

# ODverse33: Is the New YOLO Version Always Better? A Multi Domain benchmark from YOLO v5 to v11

Tianyou Jiang<sup>1,\*</sup>, Yang Zhong<sup>1,\*</sup>

College of Information Science and Engineering, Shandong Agricultural University,  
Tai'an 271018, China  
s1729041183@gmail.com, inawjsyg@gmail.com

**Abstract.** You Look Only Once (YOLO) models have been widely used for building real-time object detectors across various domains. With the increasing frequency of new YOLO versions being released, key questions arise. Are the newer versions always better than their previous versions? What are the core innovations in each YOLO version and how do these changes translate into real-world performance gains? In this paper, we summarize the key innovations from YOLOv1 to YOLOv11, introduce a comprehensive benchmark called ODverse33, which includes 33 datasets spanning 11 diverse domains (Autonomous driving, Agricultural, Underwater, Medical, Videogame, Industrial, Aerial, Wildlife, Retail, Microscopic, and Security), and explore the practical impact of model improvements in real-world, multi-domain applications through extensive experimental results. We hope this study can provide some guidance to the extensive users of object detection models and give some references for future real-time object detector development.

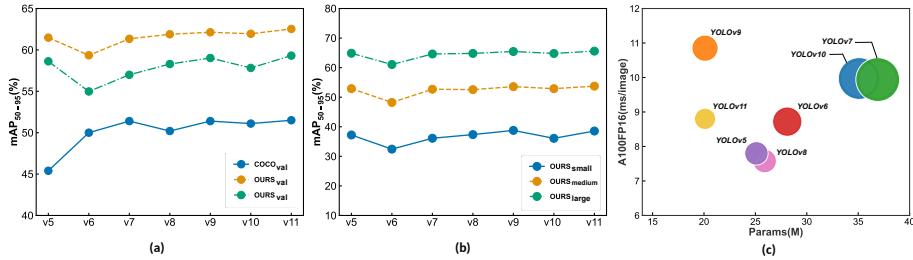
**Keywords:** YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLOv9, YOLOv10, YOLOv11, Object Detection

## 1 Introduction

With the rapid advancement in both artificial intelligence and computer vision, object detection has seen remarkable breakthroughs, garnering considerable attention in recent years [55]. One of the most influential models in this field is You Only Look Once (YOLO), first introduced by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in 2016 [34]. YOLO quickly gained traction due to its impressive detection accuracy and fast inference speed, making it a leading choice for real-time object detection tasks. Since its inception till January 1, Year 2025, YOLO has evolved through 11 major versions (YOLOv1 to YOLOv11). While the original authors developed YOLOv1 through YOLOv3, subsequent versions (YOLOv4 to YOLOv11) were developed by different teams within the active YOLO community. Despite sharing the core YOLO concepts, many later

---

\* Equal contribution and corresponding authors



**Fig. 1:** Evaluation results of YOLOv5 to YOLOv11. (a) Performance on the COCO validation set (reported in their original projects) and on our ODverse33 validation and test sets. (b) Performance on small, medium, and large objects in the ODverse33 test set. (c) Inference speed per image using a single NVIDIA A100 GPU and number of parameters for each YOLO model, where the size of the circles represents the product of these two metrics.

versions integrate innovative techniques and architectural changes, reflecting diverse contributions and advancements in object detection methodologies.

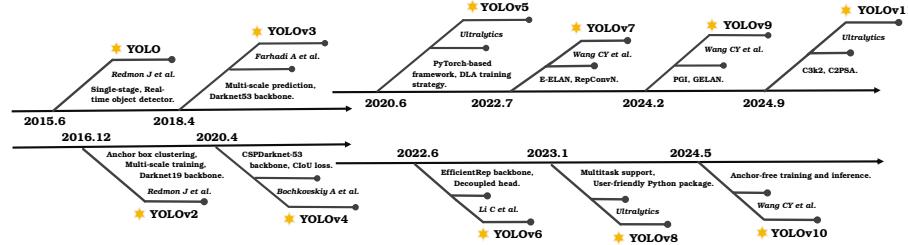
Evaluations of object detectors, including the YOLO series, have traditionally focused on the Common Objects in Context (COCO) dataset, where higher YOLO versions have consistently demonstrated improved performance in their original projects or papers [24]. However, a significant gap remains in understanding how different YOLO versions perform across various domain-specific datasets, making this a topic of strong research interest. Is the new version of YOLO always better than the previous version on different tasks? How should practitioners select the best YOLO version for their specific projects? What are the core improvements in each YOLO version, and how do these changes translate into real-world performance gains? This paper aims to address these questions through a comprehensive analysis and in-depth discussion.

While the YOLO series has been widely adopted across various real-world applications, a common misconception still persists that newer YOLO versions are inherently superior, much like the assumption that the latest hardware updates always lead to better performance. However, this is not necessarily the case. Adding a sleek spoiler doesn't necessarily improve a car's performance, the effectiveness of a YOLO model should depend on its alignment with the task at hand, not just its novelty. While YOLOv1 through YOLOv5 marked significant evolutionary milestones, later versions focused on refining existing architectures to further improve model performance and inference efficiency. These refinements, though valuable, may not always translate into robust improvements for broader object detection tasks beyond COCO benchmark. As shown in Figure 1, our results indicate that post-YOLOv5 versions exhibit fluctuating performance across domain-specific applications, sometimes failing to surpass their predecessors.

In this paper, we aim to reveal the evolution of YOLO series models, provide a comprehensive benchmark, offer guidance for those involved in object detection applications, and propose references for the development of future real-time

object detectors. To achieve this, we summarize the key innovations introduced in YOLOv1 through YOLOv11 and conduct extensive evaluations of the models from YOLOv5 to YOLOv11. Unlike most existing benchmarks that focus primarily on the COCO dataset, our study spans a variety of domains and conditions. We trained and evaluated YOLO models on 33 datasets, covering a wide range of applications, including aerial imagery, autonomous driving, medical imaging, microscopy, underwater imaging, agriculture, industry, video games, wildlife, retail, and security. This diverse, multidisciplinary dataset collection allows us to assess each YOLO version’s performance across a broad spectrum of real-world challenges. Meanwhile, to ensure a fair and consistent benchmark, we standardized experimental setups, including uniform dataset splits, consistent data augmentation techniques, and the same hyperparameter configurations for each YOLO model. Each model was trained for 300 epochs on each dataset, and their evaluation metrics on the test sets— $mAP_{50}$ ,  $mAP_{50-95}$ ,  $mAP_{small}$ ,  $mAP_{medium}$ , and  $mAP_{large}$  were calculated and compared. We refer to this benchmark as the ODverse33 benchmark, which encompasses 33 object detection datasets across various domains. More details about the benchmark and related information about the datasets and experiments involved in this paper can be found at <https://github.com/SkyCol/ODverse33>.

## 2 Related works



**Fig. 2:** Timeline of YOLO’s development, reflecting the earliest release time of their code repository or pre-print.

### 2.1 YOLO: You Only Look Once model series

YOLO (You Only Look Once) was first introduced by Joseph Redmon and Ali Farhadi et al. in 2015 [34], marking a major advancement in real-time object detection by unifying the tasks of bounding box prediction and class probability estimation into a single-stage network. Unlike traditional two-stage detectors like R-CNN and Faster R-CNN [12, 37], YOLO achieves rapid detection speeds by predicting both bounding boxes and class probabilities directly from full images in a single forward pass. A timeline of YOLO’s development is shown in Figure 2,

where the time reflects the earliest release time of their code repository or pre-print.

In 2016 and 2018, the original authors introduced YOLOv2 and YOLOv3, respectively, further enhancing YOLO model’s detection capability [35,36]. YOLOv2, also known as YOLO9000, introduced anchor boxes for more precise detection and enabled real-time recognition across over 9000 classes by pre-training its Darknet-19 backbone on a general classification task. During training, it incorporated a multi-scale training approach, which allows the model to adapt to various input sizes, improving versatility. With k-means clustering, YOLOv2 optimized anchor box sizes, while multi-scale training enhances versatility across various input sizes. YOLOv3 improved accuracy by incorporating a deeper architecture (Darknet-53) and multi-scale predictions, which detects objects at three different scales to capture varying sizes more effectively, similar to the Feature Pyramid Networks (FPN) [23].

In 2020, Alexey Bochkovskiy and his collaborators introduced YOLOv4 [2], delivering substantial improvements in architecture and training techniques. YOLOv4 upgraded the backbone to CSPDarknet-53, incorporated Cross Stage Partial (CSP) connections [47], and added an FPN-PAN (Path Aggregation Network) feature that combines top-down and bottom-up feature fusion for enhanced multi-scale object detection [27]. It also introduced CIoU (Complete Intersection over Union) loss [54], improving bounding box localization accuracy by factoring in center distance, aspect ratio, and overlap area.

Later in 2020, Ultralytics released YOLOv5 [16], marking a new era for the YOLO family by introducing a flexible, PyTorch-based framework that emphasizes usability, modularity, and ease of deployment. Building on YOLOv4’s innovations, YOLOv5 featured multiple model sizes (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) to accommodate different computational needs. It also implemented Dynamic Label Assignment (DLA) to enhance training efficiency by dynamically selecting the best positive samples [8].

Between 2022 and 2024, researchers and Ultralytics introduced additional YOLO versions. In 2022, Meituan’s visual intelligence team released YOLOv6, featuring EfficientRep as its backbone and a decoupled head that separates classification and localization tasks to enhance precision [20]. The same year, YOLOv7 developed by Chien-Yao Wang et al. introduced advanced E-ELAN (Extended Efficient Layer Aggregation Network) and Re-Parameterized Convolution (RepConvN) to strengthen the backbone and improve model performance [45].

In 2023, Ultralytics released their second YOLO repository, YOLOv8 [17]. Building on the foundation of YOLOv5, YOLOv8 introduced updates to the model architecture and added support for various tasks by incorporating additional heads for instance segmentation, pose keypoint detection, oriented bounding box (OBB) detection, and classification tasks. Moreover, it provided a unified PyTorch-based interface through the Ultralytics Python package, allowing users to more easily train, validate, and deploy the model with minimal configuration.

In 2024, YOLOv9 was introduced by Chien-Yao Wang and his collaborators [48]. As the authors of YOLOv7, they designed YOLOv9 to integrate Pro-

grammable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), where PGI was developed to overcome the information bottleneck problem and GELAN was developed by combining CSPNet and ELAN to improve architectural efficiency and performance.

Later in 2024, Ao Wang, Hui Chen, and their collaborators released YOLOv10 [46], which uses a One-to-Many head to generate multiple predictions per object during training and a One-to-One head to generate a single best prediction per object during inference, whereas some previous YOLO versions usually achieve anchor-free detection by directly predicting object centers and sizes. However, we find YOLOv10 has been affected in terms of detection performance, where it performs comparatively poorly on detection precision, particularly for small objects (see Figure 1).

Most recently, and also in 2024, Ultralytics released their third repository, YOLOv11 [18]. Building upon the impressive advancements of YOLOv8, YOLOv11 further improves the backbone using C3k2 (Cross Stage Partial with kernel size 2) blocks and C2PSA (Convolutional block with Parallel Spatial Attention) components. Similar to YOLOv8, YOLOv11 supports a range of tasks, including object detection, instance segmentation, pose estimation, and OBB detection, positioning it as one of the most versatile and capable object detectors to date.

## 2.2 YOLO applications

The YOLO series has become one of the most widely used methods for real-time object detection, with extensive applications in both academia and industry. Its versatility spans autonomous driving, remote sensing, robotics, surveillance, facial recognition, visual search engines, and numerous other domains. However, with the rapid emergence of new YOLO versions, many researchers face uncertainty when selecting the most suitable model for their specific tasks.

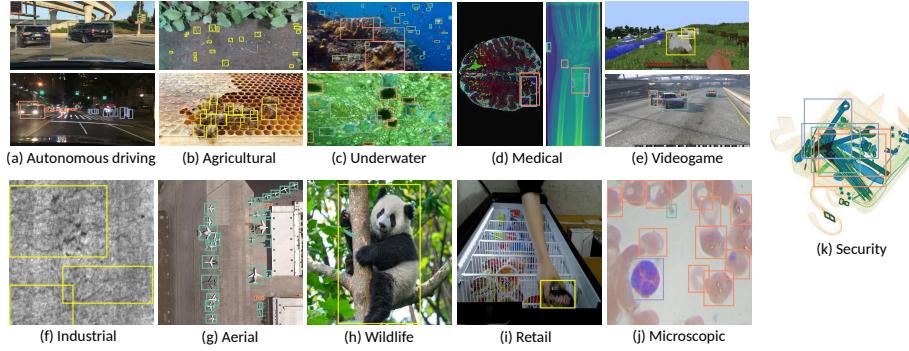
While newer YOLO versions continue to be released, studies have shown that models from YOLOv6 onward do not always outperform their predecessors in domain-specific applications. For example, in a study on wheat head counting, YOLOv7 outperformed YOLOv8, while in underwater pipeline detection, YOLOv5 achieved better results than YOLOv6, YOLOv7, and YOLOv8 [7]. Similarly, a study on hazards in knife handling found that YOLOv5 and YOLOv8 exhibited higher detection accuracy than YOLOv10, while YOLOv10 had the highest misclassification rate [9].

Notably, in 32 preprints and indexed papers we collected that compared YOLOv9 and YOLOv10, 26 papers reported that YOLOv9 outperformed YOLOv10, highlighting the uncertainty and domain-specific performance of YOLO upgrades in real-world applications. To address these inconsistencies, this paper presents a comprehensive benchmark ODverse33 to evaluate different YOLO models across multiple domains, providing clear guidance for model selection.

### 3 Methods

#### 3.1 Multi Domain Datasets

A total of 33 datasets are included in our ODverse33 benchmark, which together comprise 3.98 million instances. In some cases, certain targets may appear partially within the boundary areas of images. For instance, in drone-captured images of wheat, portions of wheat heads might be visible only at the edges, and while the straw portion of the wheat head is important for defining the object, it may or may not lie within the image boundary. This introduces ambiguity regarding whether such instances should be considered as valid objects for detection, which in turn affects the evaluation of experimental results. To address this issue and improve the reliability of our benchmark, we excluded datasets where such ambiguities were prevalent.



**Fig. 3:** Sample images in 11 domains of the ODverse33 benchmark.

Datasets in ODverse33 benchmark span 11 diverse domains: **a. Autonomous driving**, comprising datasets of BDD100K (diverse driving dataset) [53], KITTI (autonomous driving dataset) [10], TSDD (traffic sign dataset) [39]. **b. Agricultural**, comprising datasets of WeedCrop (weed detection dataset) [40], Honey-Bee (honeybee detection dataset) [33], Pear640 (pear detection dataset) [19]. **c. Underwater**, comprising datasets of DUO (underwater dataset for robot picking) [25], RUOD (dataset for underwater detection in general scene) [5], UWD (underwater waste dataset) [49]. **d. Medical**, comprising datasets of ChestX-Det (chest X-ray dataset) [26], GRAZPEDWRI-DX (pediatric wrist trauma radiography dataset) [32], BCD (brain cancer dataset) [52], BBD (broken areas of body dataset) [44]. **e. Videogame**, comprising datasets of MC (first-person perspective images in Minecraft) [31], CS2 (first-person perspective images from Counter Strike 2) [14], GTA5 (monitoring perspective images from Grand Theft Auto V) [3]. **f. Industrial**, comprising datasets of DeepPCB (PCB defects dataset) [42], GC10-DET (metallic surface defect detection dataset) [28], NEU-SDD (surface defect database) [15]. **g. Aerial**, comprising datasets of DIOR

(optical remote sensing detection dataset) [21], DOTA (dataset for object detection in general aerial images) [50], HIT-UAV (high-altitude infrared thermal dataset) [41]. **h. Wildlife**, comprising datasets of ADID (animal detection dataset) [1], EAD (endangered animal detection dataset) [4]. **i. Retail**, comprising datasets of SFD (smart fridge detection dataset) [11], Holoselecta (packaged products detection dataset) [6], SKU110K (dataset for object detection in densely packed scenes) [13]. **j. Microscopic**, comprising datasets of BCCD (blood cell count dataset) [51], LDD (liver disease dataset) [38], MiAMIA-SVDS (sperm detection dataset) [29]. **k. Security**, comprising datasets of SIXray (security inspection x-ray dataset) [30], HiXray (security inspection x-ray dataset) [43], MGD (non-canonical firearm detection dataset) [22]. Samples from these 11 diverse domains are shown in Figure 3.

Specifically, for datasets with predefined training, validation, and test set partitions, we preserved their original partitioning. For datasets without such divisions, or when test set labels were unavailable, we split the data into training, validation, and test sets using an 8:1:1 ratio. This is to ensure that the evaluation results can be derived from the test sets, rather than being traditionally based solely on the validation set results.

### 3.2 Experimental Setups

To ensure a fair benchmark, we meticulously adjusted the preprocessing procedures for each YOLO model to maintain a consistent setup across all experiments. During training, all models were uniformly configured with identical data augmentation techniques, including Random Translation, Random Scaling, Flip, Color Augmentation, and Mosaic. To avoid any potential decline in model performance, certain data augmentation methods, such as Random Covering and Random Cropping, were excluded. Hyperparameters were consistently set with a batch size of 32 and an image size of 640 x 640, with images padded and resized to form square inputs for batching. Each model was trained for 300 epochs per dataset for several times, and the final performance metrics were computed on the test sets from the best run checkpoint.

### 3.3 Evaluation Metrics

To establish a comprehensive benchmark, we evaluate object detection performance on the test sets using several core metrics, including mean average precision (mAP) evaluated at Intersection over Union (IoU) thresholds of 50% (mAP<sub>50</sub>), and the mean average precision across IoU thresholds from 50% to 95% (mAP<sub>50-95</sub>). In addition, we assess mAP performance for small (mAP<sub>small</sub>), medium (mAP<sub>medium</sub>), and large objects (mAP<sub>large</sub>) respectively at IoU thresholds ranging from 50% to 95%. Objects are categorized based on their size and their proportion relative to the total image area as follows [24]: small objects have an area smaller than 32 x 32 pixels or cover less than 0.1% of the total image area; medium objects have an area between 32 x 32 and 96 x 96 pixels, or

occupy between 0.1% and 1% of the total image area; and large objects have an area larger than  $96 \times 96$  pixels or cover more than 1% of the total image area.

Specifically, many YOLO frameworks filter predictions based on a pre-set confidence threshold, such as 0.25 or 0.5, to focus on high-confidence detections. However, in our evaluation, which follows the COCO evaluation standard, no such filtering is applied. The COCO evaluation standard provides a rigorous and comprehensive assessment by evaluating predictions across a range of confidence thresholds—from 0.0 to 1.0 in 0.01 increments. This exhaustive evaluation may result in relatively lower metrics compared to those displayed by some YOLO projects.

## 4 Experiments

### 4.1 ODverse33 benchmark

Our ODverse33 benchmark comprises 33 datasets spanning 11 diverse domains. Across all these datasets, we observed fluctuations in the performance of YOLO series models, as illustrated in Figure 1 and detailed in Table 1. The respective results across 11 diverse domains are presented in Table 2. More detailed experimental results for each dataset can be found in the appendix.

	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
mAP <sub>50</sub>	0.7846	0.7674	0.7826	0.7812	0.7913	0.7761	<b>0.7927</b>
mAP <sub>50–95</sub>	0.5862	0.5498	0.5699	0.5829	0.5902	0.5782	<b>0.5931</b>
mAP <sub>small</sub>	0.3722	0.3243	0.3612	0.3735	<b>0.3877</b>	0.3609	0.3855
mAP <sub>medium</sub>	0.5290	0.4822	0.5269	0.5256	0.5357	0.5289	<b>0.5374</b>
mAP <sub>large</sub>	0.6487	0.6106	0.6463	0.6481	0.6546	0.6480	<b>0.6559</b>

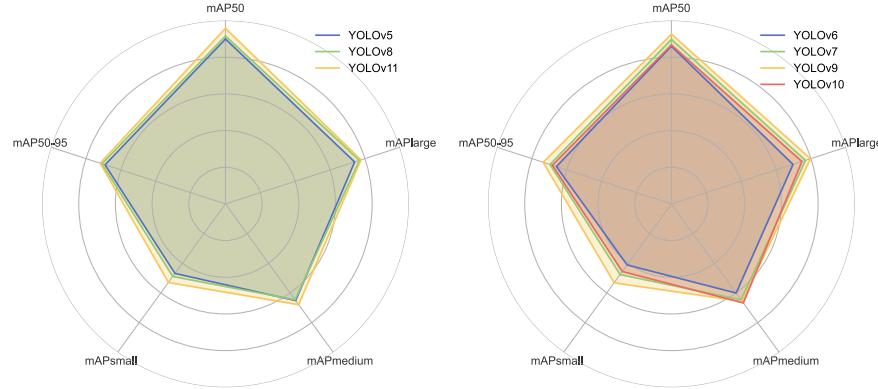
**Table 1:** Overall results.

For the overall results across 33 datasets from all 11 domains, the ranking of these seven models based on mAP<sub>50</sub> is as follows: YOLOv11, YOLOv9, YOLOv5, YOLOv7, YOLOv8, YOLOv10, and YOLOv6. A similar trend is observed for mAP<sub>50–95</sub>, with the ranking being YOLOv11, YOLOv9, YOLOv5, YOLOv8, YOLOv10, YOLOv7, and YOLOv6. These results highlight YOLOv11 as the most accurate model overall while also reinforcing the notion that newer YOLO versions are not necessarily superior. Notably, YOLOv10 underperforms compared to YOLOv8, and YOLOv6 lags behind YOLOv5. Among the 11 domains included in our benchmark, YOLOv11 achieves the highest mAP<sub>50</sub> for 6 domains (Aerial, Agricultural, Autonomous driving, Video-game, Microscopic, and Wildlife). Given the significance of mAP<sub>50</sub> in building real-world applications, YOLOv11 demonstrates outstanding capabilities. In the remaining five domains, different YOLO versions perform best: YOLOv9 excels in industrial and medical images, YOLOv8 leads in retail and security images, while YOLOv5 achieves the highest mAP<sub>50</sub> for Underwater images.

Domain	Metric	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Aerial	mAP <sub>50</sub>	0.6968	0.6687	0.6910	0.6966	0.7065	0.6915	<b>0.7101</b>
	mAP <sub>50–95</sub>	<b>0.5175</b>	0.4643	0.4865	0.5039	0.5060	0.5024	0.5104
	mAP <sub>small</sub>	0.2476	0.1936	0.2327	0.2388	0.2400	0.2311	<b>0.2494</b>
	mAP <sub>medium</sub>	0.4592	0.4379	0.4543	0.4668	0.4632	0.4558	<b>0.4768</b>
	mAP <sub>large</sub>	0.7248	0.6496	0.7046	0.7157	0.7314	<b>0.7370</b>	0.7207
Agricultural	mAP <sub>50</sub>	0.8832	0.8681	0.8749	0.8648	0.8781	0.8752	<b>0.8922</b>
	mAP <sub>50–95</sub>	<b>0.6637</b>	0.5924	0.6103	0.6505	0.6127	0.6503	0.6562
	mAP <sub>small</sub>	0.4391	0.3964	0.4389	0.4097	0.4380	0.4219	<b>0.4857</b>
	mAP <sub>medium</sub>	0.7019	0.6558	0.6702	<b>0.7223</b>	0.6695	0.7093	0.7016
	mAP <sub>large</sub>	0.7944	0.6646	0.7346	<b>0.8218</b>	0.7333	0.7964	0.7999
Autonomous Driving	mAP <sub>50</sub>	0.7277	0.7169	0.7325	0.7302	0.7361	0.7323	<b>0.7384</b>
	mAP <sub>50–95</sub>	0.5850	0.5541	0.5772	0.5900	<b>0.5976</b>	0.5902	0.5956
	mAP <sub>small</sub>	<b>0.4506</b>	0.3590	0.4239	0.4345	0.4388	0.4346	0.4447
	mAP <sub>medium</sub>	0.6229	0.5929	0.6130	0.6242	<b>0.6378</b>	0.6329	0.6332
	mAP <sub>large</sub>	0.6787	0.6720	0.6809	0.6679	0.6917	0.6812	<b>0.6925</b>
Videogame	mAP <sub>50</sub>	0.9125	0.9224	0.9382	0.9381	0.9372	0.9369	<b>0.9436</b>
	mAP <sub>50–95</sub>	0.8026	0.7716	0.7705	<b>0.8141</b>	0.8091	0.7954	0.8035
	mAP <sub>small</sub>	0.5112	0.4087	0.5106	<b>0.5464</b>	0.5162	0.5113	0.4785
	mAP <sub>medium</sub>	0.7738	0.7150	0.7144	<b>0.7847</b>	0.7830	0.7748	0.7761
	mAP <sub>large</sub>	0.8859	0.8883	0.8489	<b>0.9008</b>	0.8950	0.8655	0.8843
Industrial	mAP <sub>50</sub>	0.7232	0.7061	0.7239	0.7305	<b>0.7621</b>	0.7207	0.7478
	mAP <sub>50–95</sub>	0.4737	0.4267	0.4683	0.4602	0.4864	0.4622	<b>0.4876</b>
	mAP <sub>small</sub>	0.6027	0.5417	0.4537	0.5783	0.6186	0.5817	<b>0.6396</b>
	mAP <sub>medium</sub>	0.4267	0.3805	0.3624	0.3994	<b>0.4318</b>	0.4109	0.4152
	mAP <sub>large</sub>	0.3459	0.3221	<b>0.3922</b>	0.3520	0.3771	0.3648	0.3812
Medical	mAP <sub>50</sub>	0.6848	0.6626	0.6862	0.6833	<b>0.7255</b>	0.6624	0.6973
	mAP <sub>50–95</sub>	0.4537	0.4105	0.4603	0.4404	<b>0.4848</b>	0.4333	0.4653
	mAP <sub>small</sub>	0.2201	0.2316	<b>0.3198</b>	0.2163	0.3090	0.2006	0.2836
	mAP <sub>medium</sub>	0.4616	0.4231	<b>0.4741</b>	0.4092	0.4651	0.4320	0.4728
	mAP <sub>large</sub>	0.5541	0.5596	0.5769	0.5390	<b>0.5814</b>	0.5297	0.5607
Microscopic	mAP <sub>50</sub>	0.7295	0.7264	0.7326	0.7122	0.7258	0.7204	<b>0.7384</b>
	mAP <sub>50–95</sub>	0.5115	0.5004	0.5131	0.5046	0.5132	0.5055	<b>0.5207</b>
	mAP <sub>small</sub>	0.2999	0.3036	0.3026	0.2980	0.2941	<b>0.3136</b>	0.3063
	mAP <sub>medium</sub>	0.5788	0.5625	0.5747	0.5727	0.5859	0.5676	<b>0.5867</b>
	mAP <sub>large</sub>	0.6000	0.5559	<b>0.6163</b>	0.5944	0.6109	0.6123	0.5843
Retail	mAP <sub>50</sub>	0.8042	0.7668	0.8040	<b>0.8101</b>	0.7958	0.7886	0.7978
	mAP <sub>50–95</sub>	0.6333	0.5562	0.6309	<b>0.6377</b>	0.6265	0.6159	0.6305
	mAP <sub>small</sub>	0.2117	0.1518	0.2349	0.2471	<b>0.2523</b>	0.2184	0.2279
	mAP <sub>medium</sub>	0.5079	0.4292	0.5138	<b>0.5138</b>	0.5099	0.4979	0.5099
	mAP <sub>large</sub>	0.6577	0.5833	0.6573	<b>0.6631</b>	0.6483	0.6421	0.6574
Security	mAP <sub>50</sub>	0.8682	0.8544	0.8649	<b>0.8701</b>	0.8692	0.8482	0.8662
	mAP <sub>50–95</sub>	0.5790	0.5625	0.5841	0.5849	<b>0.5912</b>	0.5841	0.5909
	mAP <sub>small</sub>	0.4720	0.4281	0.4505	0.4614	0.4664	0.4711	<b>0.5025</b>
	mAP <sub>medium</sub>	0.5113	0.4569	<b>0.5334</b>	0.5203	0.5235	0.5175	0.5306
	mAP <sub>large</sub>	0.6334	0.6081	0.6367	0.6333	<b>0.6470</b>	0.6423	0.6373
Underwater	mAP <sub>50</sub>	<b>0.7978</b>	0.7723	0.7792	0.7884	0.7703	0.7759	0.7922
	mAP <sub>50–95</sub>	0.5827	0.5597	0.5698	0.5753	0.5790	0.5738	<b>0.5895</b>
	mAP <sub>small</sub>	0.2508	0.2379	0.2360	0.2565	<b>0.2947</b>	0.2250	0.2575
	mAP <sub>medium</sub>	0.4751	0.4416	0.4524	0.4742	0.4790	0.4655	<b>0.4889</b>
	mAP <sub>large</sub>	0.6045	0.5819	0.5923	0.6002	0.6037	0.5987	<b>0.6143</b>
Wildlife	mAP <sub>50</sub>	0.7732	0.7701	0.7824	0.7687	0.7937	0.7496	<b>0.7959</b>
	mAP <sub>50–95</sub>	0.6455	0.6494	0.6579	0.6595	<b>0.6744</b>	0.6473	0.6709
	mAP <sub>small</sub>	-	-	-	-	-	-	-
	mAP <sub>medium</sub>	0.1478	0.0890	<b>0.1884</b>	0.1463	0.1644	0.1573	0.1551
	mAP <sub>large</sub>	0.6559	0.6612	0.6682	0.6610	<b>0.6846</b>	0.6577	0.6820

**Table 2:** Results across 11 diverse domains, with the best-performing evaluation metrics in each domain highlighted in bold.

## 4.2 YOLO Community



**Fig. 4:** Comparison of models developed by different teams.

After YOLOv5, various teams in the YOLO open-source community introduced new models. While models from YOLOv5 to YOLOv11 show fluctuating performance across different domains, those developed by Ultralytics demonstrate a consistent improvement. As depicted in Figure 4(a), the radar map areas for YOLOv5, YOLOv8, and YOLOv11 gradually expand, highlighting their steady progress. In contrast, as shown in Figure 4(b), models from other teams rank as follows on the radar map: YOLOv9, YOLOv7, YOLOv10, and YOLOv6. Notably, YOLOv9, developed by the same team behind YOLOv7, also outperforms YOLOv7 in several key metrics.

Specifically, YOLOv9 excels in detecting small objects, demonstrating the high efficiency of its PGI and GELAN architecture. These innovations effectively address the information bottleneck problem, contributing to enhanced detection performance. The PGI module, which incorporates the concept of multi-level auxiliary information, integrates an additional network between the feature pyramid hierarchy layers of auxiliary supervision and the main branch. By merging returned gradients from different prediction heads, this mechanism has proven effective for detecting small objects in our ODverse33 benchmark.

## 5 Discussion

In this paper, we provide a comprehensive overview of the core innovations introduced from YOLOv1 to YOLOv11, tracing the development history and framework updates of each model. Rather than relying solely on the conventional

COCO training and validation sets, we introduce the ODverse33 benchmark, which comprises 33 diverse datasets across 11 distinct domains. This benchmark allows for a more nuanced and comprehensive evaluation of model performance, reflecting a broader range of real-world applications. By leveraging ODverse33, researchers and practitioners can make more informed decisions tailored to their specific tasks and objectives.

The multi-domain nature of the ODverse33 benchmark reveals key insights into the performance of different YOLO versions across varied application areas. Notably, YOLOv11 performs particularly well in detecting objects within aerial, agricultural, autonomous driving, video game, microscopic, and wildlife imagery. These domains often involve complex scenarios, such as small object detection, occlusion, and varying image resolutions, where YOLOv11 demonstrates its robustness. Meanwhile, YOLOv9 shows particular strength in processing industrial and medical images, with a special emphasis on small object detection. YOLOv9 achieves the highest  $mAP_{small}$  score among all evaluated models, underscoring its effectiveness in environments where precise localization of small objects is critical.

These findings provide valuable guidance for researchers and practitioners in selecting the most appropriate YOLO model based on the specific demands of their application domains. The ability to choose a model with tailored strengths allows for more efficient and effective application development, ultimately improving the performance of their projects.

Overall, the ODverse33 benchmark reveals the fluctuation of model performance across different YOLO versions and professional domains, emphasizing that the newer YOLO versions are not always guaranteed to outperform their predecessors. The fluctuation in performance highlights that, despite advancements in model architecture and training strategies, improvements may not always translate into better results across all domains. This observation challenges the common assumption that the latest versions are universally superior and suggests that careful evaluation across diverse contexts is essential.

The comparison between different YOLO versions also underscores the influence of the development teams behind each model. Notably, models released by the same team often exhibit a consistent trajectory of improvement. For example, YOLOv5, YOLOv8 and YOLOv11, all developed by Ultralytics, showcase a clear and steady advancement in performance, reflecting the team's strong focus on refining and optimizing their models. Similarly, YOLOv7 and YOLOv9, created by another research group, also show consistent progress, with YOLOv9 outperforming its predecessor. These trends highlight the importance of long-term commitment and iterative refinement by dedicated development teams, with Ultralytics standing out as a particularly reliable force in the YOLO development community.

## References

1. Antoreepjana: Animals detection images dataset. Kaggle (2023), available

- at: <https://www.kaggle.com/datasets/antoreepjana/animals-detection-images-dataset/data>. Accessed on: Dec. 13, 2024
2. Bochkovskiy, Alexey, W.C.Y.L.H.Y.M.: Yolov4: Optimal speed and accuracy of object detection (2020), in arXiv preprint arXiv:2004.10934
  3. Drone: Detect object gta5 dataset. Roboflow Universe (2023), available at: <https://universe.roboflow.com/drone-hikxp/detect-object-gta5>. Accessed on: Dec. 13, 2024
  4. Endangeredanimals: Endangered animal detection dataset. Roboflow Universe, available at: <https://universe.roboflow.com/endangeredanimals/endangered-animal-detection-fm25l>. Accessed on: Dec. 13, 2024
  5. Fu, C., L.R.F.X.C.P.F.H.Y.W.o.L.Z.: Rethinking general underwater object detection: Datasets, challenges, and solutions. Neurocomputing **517**, 243–256 (2023)
  6. Fuchs, Klaus, G.T.F.E.: Towards identification of packaged products via computer vision. In: Proceedings of the IoT 2019 (2019)
  7. Gašparović, B., M.G.R.J.o.: Evaluating yolov5, yolov6, yolov7, and yolov8 in underwater environment: Is there real improvement? In: Proceedings of the International Conference on Smart and Sustainable Technologies. pp. 1–4 (2023)
  8. Ge, Zheng, L.S.L.Z.o.: Ota: Optimal transport assignment for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 303–312 (2021)
  9. Geetha, A. S., H.M.: A comparative analysis of yolov5, yolov8, and yolov10 in kitchen safety (2024), in arXiv preprint arXiv:2407.20872
  10. Geiger, Andreas, L.P.U.R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR). pp. 3354–3361 (2012)
  11. Ggong: Fidge dataset. Roboflow Universe (2024), available at: <https://universe.roboflow.com/ggong/fidge>. Accessed on: Dec. 13, 2024
  12. Girshick, Ross, D.J.D.T.M.J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 580–587 (2014)
  13. Goldman, Eran, H.R.E.A.G.J.H.T.: Precise detection in densely packed scenes. In: IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR). pp. 5227–5236 (2019)
  14. Günaydin, Ö.F.: Counter strike 2 body and head classification. Kaggle (2024), available at: <https://www.kaggle.com/datasets/merfarukgnaydn/counter-strike-2-body-and-head-classification>. Accessed on: Dec. 13, 2024
  15. He, Yu, S.K.M.Q.Y.Y.: An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE Transactions on Instrumentation and Measurement **69**(4), 1493–1504 (2020)
  16. Jocher, Glenn, o.: Yolov5. GitHub repository (2020), available: <https://github.com/ultralytics/yolov5>
  17. Jocher, Glenn, o.: Yolov8. GitHub repository (2023), available: <https://github.com/ultralytics/ultralytics>
  18. Jocher, Glenn, o.: Yolov11. GitHub repository (2024), available: <https://github.com/ultralytics/ultralytics>
  19. Kodors, Sergejs, o.: Rapid prototyping of pear detection neural network with yolo architecture in photographs. In: ENVIRONMENT. TECHNOLOGIES. RESOURCES. Proceedings of the International Scientific and Practical Conference. vol. 1 (2023)
  20. Li, Cheng, o.: Yolov6: A single-stage object detection framework for industrial applications (2022), in arXiv preprint arXiv:2209.02976

21. Li, Ke, o.: Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing* **159**, 296–307 (2020)
22. Lim, JunYi, o.: Deep multi-level feature pyramids: Application for non-canonical firearm detection in video surveillance. *Engineering applications of artificial intelligence* **97**, 104094 (2021)
23. Lin, Tsung-Yi, D.P.G.R.o.: Feature pyramid networks for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2117–2125 (2017)
24. Lin, Tsung-Yi, M.M.B.S.o.: Microsoft coco: Common objects in context. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 740–755 (2014)
25. Liu, C., L.H.W.S.o.: A dataset and benchmark of underwater object detection for robot picking. In: *IEEE international conference on multimedia & expo workshops*. pp. 1–6. IEEE (2021)
26. Liu, Jingyu, L.J.Y.Y.: Chestx-det10: chest x-ray dataset on detection of thoracic abnormalities (2020), arXiv preprint arXiv:2006.10550
27. Liu, Shilong, Q.L.Q.H.o.: Path aggregation network for instance segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8759–8768 (2018)
28. Lv, Xiaoming, o.: Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **20**(6), 1562 (2020)
29. MIaMIA Group: Miamia-svds. figshare (2021), available at: <https://doi.org/10.6084/m9.figshare.15074253.v1>. Accessed on: Dec. 13, 2024
30. Miao, Caijing, o.: Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. pp. 2119–2128 (2019)
31. Minecraft Object Detection: Minecraft mob detection dataset. Roboflow Universe (2024), available at: <https://universe.roboflow.com/minecraft-object-detection/minecraft-mob-detection>. Accessed on: Dec. 13, 2024
32. Nagy, E., J.M.H.F.S.E.T.S.: A pediatric wrist trauma x-ray dataset (grazpedwridx) for machine learning. *Scientific Data* **9**(1), 222 (2022)
33. Nudi, M.: Honey bee detection model dataset. Roboflow Universe (2022), available at: <https://universe.roboflow.com/matt-nudi/honey-bee-detection-model-zgjnb>. Accessed on: Dec. 13, 2024
34. Redmon, J., D.S.G.R.F.A.: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)* (2016)
35. Redmon, Joseph, F.A.: Yolo9000: Better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)
36. Redmon, Joseph, F.A.: Yolov3: An incremental improvement (2018), in arXiv preprint arXiv:1804.02767
37. Ren, Shaoqing, H.K.G.R.o.: Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(6), 1137–1149 (2016)
38. Roboflow 100: Liver disease dataset. Roboflow Universe (2023), available at: <https://universe.roboflow.com/roboflow-100/liver-disease>. Accessed on: Dec. 13, 2024

39. Self-Driving Cars Dataset: Self-driving cars dataset. Roboflow Universe (2023), published by Roboflow. Available at: <https://universe.roboflow.com/selfdriving-car-qtywx/self-driving-cars-lfjou>. Accessed: 2024-12-13
40. Sudars, K., J.J.N.I.O.L.B.N.: Dataset of annotated food crops and weed images for robotic computer vision control. Data in brief **31**, 105833 (2020)
41. Suo, Jiashun, o.: Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection. Scientific Data **10**(1), 227 (2023)
42. Tang, Sanli, o.: Online pcb defect detector on a new pcb defect dataset (2019), arXiv preprint arXiv:1902.06197
43. Tao, Renshuai, o.: Towards real-world x-ray security inspection: A high-quality benchmark and lateral inhibition module for prohibited items detection. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). pp. 10923–10932 (2021)
44. Team KS: Broken areas of body dataset. Roboflow Universe, available at: <https://universe.roboflow.com/team-ks-ipmkt/broken-areas-of-body>. Accessed on: Dec. 13, 2024
45. Wang, Chien-Yao, B.A.L.H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7464–7475 (2023)
46. Wang, A., C.H.L.O.: Yolov10: Real-time end-to-end object detection (2024), in arXiv preprint arXiv:2405.14458
47. Wang, Chien-Yao, L.H.Y.M.W.Y.H.o.: CspNet: A new backbone that can enhance learning capability of cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 390–391 (2020)
48. Wang, Chien-Yao, Y.I.H.L.H.Y.M.: Yolov9: Learning what you want to learn using programmable gradient information. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 1–21 (2025)
49. Waste: Under waste dataset. Roboflow Universe (2024), available at: <https://universe.roboflow.com/waste-wxvgs/under-waste>. Accessed on: Dec. 13, 2024
50. Xia, Gui-Song, o.: Dota: A large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3974–3983 (2018)
51. Yao, Jiangfan, o.: High-efficiency classification of white blood cells based on object detection. Journal of Healthcare Engineering **2021**(1), 1615192 (2021)
52. YOLO: Yolo dataset. Roboflow Universe (2022), available at: <https://universe.roboflow.com/yolo-hz3ua/yolo-fj4s3>. Accessed on: Dec. 13, 2024
53. Yu, Fisher, o.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). pp. 2636–2645 (2020)
54. Zheng, Zhaozhi, W.P.L.W.o.: Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12993–13000 (2020)
55. Zou, Z., C.K.S.Z.o.: Object detection in 20 years: A survey. Proceedings of the IEEE **111**(3), 257–276 (2023)

## Appendix

Dataset	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
DIOR	0.8706	0.8659	0.8698	0.8726	<b>0.8748</b>	0.8705	0.8715
DOTA	0.4344	0.3726	0.4152	0.4343	0.4606	0.4392	<b>0.4670</b>
HIT-UAV	0.7855	0.7675	0.7880	0.7830	0.7841	0.7649	<b>0.7918</b>
HoneyBee	0.7865	0.7897	0.8021	0.7251	<b>0.8055</b>	0.7765	0.8051
Pear640	0.8788	0.8766	0.8813	0.8814	0.8941	0.8881	<b>0.8965</b>
WeedCrop	0.9844	0.9379	0.9412	0.9879	0.9346	0.9611	0.9751
BDD100K	0.3473	0.3496	0.3576	0.3495	0.3715	0.3715	<b>0.3748</b>
KITTI	0.8958	0.8921	0.8920	<b>0.9042</b>	0.8925	0.8932	0.8968
TSDD	0.9401	0.9074	<b>0.9478</b>	0.9369	0.9443	0.9323	0.9435
CS2	0.9636	0.9555	<b>0.9687</b>	0.9637	0.9632	0.9632	0.9570
GTA5	0.9449	0.9361	0.9479	0.9373	0.9363	<b>0.9506</b>	0.9503
MC	0.9187	0.8967	0.8980	0.9132	0.9143	0.8970	<b>0.9236</b>
DeepPCB	0.9810	0.9615	0.9527	0.9782	<b>0.9826</b>	0.9762	0.9804
GC10-DET	0.5498	0.5363	0.5796	0.5740	<b>0.5869</b>	0.5410	0.5826
NEU-SDD	0.6389	0.6205	0.6394	0.6394	<b>0.7167</b>	0.6449	0.6805
BCD	<b>0.7886</b>	0.7881	0.7607	0.7811	0.7773	0.7541	0.7818
BBD	0.9505	0.9315	0.9612	0.9449	<b>0.9752</b>	0.9013	0.8861
ChestX-Det	0.3934	0.3568	0.3853	0.3839	<b>0.4289</b>	0.3987	0.3847
GRAZPEDVRI-DX	0.6066	0.5740	0.6374	0.6233	0.7206	0.5955	<b>0.7364</b>
BCCD	0.8359	0.8050	0.8207	0.7707	0.7998	0.8257	<b>0.8400</b>
MlaMIA-SpermVideo	0.9784	0.9778	0.9780	<b>0.9828</b>	0.9780	0.9779	0.9821
LDD	0.3742	0.3964	0.3990	0.3830	<b>0.3995</b>	0.3576	0.3932
Holoselecta	0.8025	0.7586	0.7923	<b>0.8176</b>	0.7753	0.7537	0.7848
SKU110K	0.6298	0.6014	<b>0.6368</b>	0.6310	0.6310	0.6335	0.6306
SFD	0.9802	0.9405	<b>0.9830</b>	0.9817	0.9810	0.9787	0.9781
SIXray	0.8968	0.8969	0.8834	0.8992	<b>0.9047</b>	0.8898	0.8968
HiXray	0.8114	0.7901	<b>0.8271</b>	0.8110	0.8139	0.7962	0.8225
MGD	0.8961	0.8761	0.8843	<b>0.9002</b>	0.8889	0.8585	0.8794
DUO	0.8274	0.8106	0.8254	0.8157	<b>0.8354</b>	0.8007	0.8177
RUOD	0.8333	0.8317	0.8302	0.8334	0.8248	0.8248	<b>0.8367</b>
UWD	<b>0.7328</b>	0.6749	0.6789	0.7161	0.6576	0.7022	0.7223
AIDD	0.7273	0.7142	0.7460	0.7412	<b>0.7502</b>	0.7485	0.7423
EAD	0.8191	0.8260	0.8187	0.7962	<b>0.8395</b>	0.8206	0.8496

**Table 3:** mAP<sub>50</sub> calculated on the test set for each model across all datasets. Bold values indicate best performance for each dataset.

Dataset	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
DIOR	0.6905	0.6624	0.6721	0.7009	0.7051	0.7024	0.7053
DOTA	0.2985	0.2258	0.2667	0.3045	0.3148	0.3065	<b>0.3183</b>
HIT-UAV	0.5635	0.5048	<b>0.5208</b>	0.5062	0.4982	0.4984	0.5078
HoneyBee	<b>0.6243</b>	0.6139	0.6189	0.5534	0.6233	0.6028	0.6215
Pear640	0.5305	0.4930	0.5367	0.5399	<b>0.5459</b>	0.5289	0.5400
WeedCrop	0.8363	0.6703	0.6752	<b>0.8583</b>	0.6688	0.8192	0.8071
BDD100K	0.2098	<b>0.3512</b>	0.2169	0.2132	0.2269	0.2269	0.2282
KITTI	0.7374	0.6738	0.7000	<b>0.7519</b>	0.7509	0.7414	0.7475
TSDD	0.8079	0.7756	0.8148	0.8049	<b>0.8151</b>	0.8024	0.8112
CS2	0.8748	0.8448	0.8537	<b>0.8920</b>	0.8840	0.8695	0.8653
GTA5	0.7829	0.7930	0.7460	0.7922	<b>0.7950</b>	0.7879	0.7913
MC	0.7502	0.6769	0.7118	<b>0.7580</b>	0.7482	0.7289	0.7538
DeepPCB	0.7898	0.6900	0.5800	0.7655	<b>0.7910</b>	0.7698	0.8046
GC10-DET	0.2781	0.2848	0.2886	0.2773	0.2892	0.2729	<b>0.2938</b>
NEU-SDD	0.3540	0.3053	0.3571	0.3377	<b>0.3989</b>	0.3438	0.3627
BCD	<b>0.5718</b>	0.4992	0.5218	0.5600	0.5654	0.5413	0.5705
BBD	0.6405	0.6000	0.6887	0.6216	<b>0.7330</b>	0.6056	0.6219
ChestX-Det	<b>0.2383</b>	0.1867	0.2291	0.1930	0.2184	0.2145	0.2025
GRAZPEDVRI-DX	0.3643	0.3560	0.4014	0.3868	0.4225	0.3717	<b>0.4661</b>
BCCD	0.5650	0.5564	<b>0.5791</b>	0.5348	0.5496	0.5636	0.5673
MlaMIA-SpermVideo	0.7221	0.7063	0.6944	0.7234	0.7218	0.7187	<b>0.7281</b>
LDD	0.2473	<b>0.2686</b>	0.2657	0.2566	0.2654	0.2342	0.2667
Holoselecta	0.6176	0.5765	0.6023	<b>0.6237</b>	0.5941	0.5723	0.6024
SKU110K	0.4192	0.3441	0.4237	0.4230	<b>0.4254</b>	0.4222	0.4249
SPD	0.8630	0.7479	<b>0.8667</b>	0.8664	0.8601	0.8531	0.8641
SIXray	0.6954	0.6495	0.6832	0.7061	0.7077	0.7023	<b>0.7080</b>
HiXray	0.5289	0.4916	0.5355	0.5268	0.5346	0.5258	<b>0.5373</b>
MGD	0.5126	0.5163	<b>0.5345</b>	0.5217	0.5312	0.5243	0.5273
DUO	0.6739	0.6365	0.6454	0.6642	<b>0.6791</b>	0.6603	0.6757
RUOD	0.6326	0.6183	0.6377	0.6305	0.6276	0.6276	<b>0.6410</b>
UWD	0.4415	0.4244	0.4264	0.4311	0.4302	0.4334	<b>0.4539</b>
AIDD	0.6343	0.6148	0.6541	0.6506	<b>0.6738</b>	0.6631	0.6560
EAD	0.6563	0.6840	0.6615	0.6503	0.6863	0.6314	<b>0.6916</b>

**Table 4:** mAP<sub>50-95</sub> calculated on the test set for each model across all datasets. Bold values indicate best performance for each dataset.

Dataset	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
DIOR	0.2277	0.19845	0.22785	0.2234	0.2429	0.2215	0.2421
DOTA	0.0957	0.0742	0.0971	0.1005	0.102	0.1031	<b>0.1139</b>
HIT-UAV	<b>0.4194</b>	0.3084	0.3734	0.3925	0.3752	0.3687	0.3922
HoneyBee	0.2287	0.3549	0.3801	0.2454	<b>0.3857</b>	0.1702	0.3704
Pear640	0.3082	0.2403	0.3234	0.3269	<b>0.3699</b>	0.3336	0.3358
WeedCrop	0.7805	0.5941	0.5983	<b>0.8098</b>	0.5913	0.7619	0.7508
BDD100K	0.0778	<b>0.1244</b>	0.0851	0.0765	0.0854	0.0854	0.0866
KITTI	0.6263	0.5199	0.5465	0.6273	<b>0.6319</b>	0.6156	0.6279
TSDD	0.6476	0.4779	0.64	0.5996	0.5992	0.6027	0.6197
CS2	0.5503	0.4566	0.5446	<b>0.5654</b>	0.5386	0.4877	0.4805
GTA5	0.3953	0.416	<b>0.4693</b>	0.4535	0.4147	0.4476	0.3753
MC	0.588	0.3535	0.5179	<b>0.6202</b>	0.5799	0.5987	0.5796
DeepPCB	0.7694	0.6625	0.4819	0.728	0.7445	0.7435	<b>0.7701</b>
GC10-DET	-	-	-	-	-	-	-
NEU-SDD	0.436	0.4209	0.4254	0.4235	0.4926	0.4199	<b>0.509</b>
BCD	<b>0.2907</b>	0.2290	0.2488	0.2926	0.2848	0.2621	0.2919
BBD	-	-	-	-	-	-	-
ChestX-Det	-	-	-	-	-	-	-
GRAZPEDVRI-DX	0.1495	0.2342	<b>0.3908</b>	0.1399	0.3331	0.1391	0.2753
BCCD	0.1364	0.1713	<b>0.1814</b>	0.1394	0.123	0.1709	0.1659
MlaMIA-SpermVideo	0.7095	0.6956	0.6777	0.7081	0.7072	0.7075	<b>0.714</b>
LDD	0.0539	0.044	0.0488	0.0464	0.0524	<b>0.0623</b>	0.0389
Holoselecta	-	-	-	-	-	-	-
SKU110K	<b>0.1082</b>	0.0732	0.1059	0.1077	0.1066	0.089	0.1036
SFD	0.3752	0.2303	0.3901	0.3865	<b>0.3979</b>	0.3477	0.3521
SIXray	0.5574	0.4692	0.5002	0.5301	<b>0.5651</b>	0.5585	0.5609
HiXray	-	-	-	-	-	-	-
MGD	0.3874	0.3869	0.4007	0.3926	0.3916	0.3836	<b>0.4036</b>
DUO	0.3275	0.3135	0.3454	0.3662	<b>0.3691</b>	0.3636	0.335
RUOD	<b>0.2012</b>	0.1912	0.1919	0.1821	0.1905	0.1905	0.1965
UWD	0.1807	0.1812	0.1708	0.2213	<b>0.2985</b>	0.1208	0.2411
AIDD	-	-	-	-	-	-	-
EAD	-	-	-	-	-	-	-

**Table 5:** mAP<sub>small</sub> calculated on the test set for each model across all datasets. Bold values indicate best performance for each dataset. Dashes indicate missing or unavailable data.

Dataset	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
DIOR	0.5007	0.49345	0.5125	0.5082	0.5067	0.5047	0.5146
DOTA	0.3716	0.2723	0.2996	0.3577	0.368	0.3608	<b>0.3872</b>
HIT-UAV	0.5053	0.5481	<b>0.5509</b>	0.5344	0.5148	0.5019	0.5287
HoneyBee	0.6379	0.6578	0.6539	<b>0.6796</b>	0.6528	0.6717	0.6612
Pear640	0.5601	0.5224	0.5647	0.5663	<b>0.5716</b>	0.5536	0.5634
WeedCrop	<b>0.9078</b>	0.7871	0.792	0.9209	0.7842	0.9025	0.8803
BDD100K	0.2596	<b>0.3452</b>	0.2813	0.2661	0.2825	0.2825	0.2774
KITTI	0.7686	0.6867	0.7126	0.782	0.7812	0.7822	<b>0.7858</b>
TSDD	0.8404	0.8335	0.8451	<b>0.8244</b>	0.8498	0.834	0.8363
CS2	0.8589	0.8218	0.8317	<b>0.8822</b>	0.877	0.8644	0.8621
GTA5	0.7088	0.729	0.659	0.7218	0.73	0.7087	<b>0.7113</b>
MC	0.7538	0.6841	0.7333	0.7502	0.7419	0.7513	<b>0.7549</b>
DeepPCB	0.8048	0.707	0.6492	0.7867	0.784	0.7819	<b>0.8195</b>
GC10-DET	0.1394	0.1534	0.147	0.1298	0.1388	0.116	0.1345
NEU-SDD	0.3459	0.299	0.291	0.2817	0.3725	0.3349	<b>0.2917</b>
BCD	<b>0.7663</b>	0.6732	0.7113	0.7461	0.704	0.7361	0.7644
BBD	0.5515	0.566	0.592	0.401	0.6252	0.5505	0.5515
ChestX-Det	0.1032	0.0742	0.1078	0.0658	0.1071	0.0871	0.0906
GRAZPEDVRI-DX	0.4255	0.379	0.4854	0.4239	0.424	0.3544	<b>0.4846</b>
BCCD	0.6183	0.5829	0.5953	0.5898	0.6095	0.6037	0.6047
MlaMIA-SpermVideo	0.8639	0.8305	0.858	0.8675	0.875	0.8629	<b>0.8808</b>
LDD	0.2542	0.274	0.2709	0.2657	0.2704	0.2363	<b>0.2747</b>
Holoselecta	-	-	-	-	-	-	-
SKU110K	0.1632	0.1246	0.1629	0.1611	0.1614	0.1547	<b>0.1646</b>
SFD	0.8526	0.7338	0.8647	0.8665	0.8583	0.841	<b>0.8552</b>
SIXray	0.594	0.5654	0.5861	0.6046	0.6018	0.6034	<b>0.5986</b>
HiXray	0.3773	0.2388	0.4078	0.3711	0.3876	0.3732	<b>0.3789</b>
MGD	0.5627	0.5664	0.6064	0.5852	0.581	0.5768	<b>0.6142</b>
DUO	0.6839	0.6467	0.6588	0.6744	<b>0.6903</b>	0.6672	0.6786
RUOD	0.4582	0.4564	0.4711	0.4511	0.4622	0.4622	<b>0.477</b>
UWD	0.2858	0.2216	<b>0.3174</b>	0.2971	0.2846	0.2672	0.3111
AIDD	0.2957	0.1735	<b>0.3765</b>	0.2926	0.3455	0.3534	0.3196
EAD	-	-	-	-	-	-	-

**Table 6:** mAP<sub>medium</sub> calculated on the test set for each model across all datasets.

Bold values indicate best performance for each dataset. Dashes indicate missing or unavailable data.

Dataset	YOLOv5	YOLOv6	YOLOv7	YOLOv8	YOLOv9	YOLOv10	YOLOv11
DIOR	0.7771	0.74375	0.75645	0.7881	0.7928	0.7898	0.7915
DOTA	0.6105	0.4175	0.5288	0.6062	<b>0.6199</b>	0.5900	0.5869
HIT-UAV	0.7869	<b>0.7874</b>	0.8286	0.7527	0.7815	<b>0.8312</b>	0.7836
HoneyBee	0.6804	0.6698	0.6759	0.5985	0.6780	0.6679	<b>0.6858</b>
Pear640	0.8000	0.6000	0.8000	<b>0.9000</b>	0.8000	0.8500	0.8000
WeedCrop	0.9029	0.7241	0.7280	0.9070	0.7218	0.8713	<b>0.9140</b>
BDD100K	0.4091	0.4453	0.4113	0.4048	0.4215	0.4215	<b>0.4339</b>
KITTI	0.7518	0.7282	0.7350	<b>0.7630</b>	0.7631	0.7290	0.7551
TSDD	0.8751	0.8897	<b>0.8963</b>	0.8360	0.8905	0.8930	0.8886
CS2	<b>0.9505</b>	0.9219	0.9219	0.9700	0.9644	0.9436	0.9464
GTA5	0.8850	0.9015	0.8671	<b>0.9047</b>	0.9017	0.8919	0.8904
MC	0.8222	0.7518	0.7578	<b>0.8578</b>	0.8005	0.7611	0.8162
DeepPCB	-	-	-	-	-	-	-
GC10-DET	0.2819	0.2890	<b>0.3072</b>	0.2809	0.2899	0.2861	0.2960
NEU-SDD	0.4098	0.3552	<b>0.4772</b>	0.4231	0.4612	0.4435	0.4663
BCD	<b>0.8258</b>	0.8182	0.7659	0.8003	0.8116	0.7927	0.8229
BBD	0.6446	0.6680	0.6910	0.6348	<b>0.7398</b>	0.6109	0.6256
ChestX-Det	0.2961	0.2761	<b>0.3576</b>	0.2639	0.3233	0.2681	0.3214
GRAZPEDVRI-DX	0.4498	0.4760	0.4931	<b>0.4570</b>	0.4507	0.4471	0.4727
BCCD	0.4121	0.3967	<b>0.4277</b>	0.3303	0.4238	0.3834	0.3659
MlaMIA-SpermVideo	<b>1.0000</b>	0.8252	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>
LDD	0.3882	0.4457	0.4211	0.4238	0.4200	<b>0.4534</b>	0.3870
Holoselecta	0.6176	0.5765	0.6023	<b>0.6261</b>	0.5941	0.5724	0.6024
SKU110K	0.4752	0.3915	0.4823	0.4807	<b>0.4827</b>	0.4826	0.4821
SFD	0.8804	0.7819	0.8874	0.8823	0.8681	0.8712	<b>0.8876</b>
SIXray	0.7451	0.6996	0.7334	0.7545	0.7558	0.7508	<b>0.7591</b>
HiXray	0.5332	0.4983	<b>0.5423</b>	0.5261	0.5365	0.5323	0.5422
MGD	0.6218	0.6264	0.6345	0.6194	<b>0.6487</b>	0.6437	0.6106
DUO	0.6748	0.6404	0.6471	0.6710	0.6810	0.6691	<b>0.6830</b>
RUOD	0.6874	0.6710	0.6937	0.6886	0.6839	0.6839	<b>0.6961</b>
UWD	0.4514	0.4343	0.4360	0.4411	0.4432	0.4431	<b>0.4637</b>
AIDD	0.6533	0.6359	0.6719	0.6696	<b>0.6936</b>	0.6825	0.6763
EAD	0.6585	0.6865	0.6645	0.6524	0.6773	0.6329	<b>0.6876</b>

**Table 7:** mAP<sub>large</sub> calculated on the test set for each model across all datasets. Bold values indicate best performance for each dataset. Dashes indicate missing or unavailable data.