HVLO-YOLO: An Ultra-Lightweight Detection Model for High-voltage Line Obstacles

Weichao Pan[1]
202211107025@stu.sdjzu.edu.cn
Xu Wang[1]
202311102025@stu.sdjzu.edu.cn
Chengze Lv[1]
202311102015@stu.sdjzu.edu.cn
Zicheng Lin[1]
202311102026@stu.sdjzu.edu.cn
Gongrui Wang[1]
202311102051@stu.sdjzu.edu.cn
Xuening Zhang[2]
yukiZhang0527@outlook.com
Yi Sun[3]
sun-y14@ulster.ac.uk
Xingbo Liu[1*]
sclxb@mail.sdu.edu.cn

[1] School of Computer and Artificial Intelligence,
Shandong Jianzhu University,
Jinan, China

[2] School of Computer Science and Technology,
Harbin Institute of Technology,
Shenzhen, China

[3] School of Computing,
Ulster University,
Northern Ireland, United Kingdom

Abstract

With the expansion of high-voltage power grids and the increase of environmental complexity, obstacle detection on high-voltage lines has become an important task to ensure the safety of power systems. Traditional methods rely on manual feature extraction, which is difficult to deal with complex environments. Although deep learning methods improve the detection accuracy, the demand for computing resources is too high to meet the requirements of real-time and lightweight. To this end, this paper proposed an ultra-lightweight **H**igh-**V**oltage **L**ine **O**bstacle detection model **HVLO-YOLO**. To achieve a better balance between detection accuracy and computational cost, three specialized lightweight modules are introduced: (1) a CSP-Partial Convolution with Four-Group module that enhances feature extraction efficiency by selectively applying partial convolution and multi-branch group strategies; (2) a Partial Convolution DownSampler module that preserves critical information during spatial resolution reduction through a dual-branch design combining max-pooling and partial convolution; and (3) a Partial Convolution Detection Head module that focuses computational resources on key feature regions through selective lightweight aggregation. These modules collaboratively reduce computational burden, minimize parameter count, and enhance obstacle detection accuracy under complex environments. Extensive experiments conducted on two benchmark datasets demonstrate that HVLO-YOLO achieves competitive detection accuracy while

* Corresponding author: Xingbo Liu

significantly reducing model complexity compared to state-of-the-art models. Our code is publicly available at https://github.com/JEFfersusu/HVLO-YOLO.

1 Introduction

The safe and stable operation of high-voltage power grids is critical for modern society. As the power network expands and the surrounding environments become increasingly complex, high-voltage lines, serving as key infrastructure for energy transmission, face growing risks from external obstacles such as birds, flying debris, drones, and trees [9, 11]. These obstacles not only threaten the continuous supply of electricity but can also trigger severe safety accidents, posing significant risks to public security. Therefore, efficient and real-time obstacle detection on high-voltage lines, particularly through video surveillance and drone imagery, has become an urgent necessity for the power industry.

While traditional obstacle detection methods based on hand-crafted features have achieved satisfactory results in simple scenes, they often struggle in complex backgrounds, failing to generalize to the diverse and dynamic environments encountered by high-voltage lines [21]. The emergence of deep learning methods, capable of automatically extracting multi-level features, has significantly improved detection performance [1, 2, 8, 22]. However, these methods usually involve deep architectures with high computational requirements, which are difficult to achieve efficient detection requirements. Thus, achieving lightweight and accurate high-voltage line obstacle detection thus remains a pressing research challenge.

Current research efforts in high-voltage line obstacle detection can be broadly categorized based on their primary strategies for addressing key challenges such as input quality enhancement, feature representation improvement, data diversity expansion, and model efficiency optimization. Specifically, they can be grouped into four major technical approaches: (1) Image preprocessing methods [12, 18], which enhance input quality through techniques like denoising, edge detection, and contrast adjustment. While mitigating environmental noise from lighting and weather conditions, these methods rely on low-level information manipulation and struggle to generalize across complex, diverse scenarios. (2) Transformer-based detection methods [23], which model global context to overcome local feature limitations, significantly improving robustness against background clutter and varying object scales. However, their high computational and memory demands make real-time or resource-limited applications challenging. (3) Synthetic data generation methods [4, 29], which expand training diversity by simulating high-fidelity obstacle scenarios, thereby enhancing model generalization. Yet, domain gaps between synthetic and real-world distributions often limit detection performance in practice. (4) Hybrid lightweight models [26, 31], which combine strategies such as lightweight backbone design, noise reduction, and efficient feature aggregation to balance accuracy and efficiency, though aggressive simplification may still compromise detection precision.

Overall, although each existing technical line contributes to advancing obstacle detection, several fundamental challenges remain unresolved. Specifically, existing methods commonly suffer from: (1) overly complex backbone networks with redundant computations, (2) inefficient downsampling structures that lead to feature degradation, and (3) detection heads with high computational overhead that impede real-time detection. These bottlenecks collectively hinder the achievement of highly accurate and efficient obstacle detection under practical resource constraints.

To address these challenges, this paper proposes HVLO-YOLO, an ultra-lightweight object detection framework specifically tailored for high-voltage line obstacle scenarios. HVLO-YOLO introduces three novel modules: (1) the **C**SP-**P**artial Convolution with **Four** Group **(CP4)** module, which enhances feature extraction under lightweight conditions through partial and group convolutions; (2) the **P**artial Convolution **Down**Sampler **(PDown)** module, which achieves efficient downsampling via parallel branches while preserving semantic information; and (3) the **P**artial Convolution **Detect**ion Head **(PDetect)** module, a selective lightweight detection head that reduces computational burden without sacrificing detection accuracy. Each component directly targets the bottlenecks identified in previous methods, ensuring a balanced trade-off between model efficiency and detection precision.

In addition to mainstream YOLO-based methods, prior works such as ShuffleNet [14] and GhostNet [13] have explored similar efficiency strategies through channel shuffling, group convolution, and ghost feature generation. Compared to these, our proposed CP4 module introduces recursive partial convolution [5] with group-based selective processing, enabling progressive feature enrichment under strict computational budgets. Unlike Ghost modules that synthesize features using linear operations, CP4 leverages spatial convolutional context while minimizing redundancy. Similarly, while PDown and PDetect also follow divide-and-conquer principles, they are specifically designed to address the unique needs of spatial resolution and localized detection under real-time constraints.

The overall contributions of this paper are as follows:

- To enhance feature extraction under strict efficiency constraints, we propose the CP4 module, which integrates Partial Convolution with a FourGroup strategy for better lightweight representation.
- The PDown module adopts a parallel path design to accelerate downsampling, effectively feature preservation and computational efficiency.
- For the detection head, we design PDetect, a streamlined yet powerful structure that maintains high detection accuracy while significantly lowering resource consumption.
- Comprehensive experiments on both the High-Voltage Line Obstacle dataset and COCO validate HVLO-YOLO's real-time performance and robust generalization across different scenarios.

2 Method

This section presents the architecture of HVLO-YOLO and details the design of its three core lightweight modules: CP4, PDown, and PDetect. HVLO-YOLO follows a classic three-stage object detection pipeline, *i.e.*, Backbone for feature extraction, Neck for multi-scale feature fusion, and Head for prediction, where each stage is equipped with a specialized lightweight module to address specific challenges. Specifically, CP4 enhances efficient feature extraction under resource constraints, PDown preserves critical semantic information during downsampling, and PDetect enables precise and lightweight object prediction. The overall framework is illustrated in Figure 1.

2.1 CSP-Partial Convolution with FourGroup (CP4) Module

Efficient feature extraction is critical for lightweight detectors. Conventional convolutional backbones either excessively reduce model parameters at the expense of feature richness

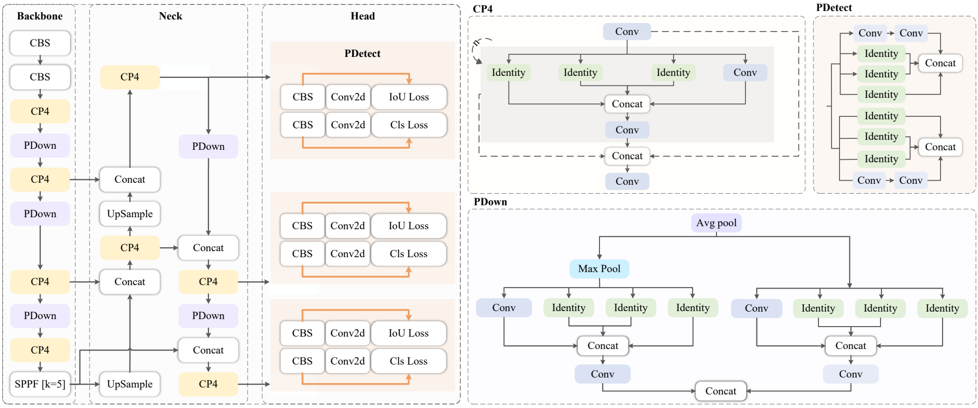

Figure 1: Overall framework of HVLO-YOLO. CP4 enhances feature extraction in Backbone and Neck, PDown enables efficient downsampling in Backbone and Neck, and PDetect improves lightweight detection in Head, achieving accurate high-voltage obstacle detection with low computational cost.

or rely on heavy modules that burden computational cost. To address this trade-off, we propose CP4, aiming to maximize feature diversity while maintaining lightweight characteristics through a selective and recursive convolution mechanism.

As illustrated in Figure 1, given an input feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, CP4 first evenly divides it into four groups: one group $\mathbf{X}^{(1)}$ is processed by a 3×3 convolution to enhance local spatial information, while the other three groups $\mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \mathbf{X}^{(4)}$ are directly preserved. The outputs are concatenated and passed through a 1×1 convolution for feature fusion:

$$\mathbf{Y}^{(1)} = \mathbf{W}_1 \times \text{ Concat}\left(\mathbf{W}_3 \times \ \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \mathbf{X}^{(4)}\right). \tag{1}$$

where $\mathbf{W}_3$ and $\mathbf{W}_1$ denote the 3×3 and 1×1 convolutional kernels, respectively. To further enhance the feature hierarchy, CP4 recursively applies a split-selective convolution mechanism at each iteration $t \geq 2$:

$$\mathbf{Y}^{(t)} = \mathbf{W}_1^{(t)} \times \text{ Concat}\left(\mathbf{Y}^{(t-1)}, \text{SplitConv}\left(\mathbf{Y}^{(t-1)}\right)\right), \tag{2}$$

where $\text{SplitConv}(\cdot)$ denotes a selective split-and-convolve operation that divides the input feature map into multiple channel groups, applies convolution to a subset of them, and preserves the rest through identity mapping. The outputs are then concatenated along the channel dimension.

Through recursive partial processing and multi-branch fusion, CP4 progressively enriches semantic features at different scales while keeping the number of parameters and operations low. This design ensures that the Backbone maintains both efficiency and representational power, providing a solid foundation for downstream detection tasks in resource-constrained environments.

2.2 Partial Convolution DownSampler (PDown) Module

Downsampling plays a vital role in reducing feature map resolution and expanding receptive fields. However, independent conventional downsampling methods such as max pooling or strided convolution often cause severe semantic information loss, especially harming small object detection and degrading feature integrity. To alleviate this, we design PDown to perform resolution reduction while preserving critical semantic details through a dual-branch selective processing mechanism.

Given the input feature map $\mathbf{X}$, PDown first applies average pooling to obtain a smooth expression version:

$$\mathbf{X}_{\text{avg}} = \text{AvgPool}(\mathbf{X}). \tag{3}$$

Two parallel paths are then constructed: The first path performs max pooling followed by partial convolution:

$$\mathbf{Y}_{\text{path1}} = \mathbf{W}_1 \times \text{ Concat}\left(\mathbf{W}_3 \times \mathbf{X}_{\text{max}}^{(1)}, \mathbf{X}_{\text{max}}^{(2)}, \dots\right), \tag{4}$$

where $\mathbf{X}_{\text{max}} = \text{MaxPool}(\mathbf{X}_{\text{avg}})$. The second path applies partial convolution directly to the average-pooled features:

$$\mathbf{Y}_{\text{path2}} = \mathbf{W}_1 \times \text{ Concat}\left(\mathbf{W}_3 \times \mathbf{X}_{\text{avg}}^{(1)}, \mathbf{X}_{\text{avg}}^{(2)}, \dots\right). \tag{5}$$

Finally, outputs from the two paths are concatenated:

$$\mathbf{Y} = \text{Concat}\left(\mathbf{Y}_{\text{path1}}, \mathbf{Y}_{\text{path2}}\right). \tag{6}$$

By combining global contextual smoothing and localized strong responses through dual-path selective processing, PDown effectively maintains semantic richness while achieving efficient downsampling. This design mitigates feature degradation during spatial resolution reduction, enabling better information flow into subsequent detection stages.

2.3 Partial Convolution Detection Head (PDetect) Module

The detection head is critical for precise localization and classification. However, traditional dense prediction heads often involve heavy convolutional operations. These operations increase computational cost and hinder deployment in resource-constrained environments. Moreover, uniformly processing all spatial regions ignores the fact that only a small portion of regions are crucial for detection. To tackle this, we propose PDetect, a selective lightweight head that focuses computational resources on important regions while reducing redundant processing.

Given an input feature map $\mathbf{X}$, PDetect first splits it into multiple branches: (1) The top and bottom branches apply lightweight 3×3 convolutions to efficiently enhance key features. (2)The middle branches are passed through via identity mapping to save computation. Each local aggregation is computed as:

$$\mathbf{Y}_{\text{local}}^{(i)} = \text{Concat}\left(\mathbf{W}_3 \times \mathbf{X}^{(i)}, \mathbf{X}^{(i+1)}, \mathbf{X}^{(i+2)}, \dots\right), \tag{7}$$

All branches are then concatenated and fused via a convolution to produce the final output:

$$\mathbf{Y} = \mathbf{W}_1 \times \text{ Concat}\left(\mathbf{Y}_{\text{local}}^{(1)}, \mathbf{Y}_{\text{local}}^{(2)}, \dots\right). \tag{8}$$

By selectively applying convolution to key regions and preserving others through identity mapping, PDetect drastically reduces computational burden while maintaining detection accuracy. This design enables HVLO-YOLO to achieve real-time object detection under strict resource constraints without compromising precision.

2.4 Loss Function

For bounding box regression, we use the Complete IoU (CIoU) loss [30]:

$$\mathcal{L}_{\text{CIoU}} = 1 - \left(\text{IoU} - \frac{\rho^2(\mathbf{b}, \mathbf{b}^{\text{gt}})}{c^2} - \alpha v \right), \tag{9}$$

where $\rho(\cdot)$ is the Euclidean distance between box centers, and c is the diagonal length of the minimum enclosing box. The aspect ratio term v and its weight α are:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{w}{h} \right)^2, \quad \alpha = \frac{v}{(1 - \text{IoU}) + v}, \tag{10}$$

where (w, h) and $(w^{\text{gt}}, h^{\text{gt}})$ denote the predicted and ground-truth box sizes.

For classification, we apply the cross-entropy loss [16]:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{11}$$

where C is the number of classes, y_i is the one-hot label, and $\hat{y}_i$ is the predicted probability.

3 Experiments

In this section, a brief overview of the experimental datasets and setup is first provided. Next, experimental results and analyses will be introduced in turn.

3.1 Datasets

The High-Voltage Line Obstacle Dataset contains approximately 1300 images annotated with precise bounding boxes. The obstacles are categorized into four types: nests, kites, garbage, and balloons. The dataset maintains a consistent training/testing split across all categories and provides sufficient diversity to support effective model training and evaluation.

COCO [10] dataset is a key computer vision benchmark widely used for the training and validation of object detection models. The dataset provides more than 200,000 meticulously annotated images, divided into training sets, validation sets, and test sets, covering 80 object classes and more than 250,000 object instances.

3.2 Experimental Setup

Experiments were conducted on a Windows 11 system with a single NVIDIA RTX 3080 Ti GPU. PyTorch 2.0.1 with CUDA 11.8 and Python 3.8 were used. All models were trained under identical settings with images resized to 640×640×3, a batch size of 8, and 300 epochs.

To comprehensively evaluate the detection performance of the proposed HVLO-YOLO model for high-voltage line obstacle scenarios, comparative experiments were conducted against a series of advanced object detection models, these include YOLOv8-n [7], YOLOv8-s, YOLOv8-m, YOLOv9-t [25], YOLOv9-s, YOLOv9-m, YOLOv10-n [24], YOLOv10-s,

https://aistudio.baidu.com/datasetdetail/223069

YOLOv10-m, YOLOv10-b, YOLOv10-l, YOLOv10-x, RT-DETR-l [28], RT-DETR-x, RT-DETR-ResNet50, and RT-DETR-ResNet101.

In these evaluations, four core detection metrics were adopted: precision (P), recall (R), mAP50, and mAP50:95 [20]. To assess the model's efficiency, the number of parameters (Params), floating point operations (FLOPs), and model size were also measured.

3.3 Experimental Results and Discussion

In order to verify the superior performance of the HVLO-YOLO object detection model proposed in this paper, a series of validations were conducted on the above two datasets, and several evaluation metrics mentioned above were used for evaluation and analyses.

Specifically, this section first conducts a comparison experiment with the proposed model HVLO-YOLO, proving the superiority and high efficiency of the proposed model. Then the results of the proposed model are evaluated, including the analysis of model recognition results analysis and confusion matrix. Then the universal version of the HVLO-YOLO model is verified on the COCO dataset. Finally, the effectiveness of the three lightweight modules proposed in this paper is verified by ablation study.

3.3.1 Performance Analysis

As shown in Table 1, the proposed HVLO-YOLO achieves a mAP50 of 89.9% and mAP50:95 of 67.6%, while maintaining an extremely lightweight design with only 1.0M parameters, 2.3 GFLOPs, and a model size of 2.2 MB.

Compared with the YOLOv8 series, HVLO-YOLO achieves higher mAP than YOLOv8 series, while using less than one-third the parameters and significantly fewer FLOPs. For instance, HVLO-YOLO requires only 2.3 GFLOPs, compared to 8.1 GFLOPs for YOLOv8-n and 28.4 GFLOPs for YOLOv8-s, making it much more suitable for real-time applications on resource-constrained devices.

Table 1: Detailed comparative results of the contrast experiments. Red font represents baseline models. The rise and fall of the metric is compared to the best baseline models. The bolded values in each column represent the best performance.

Model	P (%)	R (%)	mAP50 (%)	mAP 50:95 (%)	Params (M)	FLOPs (G)	Model Size (MB)
YOLOv8-n	95.7	76.6	84.6	63.1	3.0	8.1	6.3
YOLOv8-s	90.1	81.5	86.8	65.2	11.13	28.4	22.5
YOLOv8-m	92.7	78.0	85.1	65.4	25.84	78.7	52.0
YOLOv9-t	97.0	77.5	88.4	67.1	2.0	7.6	4.7
YOLOv9-s	96.9	80.6	88.4	70.2	6.0	20.8	12.9
YOLOv9-m	96.6	82.0	88.8	70.8	20.0	76.5	40.8
YOLOv10-n	84.7	77.1	82.8	62.8	2.7	8.2	5.8
YOLOv10-s	90.4	80.7	85.3	67.2	7.3	21.2	15.1
YOLOv10-m	89.0	78.7	86.2	67.6	15.7	59.5	31.9
YOLOv10-b	91.2	83.0	88.2	70.9	19.6	93.6	39.8
YOLOv10-l	95.1	77.8	87.4	69.6	24.9	122.0	50.6
YOLOv10-x	**97.1**	80.3	89.8	**71.5**	31.6	169.8	64.1
RT-DETR-l	88.5	**85.9**	87.3	69.7	32.0	103.4	66.2
RT-DETR-x	96.5	77.9	84.5	66.8	65.5	222.5	135.4
RT-DETR-ResNet50	91.9	84.9	87.4	69.6	41.9	125.6	86.0
RT-DETR-ResNet101	92.0	84.1	85.1	69.4	60.9	186.2	124.2
HVLO-YOLO (ours)	91.7 (↓5.3)	82.9 (↑5.4)	**89.9** (↑1.5)	67.6 (↑0.5)	**1.0** (↓1.0)	**2.3** (↓5.3)	**2.2** (↓2.5)

When compared with YOLOv9 and YOLOv10, HVLO-YOLO offers a compelling balance between performance and efficiency. It outperforms YOLOv9-t and YOLOv10-n in mAP50, while using fewer parameters and lower FLOPs. Notably, compared to the state-of-the-art real-time model YOLOv10-x, HVLO-YOLO achieves a comparable mAP50 (89.9% vs. 89.8%), but with 31.6 times fewer parameters and 73.8 times fewer FLOPs.

These results highlight that HVLO-YOLO, through its integrated lightweight components, CP4 for efficient feature extraction, PDown for fast downsampling, and PDetect for streamlined prediction, achieves excellent detection accuracy with ultra-low computational cost, making it well-suited for real-time high-voltage line obstacle detection in real-world, resource-limited scenarios.

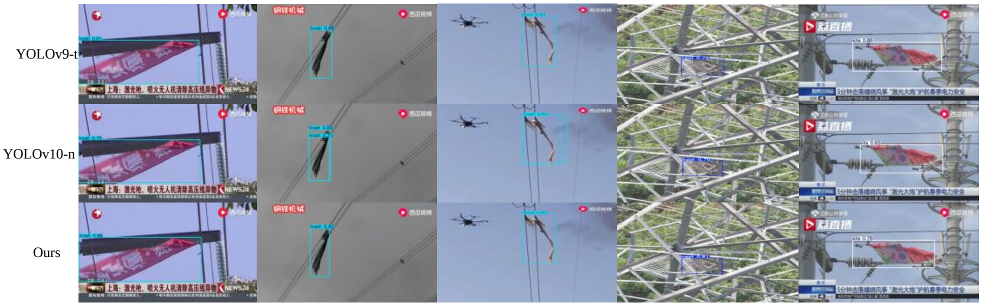

Figure 2: Comparison with the detection results of the baseline models.

As shown in Figure 2, the detection results comparison highlights the superiority of the proposed HVLO-YOLO model over YOLOv9-t and YOLOv10-n. Across various challenging scenarios, such as dense power lines, background clutter, and object occlusion, HVLO-YOLO consistently produces more accurate and complete bounding boxes, while significantly reducing false detections and missed objects. This improvement is attributed to the CP4 module's ability to enhance feature representation under lightweight constraints, the PDown module's effective multi-path downsampling that preserves critical information, and the PDetect module's selective convolution design that focuses computational effort on important regions. Together, these modules enable HVLO-YOLO to maintain high detection precision and robustness even in complex and noisy environments.

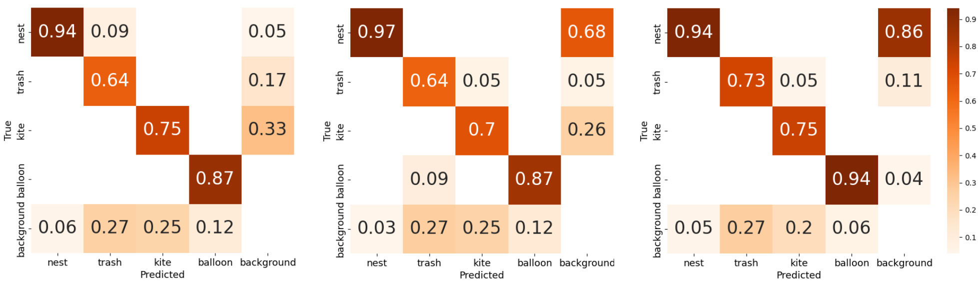

Figure 3: Comparison with the confusion matrix of the baseline models. From left to right are YOLOv9-t, YOLOv10-n and our model.

As shown in Figure 3, compared to YOLOv9-t and YOLOv10-n, our model achieves higher accuracy across most categories, especially for "balloon" (0.94) and "trash" (0.73), while significantly reducing misclassifications such as confusing "kite" with "trash". Overall, it demonstrates more balanced and robust performance in distinguishing between similar and background classes.

3.3.2 Universal Object Detection Experiment

In order to validate the performance of our proposed ultra-lightweight model HVLO-YOLO for generic object detection, this paper compared it with advanced and popular lightweight models on COCO, and the results are shown in the above Table 2.

Table 2: Detailed comparison of models for universal object detection.

Model	**mAP50 (%)**	**mAP 50:95 (%)**	**Params (M)**	**Model Size (MB)**
NanoDet-m [19]	36.5	21.6	0.95	1.8
YOLO-Fastest-1.1-xl [6]	35.4	20.8	0.93	3.7
YOLOv4-t [3]	38.0	23.2	6.1	23.2
MobileDets [27]	42.8	28.1	4.9	13.8
Mobile-ViT-SSDLite [17]	41.1	26.3	2.7	6.3
FastestDet [15]	28.4	14.6	**0.3**	**1.2**
YOLOv5-n [7]	49.1	28.4	1.9	3.9
YOLOv8-n [7]	51.9	37.3	3.2	6.2
YOLOv9-t [25]	52.2	38.3	2.0	4.4
YOLOv10-n [24]	**53.2**	**38.5**	2.3	5.7
HVLO-YOLO (ours)	39.3	25.5	1.0	2.3

The results show that the HVLO-YOLO model achieves 39.3% mAP50 and 25.5% mAP50:95. Compared with other popular ultralight models, such as NanoDet-m, YOLO-Fastest-1.1-xl, and MobileDets, the HVLO-YOLO model demonstrates not only superior mAP performance but also notable advantages in terms of model size and parameter efficiency. Specifically, it requires only 1.0M parameters and has a compact model size of 2.3MB, highlighting its efficiency in comparison to other models. This excellent balance of performance and efficiency makes the HVLO-YOLO model an ideal choice for resource-constrained environments that require fast and accurate object detection.

3.3.3 Ablation Study

In this ablation study, we evaluated the specific impact of the three core modules, *i.e.*, CP4, PDown, and PDetect, on the object detection model's performance, both individually and in various combinations. By analyzing key metrics such as precision, recall, and mAP (mAP50 and mAP50:95) at different IoU thresholds, the results are shown in Table 3.

Table 3: Detailed results of ablation study. The baseline model is YOLOv8-n.

CP4	**PDown**	**PDetect**	**P (%)**	**R (%)**	**mAP50 (%)**	**mAP50:95 (%)**	**Params (M)**	**FLOPs (G)**	**Model Size (MB)**
			95.7	76.6	84.6	63.1	3.0	8.1	6.3
✓			97.2	82.8	87.0	67.1	2.12	5.9	4.5
✓	✓		89.6	82.3	89.1	65.6	1.04	2.6	2.3
✓		✓	**98.4**	80.2	84.6	64.6	1.54	3.3	3.3
	✓		94.4	**83.1**	88.5	**69.4**	2.47	7.1	5.2
	✓	✓	92.8	80.6	88.1	66.5	1.89	4.4	4.0
		✓	96.7	77.9	86.2	66.1	2.42	5.5	5.1
✓	✓	✓	91.7	82.9	**89.9**	67.6	**1.0**	**2.3**	**2.2**

The ablation study reveals distinct contributions from each module. The baseline YOLOv8-n achieves solid performance, but with higher complexity. Introducing CP4 alone significantly enhances precision and mAP while reducing parameters, showing its strength in feature extraction. Adding PDown to CP4 further improves mAP50 to 89.1% with major efficiency gains, though precision drops slightly. Combining CP4 + PDetect yields the highest precision, indicating PDetect's role in precise localization, despite lower recall. PDown alone achieves the best recall and highest mAP50:95, excelling in scale-aware detection. PDown + PDetect maintains solid overall accuracy, balancing recall and precision. PDetect alone boosts precision but lags in recall and mAP. Ultimately, the full combination of CP4 + PDown + PDetect delivers the best trade-off, achieving 89.9% mAP50 with only 1.0M parameters and 2.2MB model size, demonstrating strong performance and compactness.

3.3.4 Error Analysis

During the experiment, although the HVLO-YOLO model showed good performance in the detection of high-voltage line obstacles, it still had some limitations. Firstly, the interference of complex background will have a certain impact on the detection effect, the model is prone to false detection or missed detection. Secondly, despite the multi-scale feature fusion strategy, the detection ability of small objects at long distances still needs to be further improved. The class imbalance problem in the dataset also affects the detection effect of some categories, resulting in the model's high detection accuracy for "bird's nest" obstacles, but its detection performance for "garbage" and "balloon" obstacles is slightly weak. Finally, the model is not robust enough under extreme lighting or weather conditions, and the accuracy of object detection is reduced under complex lighting conditions.

4 Conclusion

This paper proposed HVLO-YOLO model, an ultra-lightweight object detection framework for efficiently detecting obstacles on high-voltage lines. By introducing three specialized lightweight modules, *i.e.*, CP4, PDown, and PDetect, the proposed model achieves significant computational reduction while maintaining high detection accuracy. Specifically, the CP4 module enhances feature extraction efficiency through partial convolution and Four-Group parallelization, effectively capturing multi-scale features with minimal computation. The PDown module employs a dual-branch downsampling strategy that combines max pooling and partial convolution to preserve key feature information during resolution reduction. Meanwhile, the PDetect module selectively aggregates important spatial features in the detection head, focusing computational resources on critical regions and improving detection precision without increasing model size. Experimental results show that HVLO-YOLO is superior to the existing models in terms of detection performance and computational efficiency, and is suitable for real-time, resource-constrained high-voltage line obstacle detection applications. Furthermore, the generalized obstacle detection task on the COCO dataset shows a significant reduction in model size and computational complexity while maintaining high mAP values.

Despite its efficiency, HVLO-YOLO still faces challenges in handling complex textures and tiny objects, mainly due to its fixed partial convolution design. Future work will explore adaptive partial convolutions and dynamic feature selection to improve robustness. Additionally, the limited dataset size restricts generalizability. Evaluating the model on larger, real-world datasets will be essential to validate its scalability and reliability.

5 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62206160), Young Talent of Lifting Engineering for Science and Technology in Shandong (SDAST2024QTA020), the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions (2024KJH084), Shandong Provincial Natural Science Foundation (ZR2022QF082), and special funds for distinguished professors of Shandong Jianzhu University.

References

[1] Shams Forruque Ahmed, Md. Sakib Bin Alam, Maliha Kabir, Shaila Afrin, Sabiha Jannat Rafa, Aanushka Mehjabin, and Amir H. Gandomi. Unveiling the frontiers of deep learning: Innovations shaping diverse domains. *Applied Intelligence*, 55(7):573, 2025. ISSN 1573-7497. doi: 10.1007/s10489-025-06259-x.

[2] R. Archana and P. S. Eliahim Jeevaraj. Deep learning models for digital image processing: a review. *Artificial Intelligence Review*, 57(1):11, 2024. ISSN 1573-7462. doi: 10.1007/s10462-023-10631-z.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] Guangqing Chen, Shikai Wang, Yongkang Liu, Jinju Li, Gaobin Qin, Jidai Wang, Aiqin Sun, and Liang Yuan. Research on transmission line hardware identification based on improved yolov5 and deblurganv2. *IEEE Access*, 11:133351–133362, 2023. doi: 10.1109/ACCESS.2023.3336905.

[5] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S.-H. Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12021–12031, 2023. doi: 10.1109/CVPR52729.2023.01157.

[6] dog qiuqiu. dog-qiuqiu/Yolo-Fastest: yolo-fastest-v1.1.0. Zenodo, July 2021.

[7] G. Jocher, J. Qiu, and A. Chaurasia. Ultralytics yolo (version 8.0.0) [computer software], 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[8] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2023. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2022.103812.

[9] Z. Li, Q. Wang, T. Zhang, C. Ju, S. Suzuki, and A. Namiki. Uav high-voltage power transmission line autonomous correction inspection system based on object detection. *IEEE Sensors Journal*, 23(9):10215–10230, 2023.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48.

[11] Li Lirong, Cui Hao, Dai Junwei, Mei Bing, Zang Zhijie, He Zhangqin, and Li Jie. High-voltage transmission line inspection based on multi-scale lightweight convolution. *Measurement*, 251:117193, 2025. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2025.117193.

[12] Gao Liu, Duanjiao Li, Wenxing Sun, Zhuojun Xie, Ruchao Liao, and Jiangbo Feng. An obstacle avoidance safety detection algorithm for power lines combining binocular vision technology and improved object detection. *Energy Informatics*, 7(1):72, 2024. ISSN 2520-8942. doi: 10.1186/s42162-024-00378-4.

[13] Zhenhua Liu, Zhiwei Hao, Kai Han, Yehui Tang, and Yunhe Wang. Ghostnetv3: Exploring the training strategies for compact models. *arXiv preprint arXiv:2404.11202*, 2024.

[14] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *arXiv preprint arXiv:1807.11164*, 2018.

[15] X. Ma. Fastestdet: Ultra lightweight anchor-free real-time object detection algorithm, 2022. [Online]. Available: https://github.com/dog-qiuqiu/FastestDet.

[16] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23803–23828. PMLR, 23–29 Jul 2023.

[17] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer, 2022.

[18] Weichao Pan, Wenqing Huan, and Lixian Xu. Improving high-voltage line obstacle detection with multi-scale feature fusion in yolo algorithm. In *2024 6th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT)*, pages 270–273, 2024. doi: 10.1109/ECNCT63103.2024.10704270.

[19] RangiLyu. Nanodet-plus: Super fast and high accuracy lightweight anchor-free object detection model, 2021. [Online]. Available: https://github.com/RangiLyu/nanodet.

[20] Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, Hong Yan, and Manoj Karkee. Yolo advances to its genesis: a decadal and comprehensive review of the you only look once (yolo) series. *Artificial Intelligence Review*, 58(9):274, 2025. ISSN 1573-7462. doi: 10.1007/s10462-025-11253-3.

[21] Pratibha Sharma, Sumeet Saurav, and Sanjay Singh. Object detection in power line infrastructure: A review of the challenges and solutions. *Engineering Applications of Artificial Intelligence*, 130:107781, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.107781.

[22] Yibo Sun, Zhe Sun, and Weitong Chen. The evolution of object detection methods. *Engineering Applications of Artificial Intelligence*, 133:108458, 2024. ISSN 0952-1976. doi: 10.1016/j.engappai.2024.108458.

[23] Mei Tao and Zheng Gengsheng. Bird's nest detection in power transmission lines with fusion of attention and multi-scale features. In *Proceedings of the 2023 6th International Conference on Robot Systems and Applications*, ICRSA '23, page 49–55, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400708039. doi: 10.1145/3655532.3655542.

[24] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 107984–108011. Curran Associates, Inc., 2024.

[25] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 1–21, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72751-1. doi: 10.1007/978-3-031-72751-1_1.

[26] Yunjie Xiang, Congliu Du, Yan Mei, Liang Zhang, Yutong Du, and Aoxing Liu. Bn-yolo: a lightweight method for bird's nest detection on transmission lines. *Journal of Real-Time Image Processing*, 21(6):194, 2024. ISSN 1861-8219. doi: 10.1007/s11554-024-01577-8.

[27] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3824–3833, 2021. doi: 10.1109/CVPR46437.2021.00382.

[28] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16965–16974, 2024. doi: 10.1109/CVPR52733.2024.01605.

[29] Z. Zheng, M. Zhao, X. Cheng, Z. Weng, and Y. Wang. Defect detection for high voltage transmission lines based on deep learning. In S. Yu et al., editors, *Pattern Recognition and Computer Vision. PRCV 2022*, volume 13537 of *Lecture Notes in Computer Science*, pages 552–563. Springer, Cham, 2022. doi: 10.1007/978-3-031-18916-6_46.

[30] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 52(8): 8574–8586, 2022. doi: 10.1109/TCYB.2021.3095305.

[31] C. Zhu, Y. Zhao, T. Li, et al. Lfe-yolo: An algorithm suitable for recognizing small target in blurred high-voltage line equipment. *Journal of Real-Time Image Processing*, 22:107, 2025. doi: 10.1007/s11554-025-01686-y.