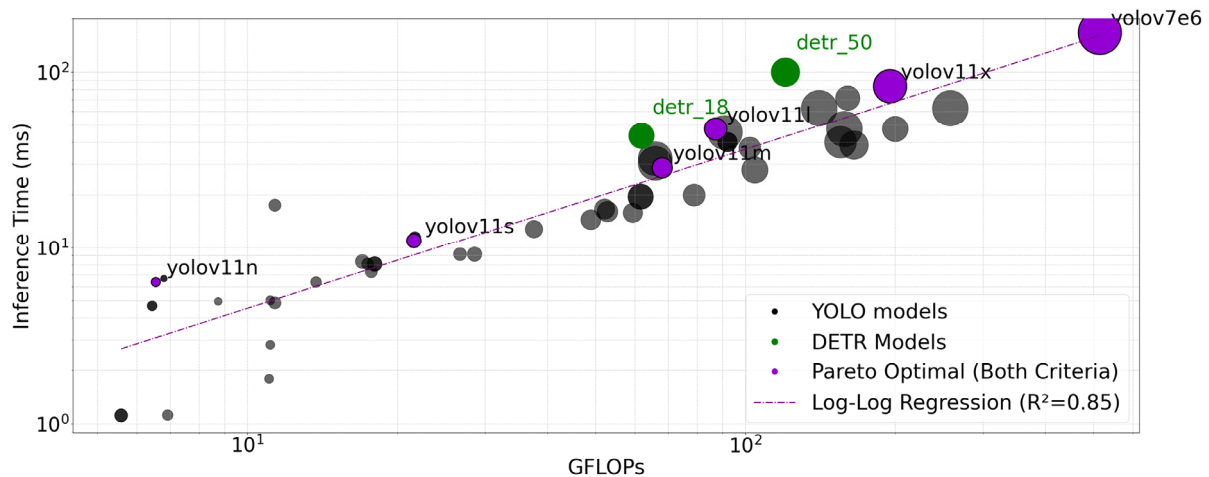


Fig. 3. Dependence of model speed on GFLOPs using the example of Hailo 8L NPU

Log-log orthogonal regression (purple dashed line) summarizes the global trend of the dependence of "GFLOPs on execution time". The analysis is based on approximately 40 data points spanning various YOLO and DETR architectures. The coefficient of determination $R^2 \approx 0.85$ globally, $R^2 \approx 0.92$ for the YOLO family, and an exceptionally high $R^2 \approx 0.98$ for the YOLOv11 subfamily. This strong correlation indicates a clear power-law relationship between inference time and GFLOPs on the target hardware, suggesting the data is statistically robust for this specific accelerator.

This high correlation confirms that GFLOPs can serve as a powerful proxy for runtime on the Hailo-8L, likely due to the high degree of optimization achieved by the Hailo Dataflow Compiler, which minimizes software overheads. However, it is crucial to discuss the limitations of this correlation. While GFLOPs measure the number of arithmetic operations, actual latency is also significantly influenced by Memory Access Cost (MAC), data movement, operator parallelism, and the efficiency of specific kernel implementations on the NPU (Ma et al., 2018).

The slightly lower global correlation ($R^2 \approx 0.85$) compared to intra-family correlation (up to 0.98) highlights architectural differences. For instance, DETR models might exhibit different latency characteristics compared to GFLOPs-equivalent YOLO models due to the differing computational nature and memory access patterns of attention mechanisms versus convolutions. Therefore, while GFLOPs provide a robust heuristic for pre-selecting models, the correlation might weaken when comparing architectures with vastly different operational profiles or when applied to different hardware platforms.

Discussion and conclusions

The results showed that among modern detectors for VisDrone2019, there is a wide range of trade-offs between accuracy and computational complexity. Table 1 shows how the average accuracy of mAP@0.50 varies from 33% for the simplest models to 54% for the best ones, while the number of parameters ranges from 2 million to 97 million and the complexity ranges from 12 to 490 GFLOPs. Thus, higher accuracy is usually achieved by models with higher resource costs, but numerous exceptions indicate significant progress in architectures. For example, the specialized transformer model Drone-DETR achieved the highest mAP = 53.9 % at 128 GFLOPs, while a typical two-stage detector achieved only 48 % at nearly 491 GFLOPs. Similarly, the improved single-stage SL-YOLO model achieved 46.9 % mAP at 36.7 GFLOPs, exceeding the accuracy of the standard large YOLOv5x (44.7 % mAP) with approximately 6 times fewer computations. Thus, through

targeted optimization for UAV applications, some models provide significantly better accuracy per unit of computation than classical detectors. This is particularly important for UAV onboard computers, where computational resources are limited.

To formalize the "accuracy-efficiency" balance, Pareto fronts were constructed based on parameters (Fig. 1) and GFLOPs (Fig. 2).

Pareto analysis objectively identified a limited set of the best solutions. In particular, the Pareto frontier mainly consists of improved versions of YOLO and DETR, specially adapted for small objects. These include MSGD-YOLO, SL-YOLO, VRF-DETR, and Drone-DETR, which are present in both diagrams (by parameters and FLOPs). This indicates their exceptional balance between model size and computational complexity while maintaining high accuracy. Other models on these fronts, such as DB-YOLOv8s, EfficientDet-D1, MPE-YOLO, and C3TB-YOLOv5, provide optimal compromises in their complexity ranges. Importantly, none of the traditional approaches, except for YOLOv8n and YOLOv10n, made it into the Pareto frontier, and none did so on both criteria simultaneously. They either lose in accuracy at equal cost or require more resources to achieve similar accuracy compared to the selected Pareto models. For example, the baseline YOLOv8x achieved 44.7 % mAP at 257.8 GFLOPs, while the much lighter MSGD-YOLO achieved 45.2 % at only 21.5 GFLOPs.

The dominance of YOLO and DETR variants on the Pareto front is consistent with general trends in object detection, but their success in the UAV domain (specifically VisDrone) warrants architectural discussion. The efficiency of YOLO architectures stems primarily from their single-stage design, which avoids the computational overhead of region proposal networks found in two-stage detectors. Furthermore, modern YOLO variants utilize efficient backbones and optimized neck/head structures that effectively balance feature fusion across scales – critical for detecting the small, densely packed objects common in aerial imagery. In contrast, DETR-based models leverage the global context provided by self-attention mechanisms. This allows them to better model relationships between objects and the complex backgrounds typical of VisDrone, potentially reducing false positives. While early DETR models struggled with small objects and high computational costs, specialized variants like Drone-DETR and VRF-DETR incorporate mechanisms (e.g., enhanced high-resolution features or specialized queries) that specifically address these limitations, making them competitive and often superior in this specific domain.

Thus, the problem of selecting the optimal model for a hardware-constrained UAV is actually solved by determining this Pareto set. The identified models achieve mAP@0.50 ~50 % with parameter sizes ≤ 15 M and costs ≤ 60 GFLOPs, which is consistent with the typical capabilities of onboard systems. This means that UAVs with different resources should use different models to achieve the best performance.

Results show how well the theoretical computational complexity of a model (GFLOPs) describes its actual performance on the target accelerator. Research using the Hailo 8L NPU confirmed a strong correlation between GFLOPs and inference time: models that perform fewer operations tend to run faster. In Fig. 3 for a set of YOLO and DETR family models, there is a near-exponential dependence between GFLOPs and execution speed, with a coefficient of determination $R^2 \approx 0.85$ globally (for all models) and up to 0.92–0.98 within a single architecture. In other words, in practice, reducing GFLOPs results in an almost proportional reduction in processing time, especially when comparing related models. This is an important result because it confirms the validity of using GFLOPs as a proxy for model speed in the early stages of selection. However, it should be noted that the correlation is not absolute: for example, DETR models may have slightly higher latency than GFLOPs-equivalent YOLO models due to differences in architecture, memory access, and optimization. Therefore, GFLOPs should only be interpreted as a rough guide, and final performance verification should be performed on the target hardware. Despite this caveat, the results obtained demonstrate that for models from the same family (e.g., different versions of YOLO), GFLOPs are almost directly proportional to inference time. Thus, computer vision system engineers for UAVs can confidently use GFLOPs as a preliminary selection metric for models capable of meeting real-time requirements, narrowing down the pool of candidates before detailed testing on hardware.

Data on the performance of over 60 object detection models on the VisDrone2019-DET dataset has been collected and systematized, creating the most comprehensive consolidated database for this task to date. The analysis revealed a wide range of characteristics: from lightweight models with 1.8M parameters (EDNet-Tiny) to heavy ones with 97.2M (YOLOv5x), with an mAP@0.50 accuracy ranging from 21.2 % to 53.9 %. This systematization allows you to quickly navigate the "zoo" of models and make objective comparisons.

Two groups of Pareto-optimal models have been identified, forming the front of the best compromises. Based on the parameter minimization criterion, the optimal models are EDNet-Tiny, MSGD-YOLO, SL-YOLO, DBYOLOv8s, VRF-DETR, and Drone-DETR, which achieve mAP@0.50 ranging from 33.3 % to 53.9 % with 1.8–28.7M parameters. Based on the criterion of minimizing GFLOPs, the optimal models are EfficientDet-D1, YOLOv10n, YOLOv8n, MPE-YOLO, C3TB-YOLOv5, MSGD-YOLO, SL-YOLO, VRF-DETR, and Drone-DETR with a range of 6.1–128.3 GFLOPs. It is noteworthy that four models (MSGD-YOLO, SL-YOLO, VRF-DETR, Drone-DETR) are present on both fronts, which indicates their exceptional balance.

The high correlation between the theoretical GFLOPs metric and the practical execution speed has been experimentally confirmed using the Hailo 8L NPU as an example. The coefficient of determination $R^2 \approx 0.85$ for all models and $R^2 \approx 0.92$ –0.98 for models of the same family indicates an almost exponential dependence, which validates the use of GFLOPs as the primary model selection criterion. This simplifies the process of selecting an architecture for specific UAV hardware constraints without the need for full-scale testing on the target platform.

Data availability: For a more detailed and interactive study of the data, allowing dynamic filtering and comparison of models, a web page has been created, available at: https://smolinyaroslav.github.io/visdrone_literature_review

Authors' contribution: Yaroslav Smolin – conceptualization, methodology, formal analysis, software, validation, writing – original draft. Viacheslav Liskin – supervision, guidance on research design and interpretation of results, writing – review & editing.

Sources of funding. This study did not receive any grant from a funding institution in the public, commercial, or non-commercial sectors.

References

- Akyon, F. C., Altınuç, S. O., & Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. In Y. Berthoumieu & P. Frossard (Eds.), *2022 IEEE International Conference on Image Processing (ICIP)* (pp. 966–970). IEEE. <https://doi.org/10.1109/ICIP46576.2022.9897990>
- An, J., Lee, D., Putro, M. D., & Kim, B. (2024). DCE-YOLOv8: Lightweight and Accurate Object Detection for Drone Vision. *IEEE Access*, 12, 170898–170912. <https://doi.org/10.1109/ACCESS.2024.3481410>
- Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving Into High Quality Object Detection. In D. Forsyth, I. Laptev, D. Ramanan, & A. Oliva (Eds.), *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6154–6162). IEEE. <https://doi.org/10.1109/CVPR.2018.00644>
- Camarlinghi, N., Michelozzi, B., Martino, G., Di Tommaso, A., Fontanelli, G., & Masini, A. (2024). Drone-based monitoring on the edge using a high-resolution payload. In J. Dijk & J. L. Sanchez-Lopez (Eds.), *Autonomous Systems for Security and Defence* (p. 1–7). SPIE. <https://doi.org/10.1117/12.3031453>
- Cao, Z., Kooistra, L., Wang, W., Guo, L., & Valente, J. (2023). Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review. *Drones*, 7(10), 620. <https://doi.org/10.3390/drones7100620>
- Chen, D., & Zhang, L. (2025). SL-YOLO: A Stronger and Lighter Drone Target Detection Model. arXiv. <https://doi.org/10.48550/arXiv.2411.11477>
- Chen, L., Hu, J., Li, X., Quan, F., & Chen, H. (2021). Onboard Real-time Object Detection for UAV with Embedded NPU. In X. Ning & Y. Haibin (Eds.), *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 192–197). IEEE. <https://doi.org/10.1109/CYBER53097.2021.9588193>
- Deng, L., Bi, L., Li, H., Chen, H., Duan, X., Lou, H., Zhang, H., Bi, J., & Liu, H. (2023). Lightweight aerial image object detection algorithm based on improved YOLOv5s. *Scientific Reports*, 13(1), 7817. <https://doi.org/10.1038/s41598-023-34892-4>
- Du, B., Liao, Z., Zhang, Y., Cai, Z., Chen, J., & Huang, D. (2024). Centerness-based Instance-aware Knowledge Distillation with Task-wise Mutual Lifting for Object Detection on Drone Imagery. arXiv. <https://doi.org/10.48550/arXiv.2411.02861>
- Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., Bo, L., Shi, H., Zhu, R., Kumar, A., Li, A., Zinollayev, A., Askergaliyev, A., Schumann, A., Mao, B., Liu, Z. (2019). VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 213–226. <https://doi.org/10.1109/ICCVW.2019.00030>
- Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Keypoint Triplets for Object Detection. In I. S. Kweon, N. Paragios, M.-H. Yang, & S. Lazebnik (Eds.), *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 6568–6577). <https://doi.org/10.1109/ICCV.2019.00667>
- Elmokadem, T., & Savkin, A. V. (2021). Towards Fully Autonomous UAVs: A Survey. *Sensors*, 21(18), 6223. <https://doi.org/10.3390/s21186223>
- Fernandez, J., Kahn, J., Na, C., Bisk, Y., & Strubell, E. (2023). The Framework Tax: Disparities Between Inference Efficiency in NLP Research and Deployment. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 1588–1600). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.emnlp-main.98>
- Guo, J., Liu, X., Bi, L., Liu, H., & Lou, H. (2023). UN-YOLOv5s: A UAV-Based Aerial Photography Detection Algorithm. *Sensors*, 23(13), 5907. <https://doi.org/10.3390/s23135907>
- Gutierrez, G., Llerena, J. P., Usero, L., & Patricio, M. A. (2025). A Comparative Study of Convolutional Neural Network and Transformer Architectures for Drone Detection in Thermal Images. *Applied Sciences*, 15(1), 109. <https://doi.org/10.3390/app15010109>
- Hailo-8l model zoo. (2025). https://github.com/hailo-ai/hailo_model_zoo/blob/master/docs/public_models/HAILO8L_object_detection.rst
- He, L., & Wang, M. (2023). SliceSamp: A Promising Downsampling Alternative for Retaining Information in a Neural Network. *Applied Sciences*, 13(21), 11657. <https://doi.org/10.3390/app132111657>
- Hua, W., & Chen, Q. (2025). A survey of small object detection based on deep learning in aerial images. *Artificial Intelligence Review*, 58(6), 162. <https://doi.org/10.1007/s10462-025-11150-9>
- Hussain, M. (2024). YOLOv5, YOLOv8, and YOLOv10: The Go-To Detectors for Real-time Vision. arXiv. <https://doi.org/10.48550/arXiv.2407.02988>
- Jayanth, R., Gupta, N., & Prasanna, V. (2024). Benchmarking Edge AI Platforms for High-Performance ML Inference. In E. Mortensen (Ed.), *2024 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1–7). <https://doi.org/10.1109/HPEC62836.2024.10938499>
- Jiao, L., & Abdullah, M. I. (2024). YOLO series algorithms in object detection of unmanned aerial vehicles: A survey. *Service Oriented Computing and Applications*, 18(3), 269–298. <https://doi.org/10.1007/s11761-024-00388-w>
- Keles, M. C., Salmanoğlu, B., Guzel, M. S., Gursoy, B., & Bostanci, G. E. (2022). Evaluation of YOLO Models with Sliced Inference for Small Object Detection. arXiv. <https://doi.org/10.48550/arXiv.2203.04799>
- Kong, Y., Shang, X., & Jia, S. (2024). Drone-DETR: Efficient Small Object Detection for Remote Sensing Image Using Enhanced RT-DETR Model. *Sensors*, 24(17), 5496. <https://doi.org/10.3390/s24175496>
- Law, H., & Deng, J. (2020). CornerNet: Detecting Objects as Paired Keypoints. *International Journal of Computer Vision*, 128(3), 642–656. <https://doi.org/10.1007/s11263-019-01204-1>
- Li, H., & Wu, J. (2024). LSOD-YOLOv8s: A Lightweight Small Object Detection Model Based on YOLOv8 for UAV Aerial Images. *Engineering Letters*, 32(11), 2073–2082.
- Li, X., Wang, F., Wang, W., Han, Y., & Zhang, J. (2024). DM-YOLOX aerial object detection method with intensive attention mechanism. *The Journal of Supercomputing*, 80(9), 1–23. <https://doi.org/10.1007/s11227-024-05944-x>
- Li, Y., Fan, Q., Huang, H., Han, Z., & Gu, Q. (2023). A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones*, 7(5), 304. <https://doi.org/10.3390/drones7050304>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2020). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- Liu, S., Zha, J., Sun, J., Li, Z., & Wang, G. (2023). EdgeYOLO: An Edge-Real-Time Object Detector. In J. Han & J. Sun (Eds.), *2023 42nd Chinese Control Conference (CCC)* (pp. 7507–7512). IEEE. <https://doi.org/10.23919/CCC58697.2023.10239786>
- Liu, W., Angelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2
- Liu, Y., He, M., & Hui, B. (2025). ESO-DETR: An Improved Real-Time Detection Transformer Model for Enhanced Small Object Detection in UAV Imagery. *Drones*, 9(2), 143. <https://doi.org/10.3390/drones9020143>
- Lu, S., Lu, H., Dong, J., & Wu, S. (2023). Object Detection for UAV Aerial Scenarios Based on Vectorized IOU. *Sensors*, 23(6), 3061. <https://doi.org/10.3390/s23063061>

- Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 122–138). Springer International Publishing. https://doi.org/10.1007/978-3-030-01264-9_8
- Mirzaei, B., Nezamabadi-pour, H., Raoof, A., & Derakhshani, R. (2023). Small Object Detection and Tracking: A Comprehensive Review. *Sensors*, 23(15), 6887. <https://doi.org/10.3390/s23156887>
- Nia, V. P., Ghaffari, A., Zolnouri, M., & Savaria, Y. (2022). *Rethinking Pareto Frontier for Performance Evaluation of Deep Neural Networks*. arXiv. <https://doi.org/10.48550/arXiv.2202.09275>
- Nikouei, M., Baroutian, B., Nabavi, S., Taraghi, F., Aghaei, A., Sajedi, A., & Moghaddam, M. E. (2025). *Small Object Detection: A Comprehensive Survey on Challenges, Techniques and Real-World Applications*. arXiv. <https://doi.org/10.48550/arXiv.2503.20516>
- Peng, J., Xie, L., Zhang, Z., Tan, T., & Wang, J. (2018). *Accelerating Deep Neural Networks with Spatial Bottleneck Modules*. arXiv. <https://doi.org/10.48550/arXiv.1809.02601>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In J. A. Schnabel & K. Mori (Eds.), *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). IEEE. <https://doi.org/10.1109/CVPR.2016.91>
- Rekavandi, A. M., Rashidi, S., Boussaid, F., Hoefs, S., Akbas, E., & Bennamoun, M. (2023). *Transformers in Small Object Detection: A Benchmark and Survey of State-of-the-Art*. arXiv. <https://doi.org/10.48550/arXiv.2309.04902>
- Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, & M. Sugiyama (Eds.), *Advances in Neural Information Processing Systems* (pp. 91–99). Curran Associates, Inc. <https://doi.org/10.48550/arXiv.1506.01497>
- Song, Z., Zhang, Y., & Ebayyeh, A. A. R. M. A. (2024). EDNet: Edge-Optimized Small Target Detection in UAV Imagery – Faster Context Attention, Better Feature Fusion, and Hardware Acceleration. In H. Liu & K. Qian (Eds.), *2024 IEEE Smart World Congress (SWC)* (pp. 829–838). <https://doi.org/10.1109/SWC62898.2024.00141>
- Su, J., Qin, Y., Jia, Z., & Liang, B. (2024). MPE-YOLO: Enhanced small target detection in aerial imaging. *Scientific Reports*, 14(1), 17799. <https://doi.org/10.1038/s41598-024-68934-2>
- Suo, J., Zhang, X., Shi, W., & Zhou, W. (2024). E\$^3\$-UAV: An Edge-based Energy-Efficient Object Detection System for Unmanned Aerial Vehicles. *IEEE Internet of Things Journal*, 11(3), 4398–4413. <https://doi.org/10.1109/IJOT.2023.3301623>
- Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. In G. Mori, K. Saenko, & S. Savarese (Eds.), *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10778–10787). <https://doi.org/10.1109/CVPR42600.2020.01079>
- Tan, Y., Xu, B., Sun, J., Xu, C., Pan, W., Dai, S., & Liu, H. (2025). DBYOLOv8: Dual-Branch YOLOv8 Network for Small Object Detection on Drone Image. *International Journal of Advanced Computer Science and Applications*, 16(1), 1301–1309. <https://doi.org/DOI.10.14569/IJACSA.2025.01601124>
- Tang, S., Zhang, S., & Fang, Y. (2024). HIC-YOLOv5: Improved YOLOv5 For Small Object Detection. In Y. Hasegawa (Ed.), *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6614–6619). <https://doi.org/10.1109/ICRA57147.2024.10610273>
- Tang, X., Chen, X., Cheng, J., Wu, J., Fan, R., Zhang, C., & Zhou, Z. (2024). YOLO-Ant: A Lightweight Detector via Depthwise Separable Convolutional and Large Kernel Design for Antenna Interference Source Detection. *IEEE Transactions on Instrumentation and Measurement*, 73, 3503817. <https://doi.org/10.1109/TIM.2024.3379397>
- Tong, Z., Chen, Y., Xu, Z., & Yu, R. (2023). Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv. <https://doi.org/10.48550/arXiv.2301.10051>
- Vaddi, S., Kim, D., Kumar, C., Shad, S., & Jannesari, A. (2021). Efficient Object Detection Model for Real-time UAV Application. *Computer and Information Science*, 14(1), 45–55. <https://doi.org/10.5539/cis.v14n1p45>
- Wan, J., Zhang, B., Zhao, Y., Du, Y., & Tong, Z. (2021). VistrongerDet: Stronger Visual Information for Object Detection in VisDrone Images. In L. O’Conner (Ed.), *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (pp. 2820–2829). IEEE. <https://doi.org/10.1109/ICCVW54120.2021.00316>
- Wang, C.-Y., Yeh, I.-H., & Mark Liao, H.-Y. (2025). YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, & G. Varol (Eds.), *Computer Vision – ECCV 2024* (pp. 1–21). Springer. https://doi.org/10.1007/978-3-031-72751-1_1
- Wang, H., Liu, J., Zhao, J., Zhang, J., & Zhao, D. (2025). Precision and speed: LSOD-YOLO for lightweight small object detection. *Expert Systems with Applications*, 269, 126440. <https://doi.org/10.1016/j.eswa.2025.126440>
- Wang, J., Liu, W., Zhang, W., & Liu, B. (2022). LV-YOLOv5: A light-weight object detector of Vit on Drone-captured Scenarios. In Z. Li (Ed.), *2022 16th IEEE International Conference on Signal Processing (ICSP)* (pp. 178–183). IEEE. <https://doi.org/10.1109/ICSP56322.2022.9965217>
- Wang, X., Peng, Y., & Shen, C. (2025). *Efficient Feature Fusion for UAV Object Detection*. arXiv. <https://doi.org/10.48550/arXiv.2501.17983>
- Wenbin, L. (2025). *An Efficient Aerial Image Detection with Variable Receptive Fields*. arXiv. <https://doi.org/10.48550/arXiv.2504.15165>
- Wu, H., Zhu, Y., & Cao, M. (2024). An algorithm for detecting dense small objects in aerial photography based on coordinate position attention module. *IET Image Processing*, 18(7), 1759–1767. <https://doi.org/10.1049/ipr2.13061>
- Wu, Q., Li, Y., Huang, W., Chen, Q., & Wu, Y. (2024). C3TB-YOLOv5: Integrated YOLOv5 with transformer for object detection in high-resolution remote sensing images. *International Journal of Remote Sensing*, 45, 2622–2650. <https://doi.org/10.1080/01431161.2024.2329528>
- Xiao, L., Li, W., Yao, S., Liu, H., & Ren, D. (2024). High-precision and lightweight small-target detection algorithm for low-cost edge intelligence. *Scientific Reports*, 14(1), 23542. <https://doi.org/10.1038/s41598-024-75243-1>
- Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., & Xia, G.-S. (2022). Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190, 79–93. <https://doi.org/10.1016/j.isprsjprs.2022.06.002>
- Yang, C., Huang, Z., & Wang, N. (2022). QueryDet: Cascaded Sparse Query for Accelerating High-Resolution Small Object Detection. In K. Dana, G. Hua, S. Roth, S. Dimitris, & R. Singh (Eds.), *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13658–13667). IEEE. <https://doi.org/10.1109/CVPR52688.2022.01330>
- Zhang, H., Liu, K., Gan, Z., & Zhu, G.-N. (2025). UAV-DETR: Efficient End-to-End Object Detection for Unmanned Aerial Vehicle Imagery. arXiv. <https://doi.org/10.48550/arXiv.2501.01855>
- Zhang, H., & Zhang, S. (2024). *Shape-IoU: More Accurate Metric considering Bounding Box Shape and Scale*. arXiv. <https://doi.org/10.48550/arXiv.2312.17663>
- Zhang, L., Xiong, N., Pan, X., Yue, X., Wu, P., & Guo, C. (2023). Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery. *Algorithms*, 16(11), 520. <https://doi.org/10.3390/a16110520>

- Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs Beat YOLOs on Real-time Object Detection. In E. Mortensen (Ed.), *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16965–16974). IEEE. <https://doi.org/10.1109/cvpr52733.2024.01605>
- Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C., Liu, X., Ma, W., Wu, H., Wang, L., Schumann, A., Brown, C., Qian, C., Li, C., Li, D., Michail, E., & Song, Z. (2019). VisDrone-DET2018: The Vision Meets Drone Object Detection in Image Challenge Results. In L. Leal-Taixé & S. Roth (Eds.), *Computer Vision – ECCV 2018 Workshops* (Vol. 11134, pp. 437–468). https://doi.org/10.1007/978-3-030-11021-5_27

Отримано редакцією журналу / Received: 08.08.25

Прорецензовано / Revised: 15.02.26

Схвалено до друку / Accepted: 16.04.26

Опубліковано / Published: 05.06.26

Ярослав СМОЛІН, асп.

ORCID ID: 0009-0007-9744-6496

e-mail: smolin.yaroslav@iit.kpi.ua

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Вячеслав ЛІСКІН, канд. техн. наук, доц.

ORCID ID: 0000-0002-9418-0633

e-mail: liskin.viacheslav@iit.kpi.ua

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

МЕТААНАЛІЗ ПАРЕТО-ОПТИМАЛЬНИХ МОДЕЛЕЙ ВИЯВЛЕННЯ ОБ'ЄКТІВ ДЛЯ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ НА НАБОРІ ДАНИХ VISDRONE2019-DET

Предметом пропонованого дослідження є сучасні моделі глибокого навчання для виявлення об'єктів із безпілотних літальних апаратів (БПЛА), навчені й оцінені на еталонному наборі даних VisDrone2019-DET. Проблема, яку розглянуто в роботі, полягала у відсутності систематичного аналізу наявних моделей на основі критерію оптимального балансу між точністю виявлення, швидкістю обчислення та розміром моделі, що є критично важливими для роботи в режимі реального часу на бортових комп'ютерах БПЛА з обмеженими ресурсами. Основний результат полягає у всебічному метааналізі понад 60 наукових публікацій і визначенні двох груп Парето-оптимальних моделей: одна зосереджена на мінімізації параметрів моделі, а інша – на мінімізації обчислювальної складності. За допомогою багатокритеріального аналізу Парето та створення найповнішої на даний момент таблиці підсумкових показників ефективності, у ході дослідження було визначено архітектури, що забезпечують найкращий компроміс між точністю, швидкістю та розміром. Ними виявилися оптимізовані версії YOLO та детекторів на основі трансформерів, таких як DETR, що пояснюється ефективністю їхньої основної архітектури та додатковим налаштуванням під особливості аерофотознімки. Використовуючи прискорювач Nvivo 8L як приклад, дослідження також продемонструвало, що обчислювальна складність моделі прямо пропорційна (з коефіцієнтом детермінації до 0,98) її швидкості виконання на типовому нейронному прискорювачі, який використовується в комп'ютерах-супутниках БПЛА. Це дає змогу інженерам оцінювати відносний час виконання на основі кількості операцій, уникаючи трудомісткого тестування. Загалом отримані результати можуть бути використані інженерами для ухвалення обґрунтованих рішень щодо архітектури детекторів залежно від конкретних обмежень апаратного забезпечення БПЛА, а дослідниками – як орієнтири для розроблення нових моделей для автономної експлуатації БПЛА в повітряному моніторингу, інспекції інфраструктури та пошуково-рятувальних операціях.

Ключові слова: безпілотні літальні апарати, VisDrone2019, детекція об'єктів, моделі глибокого навчання, аналіз Парето.

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; у рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.