2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

A Lightweight Detection Network inspired by the Olfactory Learning Circuit of Caenorhabditis elegans

Jiangpeng Zheng[1,2][0009-0007-7644-6083], Jiacheng Zhao[1,2][0009-0001-1174-2945], Xuebin Wang[3][0000-0002-3095-4417], Meng Zhao[1,2][0000-0002-5060-9223], Fan Shi[1,2][0000-0003-2074-0228], and He Liu[3][0000-0001-9418-9171]

[1] Tianjin University of Technology, Tianjin 300384, China
[2] Key Laboratory of Computer Vision and System (Ministry of Education), Tianjin University of Technology, Tianjin, China
[3] Department of Systems Science, aculty of Arts and Sciences, Beijing Normal 6 University, Beijing 100875, China

zjp_1997@stud.tjut.edu.cn, zjc19@stud.tjut.edu.cn, wangxb@mail.bnu.edu.cn, zh_m@tju.edu.cn, shifan@email.tjut.edu.cn, heliu@bnu.edu.cn

Abstract. Real-time detection models often struggle in resource-constrained environments due to high computational demands. Inspired by the neural circuitry of Caenorhabditis elegans (C. elegans), which efficiently detects and localizes aversive odors through a simple yet effective hierarchical structure, we redesign the YOLOv8 architecture to enhance its efficiency. The proposed model incorporates a biologically inspired neural circuit structure to strengthen feature extraction and edge exploration. Additionally, we introduce an AdaptiveAdd module, modeled after synaptic connectivity, which dynamically adjusts feature fusion to emphasize critical information. Evaluated on the frontier and object datasets, our model achieves comparable accuracy to YOLOv8 while reducing parameters by 13.9%, demonstrating its suitability for real-time applications on edge devices.

Keywords: Caenorhabditis elegans, Neural circuits, Artificial neural networks, Object detection, Lightweight network.

1 Introduction

In recent years, real-time visual object detection [1,2,3] has become a critical task in fields such as autonomous driving [4,5,6], security monitoring [7], and robotics [8, 9, 10]. However, in resource-constrained environments, particularly on embedded platforms and mobile devices, deep learning models—especially those designed for object detection—often face significant computational efficiency bottlenecks. These bottlenecks lead to high computational costs and excessive resource consumption. Consequently, reducing computational requirements while maintaining high performance has become a core challenge for these applications.

Traditional object detection frameworks, such as the YOLO (You Only Look Once) family [11], perform well in real-time tasks but are often difficult to deploy in resource-constrained embedded systems due to their high computational cost and heavy memory and processing demands. To address this issue, researchers have proposed various lightweight methods, including model pruning [12], quantization [13], knowledge distillation [14], and low-rank decomposition [15]. These methods improve efficiency by reducing the number of model parameters, optimizing computational processes, or simplifying the network structure. However, they often suffer from instability, reliance on empirical expertise, and limitations in generalization and interpretability. In contrast to the manual and experience-dependent design of network architectures, an approach known as neural architecture search (NAS) [16,17] and evolutionary algorithms offer new ways to automate the design of efficient network architectures. These methods identify optimal structures within the architecture space via search mechanisms, although they typically incur high computational costs.

In nature, biological systems are capable of accomplishing complex tasks with remarkable efficiency in resource utilization. Neurons and their connections in biological neural networks are formed through chemical synapses or gap junctions, enabling efficient computation despite limited resources. In recent years, there has been growing interest in enhancing the computational efficiency of deep learning models by simulating the structure of biological neural networks [18,19,20,21]. For instance, Changjoo Park et al. [22] constructed a directed acyclic graph (DAG) based on the connectome of the Caenorhabditis elegans nematode to perform image classification. Another study [23] improved deep learning systems by modeling the connectome topology of the Hidradius crypticus nematode. Additionally, research has demonstrated that a neural controller with 19 control neurons and 253 synapses can effectively map high-dimensional inputs to outputs such as steering commands. This compact system provides an efficient solution for self-driving cars performing specific tasks.

While most existing studies have focused on image classification tasks, the application of biological connectome topology to target detection tasks remains unexplored. At the same time, research by [24] has shown that neural connectivity rules in the Drosophila optic lobe are crucial for understanding its visual processing mechanisms. This insight led the hypothesis that neural structure from other animal model, such as C. elegans, can be applied for inspiring AI architecture for visual tasks such as imaging detection, etc. Further studies have demonstrated the topological similarity of neural circuits across different species, providing a foundation for applying neural network principles across species. Furthermore, [25] investigates how the multilayer structure of the C. elegans connectome, including the connectivity between neurons within the same layer, impacts the overall behavior of the network. Inspired by these findings, we propose to apply the Olfactory Learning Circuit of C. elegans to the object detection model.

The proposed C. elegans YOLOv8 (CEYOLOv8) framework is illustrated in Fig. 1. This model addresses computational challenges in resource-constrained environments while introducing a biologically inspired attention mechanism. By leveraging the simple yet efficient neural architecture of Caenorhabditis elegans, it reduces parameter count without sacrificing detection accuracy.The main contributions are summarized as follows:

- Our contribution is inspired by the hierarchical neural Olfactory Learning Circuit of Caenorhabditis elegans, which enables rapid and efficient integration, computation, and execution of neural information. By mimicking thisunique neural connectivity structure, we propose a lightweight object detection backbone that reduces the parameter count of the YOLOv8 model. This results in enhanced computational efficiency, making the model more suitable for deployment in resource-constrained environments.
- We propose the AdaptiveAdd module, a biologically inspired feature fusion mechanism that mimics synaptic weight dynamics in biological neural networks. This module dynamically adjusts fusion weights between neuron outputs during training, enabling context-aware integration of multi-node features. Compared to traditional fusion strategies, it streamlines inter-layer information flow while maintaining detection accuracy, offering a principled approach to bridge biological neural efficiency with deep learning model design.
- Through extensive experiments on frontier and object datasets, we demonstrate CEYOLOv8's robustness under computational constraints. Ablation studies further validate that our bio-inspired design reduces reliance on empirical architecture tuning, offering a scalable solution for real-time applications such as autonomous systems and edge devices.

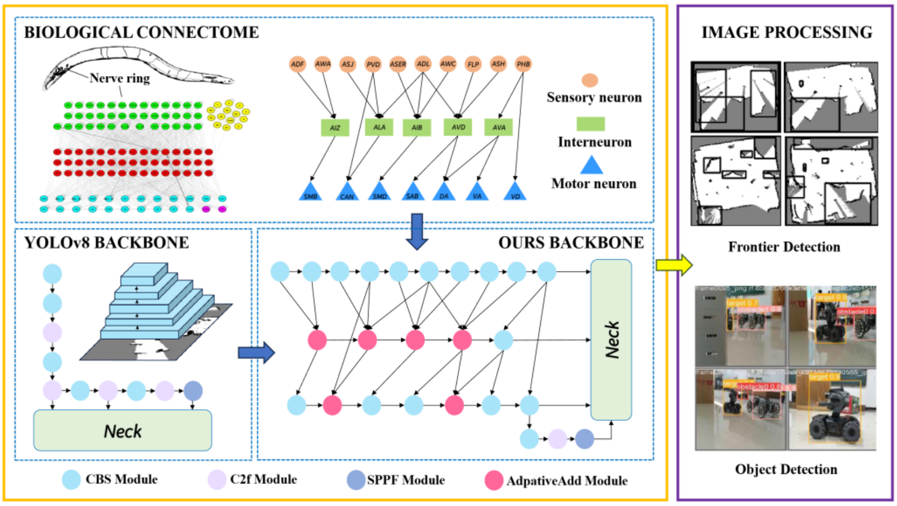

Fig. 1. General framework of this work. Inspired by the functional neural circuits of aversive olfactory learning in C. elegans,a new lightweight YOLO architecture is designed by obtaining its Olfactory LearningCircuit.

2 Related Work

2.1 LightWeight YOLOv8 Framework

Significant progress has been made in target detection models, with a focus on improving accuracy while reducing computational costs. Among them, the YOLO series stands out for its real-time performance and has been widely adopted across various domains due to its speed and precision.

With the rise of embedded and mobile platforms, recent research has prioritized optimizing YOLO for resource-constrained devices by minimizing model size and enhancing computational efficiency without sacrificing accuracy. A variety of techniques have emerged to support edge deployment, among which Neural Architecture Search (NAS) [26,27] has gained traction. NAS automates the search for optimal network structures, improving both design efficiency and model performance. For example, YOLO-NAS [28] integrates NAS into the YOLOv8 framework to discover suitable architectures. While its accuracy is slightly lower than standard YOLOv8, it significantly reduces model parameters, improving runtime efficiency.

However, NAS also presents challenges, including high computational cost, expansive search space, slow convergence, and susceptibility to overfitting, limiting its practical deployment. Thus, efficient and lightweight architecture design remains an ongoing challenge. Beyond NAS, other efforts have targeted YOLO optimization. Lai et al. [29] proposed YOLOv8-lite, an interpretable lightweight model for UAV-based detection, which integrates depthwise separable convolutions and a P2 layer in the neck to reduce computational overhead while improving performance. Similarly, Di et al. [30] introduced an attention-enhanced lightweight model for small pest detection, significantly boosting small-target accuracy through improved feature extraction. Nonetheless, effective lightweighting of the backbone network remains an unresolved issue.

2.2 C.elegans Neural Network Developmental Olfactory Learning Circuit

In the field of biologically inspired neural network design [31, 32], researchers have sought to develop more streamlined and efficient architectures by mimicking the neural connectivity patterns of biological systems, particularly for complex tasks. Beyond modeling information processing, these efforts aim to optimize artificial intelligence models by leveraging biological insights. For instance, Manglani et al. [33] applied brain connectome-based models to predict cognitive impairments, while Bardozzo et al. [34] utilized the linker topology of Caenorhabditis elegans to design deep learning and reservoir structures for classification tasks.

These studies demonstrate that mimicking the connectivity of simple biological networks can significantly enhance deep learning efficiency while maintaining high accuracy, particularly in resource-constrained environments. However, complex tasks often require more neurons, leading to increased network size and computational demands. To address this, Roberts et al. [35] combined real-world neural topologies with convolutional neural networks (CNNs) to reduce model complexity without sacrificing computing power. Similarly, Su et al. [36] proposed network structures based on biological connectivity groups, further validating the potential of biologically inspired architectures in optimizing convolution operations.

More recent work has shown that neural networks inspired by the structure of C. elegans can perform complex tasks with fewer neurons without compromising performance. Lechner et al. [37] demonstrated efficient real-time decision-making under limited computational resources, while Razzaq et al. [38] proposed a hybrid architecture combining biological neural circuits with CNNs, achieving high-risk decision-making using only a small number of neurons. These findings underscore the potential of lightweight, bio-inspired networks for efficient, high-performance applications.

Building on this foundation, our work proposes a novel lightweight YOLOv8 architecture inspired by biological neural circuits, focusing on optimizing the backbone to better balance accuracy and efficiency. This approach offers a promising direction for enhancing YOLOv8 deployment in resource-constrained environments without compromising detection performance.

3 Methodology

This section provides an overview of the proposed CEYOLOv8 model, inspired by the Olfactory Learning Circuit of C. elegans. The network architecture is designed to leverage the efficient neural circuits observed in C. elegans, focusing on minimizing redundant connections and improving computational efficiency for real-time detection tasks. The construction of the network can be broken down into three main components: Backbone, Neck, and Head, as depicted in Fig.2.

3.1 Olfactory Learning Circuit of C. elegans

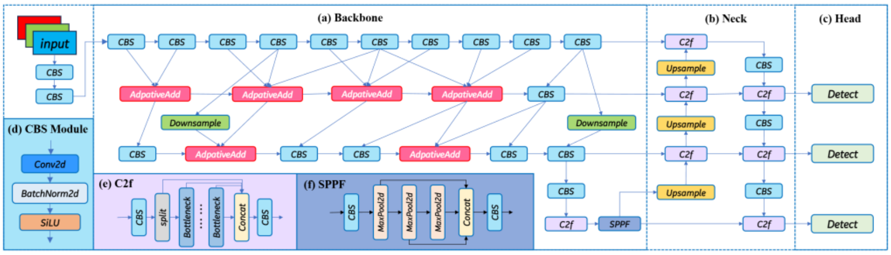

Fig. 2. The network of CEYOLOv8. Each CBS module simulates the neuron nodes in the olfactory learning Circuit of Caenorhabditis elegans. The network improves feature extraction through the Olfactory Learning Circuit derived from C. elegans.

The C. elegans has a simplified nervous system consisting of 302 neurons and their synaptic connections. The nervous system consists of three main types of neurons: sensory neurons, interneurons, and motor neurons, which are interconnected through synapses to form a highly optimized information processing network.

In this neural network, the neural information flow starts from sensory neurons, passes through intermediate neurons for processing, and finally reaches motor neurons. This efficient neural circuit enables C. elegans to exhibit various behavioral mechanisms, including olfactory learning, tactile response, and foraging behavior, relying on its concise and efficient neural network structure, making it an important model for functional research and biomimetic design of neural circuits. However, modern neural network models often face problems of redundant parameters and inefficient structures, which limit their learning and reasoning abilities. F Bardozzo's research [23] suggests that the design of network architecture has a significant impact on the learning ability of models, and complex or redundant structures may hinder the effective learning of networks. Therefore, drawing on the efficient neural connection structure of C. elegans, optimizing neural networks is a path worth exploring.

The neural circuits of C. elegans have been optimized under long-term evolutionary pressure, demonstrating high functionality and information processing efficiency. The neural connectivity group has been extensively studied and mapped, fully revealing the ways in which neurons are connected. In previous study [39], the hierarchical structure of the nematode nervous system was clarified: sensory neurons are located in the upper layer, interneurons are located in the middle layer, and motor neurons are located in the lower layer. The study also suggests that in the context of aversive olfactory learning, strongly connected neurons play a crucial role, while neurons with weaker synaptic connections are ignored. Through this method, the neural circuit responsible for odor aversion learning in the nematode neural network was identified. This circuit consists of 22 functional neurons, including 10 sensory neurons, 5 interneurons, and 7 motor neurons, which interact with each other through 21 synaptic connections, as shown in Fig. 1.

3.2 YOLOv8 Architecture

YOLOv8 is a fast and accurate single-stage object detector that directly outputs bounding boxes and class probabilities in a single forward pass. Its architecture comprises three modules: Backbone, Neck, and Head. The Backbone extracts multi-scale features from 640×640 input images using convolutional layers and C2f (Cross-Stage Feature Fusion) blocks for enhanced cross-stage fusion. The Neck adopts a PANet structure to aggregate features across scales, improving spatial and semantic consistency. The Head generates final predictions, including object categories, confidence scores, and bounding box coordinates.

3.3 The Proposed CEYOLOv8 Network

The overall architecture is shown in Fig.2. Our proposed CEYOLOv8 model largely follows the traditional YOLOv8 architecture, with the primary modifications occurring in the Backbone, while the Neck and Head networks remain consistent with those in YOLOv8.

We construct the Backbone of the neural network architecture based on the same network topology as prior studies [39]. To facilitate a clearer understanding of the feature extraction process, we divide the network into three distinct phases: S1, S2, and S3. After each phase, the size of the feature map is reduced to one-fourth of its original size, while the number of channels is doubled. Inspired by the execution paths of nematode neural networks and the hierarchical structure of convolutional neural networks, we define these phases according to their respective functions: S1 corresponds to the motor layer, S2 to the intermediate layer, and S3 to the decision layer. This simplified and efficient structure informs the design of the Backbone, where each layer's neuron type directly corresponds to a layer in the network. Specifically, the motor layer consists of 10 motor neurons (ADF, AWA, ASJ, PVD, ASER, ADL, AWC, FLP, ASH, and PHB); the intermediate layer contains 5 intermediate neurons (AIZ, ALA, AIB, AVD, and AVA); and the decision layer is composed of 7 decision neurons (SMB, CAN, SMD, SAB, DA, VA, VD). In each layer, neurons are processed through 2D convolution with a kernel size of 3, stride of 1, and padding of 1. All nodes in a given layer share the same processing unit, which consists of a SILU activation function, convolutional operation, followed by batch normalization, collectively forming the CBS (Convolution-BatchNorm-SILU) module.

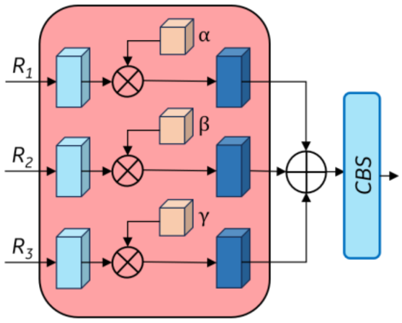

Fig. 3. The structure of AdaptiveAdd module.

Sequential processing starts at the motor layer. The output of each node is passed to the next node via a directed edge, mimicking synaptic connections in biological circuits. The intermediate layer receives outputs from the motor layer, combines the information from corresponding nodes, and passes the fused results to the next layer. Since some nodes receive inputs from multiple other nodes, the features of these nodes must be fused. Inspired by biological neural systems, we design a learnable fusion module that automatically adjusts the weights of the inputs from different nodes and effectively fuses their features. For example, when fusing the features of three nodes, as shown in Fig.3, we derive the fused representation. Specifically, the input consists of nodes: $\{R_i | i = 1,2,3\}$, where i indicates the node number, and R_i is the associated output for node r. The output feature R_o is a weighted sum of the three input features, with adaptive parameters governing the fusion weights. The mathematical formulation of this process is:

$$R_o = \alpha * R_1 + \beta * R_2 + \gamma * R_3 \tag{1}$$

in which α, β, γ are learnable weights. In the input of the network, we begin by feeding the 3-channel image into the model, followed by downsampling using 2D convolutions with a kernel size of 3, stride 2, and padding 1. This pre-extends the number of feature channels, initially increasing from 3 to 32, and then from 32 to 64 after two downsampling operations. The resulting features are then passed as input F_1 to each stage, yielding the final features F_i at each stage.

$$F_1 = S1(F_0), F_2 = S2(F_1), F_3 = S3(F_2) \tag{2}$$

We use three consecutive stages to progressively extract features, with the final stage incorporating the Spatial Pyramid Pooling Fusion (SPPF) module. The SPPF module performs pooling operations at multiple spatial scales, capturing fine and coarse features from various image regions. This enhances the model’s ability to learn multi-scale features, particularly in handling objects of varying sizes. The specific expression form is as follows:

$$F_4 = SPPF(Conv(F_3)) \tag{3}$$

To fuse stage features in object detection, we introduce an additional branch of the motion layer at the head of the network, as shown in Fig.2. The motion layer is the first layer of the network that is used for the detection of small objects. This layer corresponds to the feature maps generated at the beginning of the network, which contain rich local information. By connecting the features of the motion layer with the previously fused features, CEYOLOv8 is able to utilize fine-grained features effectively and organically fuse high-resolution and low-resolution features. This enhances the ability of the neck layer to extract multi-scale features by equation (4), thus improving the model's integration and utilization of features from all stages of the backbone.

$$P_i^{out} = C2f(Concat(Up(P_{i-1}^{in}), F_{4-i}), n) \tag{4}$$

where Up is upsample operations, F and P correspond to the feature maps at the backbone and neck respectively.

4 Evaluation Results

To evaluate our model’s effectiveness, we conducted experiments on frontier and object detection tasks in exploration scenarios using two public datasets. A randomly selected labeled subset was used to verify generalization. We compared CEYOLOv8 model against YOLOv8 variants of different sizes. Training was performed with a batch size of 16 and a learning rate of 0.001 using Stochastic Gradient Descent(SGD) and cosine annealing. Implementation was based on PyTorch [40] and executed on an NVIDIA RTX 4090 (24GB). Deployment tests were conducted on NVIDIA Jetson Orin Nano. The results demonstrate that our model achieves a reduction while maintaining comparable accuracy and inference speed, particularly under constrained computational resources.

4.1 Evaluation Criteria

To comprehensively assess performance and generalization, we adopted standard metrics including precision (P), recall (R), mAP@50, and mAP@50:95. Precision measures the proportion of true positives among predicted positives P, while recall measures the proportion of true positives among all actual positives R. mAP@50 evaluates mean average precision at an IoU threshold of 0.5, and mAP@50:95 averages AP across IoUs from 0.5 to 0.95 in 0.05 increments. Additionally, we report the number of learnable parameters and FLOPs to quantify model complexity.

4.2 Results on Frontier Dataset

Data and Setups. We used the Frontier dataset, containing 346 training, 40 validation, and 40 test samples, with 256×256×3 input size. The model was trained for up to 500 epochs using SGD (learning rate 0.01). Early stopping was applied if no improvement occurred over 10 consecutive epochs, selecting the best model based on validation accuracy.

Quantitative Evaluation. As shown in the table 1, we present a detailed comparison of our model with several novel lightweight algorithms and YOLOv8n. In terms of parameter count and performance metrics, our approach offers the best cost-performance ratio. Compared to YOLOv8n, our model achieves a 16% reduction in parameter count while significantly outperforming YOLOv8n across all metrics. When compared to the YOLO-LITE algorithm, our model has only a marginally higher parameter count of 0.16M, but it outperforms YOLO-LITE in all evaluated metrics. Although the parameter count of our model is higher compared to Backbone optimization algorithms such as CSCDense, under stringent evaluation criteria (mAP@0.5:0.95), our algorithm maintains SOTA performance. This suggests that the optimized C. elegans-inspired connection method strikes an effective balance between lower parameter count and high performance, making it a reliable choice.

Table 1. Detection results on Frontier dataset with comparative models.

Model	Params(M)	Precision(%)	Recall(%)	mAP0.5(%)	mAP0.5:0.95(%)
YOLOv8n	2.87	78.1	77.3	78.8	54.3
YOLO-LITE	2.24	84.5	76.8	83.1	58.9
CSCDense	**2.06**	78.7	72.9	82.3	58.1
EfficientRep	2.65	**87.2**	72.9	83.9	58.7
TP-YOLO	4.08	82.7	71.4	82.7	56.4
OURS	2.4	86.5	**82.3**	**86.9**	**59.1**

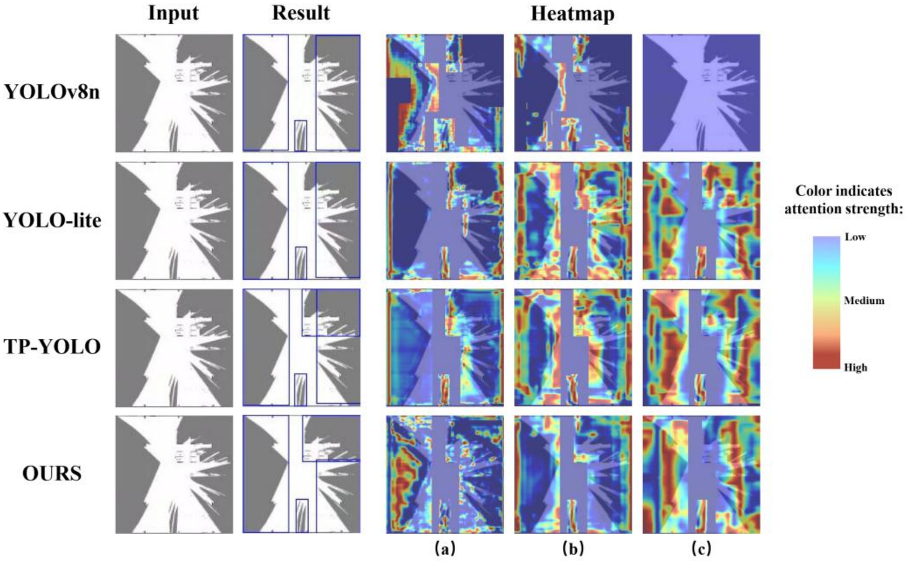

Fig. 4. The heatmap of different models

Heatmap analysis. To gain a deeper understanding of the model's perception and attention differences towards boundary data, we compared a YOLO optimization algorithm with three detection heads. As shown in the figure \ref{fig:heatmap}, the model receives the raw boundary images as input and outputs the detected boundary boxes, while also capturing heatmaps from the outputs of different detection heads at various scales. We observed a significant sparsity in the attention maps of the YOLOv8n algorithm's detection heads, with the third detection head losing fine-grained attention to details (as shown in the heatmap in part c of the figure). In comparison to other lightweight algorithms, our method provides more accurate detection results, with finer boundary delineation. Additionally, the heatmaps from our three detection heads sequentially focused on unexplored regions (as shown in heatmap a), known/unknown boundary areas (as shown in heatmap b), and the global boundary regions (as shown in heatmap b). Our experiments demonstrate that the inclusion of the C. elegans-inspired neural circuit mechanism in our algorithm enhances the interpretability of the model's attention and leads to better performance.

4.3 Results on Objects Dataset

Data and Setups. The objects dataset consists of one type of target and three types of obstacles 640 × 640 color images, divided into 284 training sets, 40 testing sets, and 12 validation sets. We trained the model using the SGD optimizer with a learning rate of 0.01 over 500 iteration cycles. Similar to the training method above, when the validation set cannot be improved, the training will also be stopped in advance.

Table 2. Detection results on Objects dataset with comparative models.

Model	Params(M)	Precision(%)	Recall(%)	mAP0.5(%)	mAP0.5:0.95(%)
YOLOv8n	2.87	94.4	73.7	87.2	68
YOLO-LITE	2.24	82.8	79.5	87.1	68.2
CSCDense	**2.06**	90.3	65.1	82.6	63.8
EfficientRep	2.64	84.9	74.5	**89.3**	66.5
TP-YOLO	4.08	**95.7**	**82.1**	87.8	67.3
OURS	2.4	89.3	72.2	88.8	**68.5**

Quantitative Evaluation. As shown in Table 2, our proposed model achieves a strong balance between parameter efficiency and detection performance, demonstrating its robustness across multiple metrics. Compared to YOLOv8n, our model reduces the parameter count by 16\% while improving mAP@0.5:0.95, which highlights its ability to maintain high accuracy under stringent evaluation. Against YOLO-LITE, our method delivers superior performance across all metrics with only a marginal increase in parameter count (+0.16M), demonstrating a more favorable trade-off between model size and effectiveness. Although TP-YOLO achieves slightly higher mAP@0.5, this improvement comes at the cost of a significantly larger parameter count (+70\% compared to ours), making our approach more efficient for resource-constrained applications. Furthermore, our model surpasses CSC-Dense and EfficientRep by a clear margin in terms of both recall and mAP, validating the effectiveness of the optimized lightweight design. These results collectively confirm that our model strikes an optimal balance between parameter size and detection performance, making it a competitive and practical choice for lightweight object detection.

4.4 Ablation studies

In this study, we assessed the effectiveness of our proposed network connectivity method by comparing it with the original YOLOv8n model and three distinct feature fusion strategies. The experimental results presented in Table 3 demonstrate that, when applying our network connection method inspired by the C. elegans circuit, all feature fusion strategies outperform YOLOv8n in terms of both parameter count and performance metrics, thereby validating the effectiveness of our C. elegans-inspired connection strategy. Among the three feature fusion strategies—Add, Concat, and our proposed AdaptiveAdd—the AdaptiveAdd module consistently achieves superior performance. Compared to the Add method, AdaptiveAdd not only maintains the same parameter count but also enhances performance. When compared to the Concat method, AdaptiveAdd reduces the number of parameters while delivering improved performance across key metrics, further solidifying its superiority as the optimal feature fusion strategy within our network. The experimental results further validate the effectiveness of our proposed connectivity method within YOLOv8, showing that the AdaptiveAdd module not only boosts performance but also outperforms traditional fusion methods in terms of both accuracy and computational cost, demonstrating its higher efficiency.

Table 3. Ablation study on our model

Model	Params(M)	Precision(%)	Recall(%)	mAP0.5(%)	mAP0.5:0.95(%)
YOLOv8n	2.87	78.1	77.3	78.8	54.3
OURS-Add	2.4	82.8	77.8	83.7	57.4
OURS-Concat	2.47	81.7	84.2	85.6	58.9
OURS-AdaptiveAdd	2.4	86.5	82.3	86.9	59.1

5 Conclusions

In this paper, we introduce a novel convolutional neural network architecture, the Nematode Connector Neural Network (CEYOLOv8), inspired by the olfactory learning circuits of C. elegans. This represents the first instance of integrating biological neural circuit principles into detection neural networks. Our key innovation lies in designing a backbone network structure directly modeled on the connectivity patterns of biological neurons in C. elegans. Detailed implementation steps and structural descriptions of CEYOLOv8 demonstrate that this biologically inspired approach substantially reduces the number of parameters required by neural networks, significantly lowering both data and hardware requirements without sacrificing overall accuracy.

However, despite its optimized architecture, CEYOLOv8 experiences performance limitations, notably reduced detection accuracy, when applied to complex or high-resolution scenes. This reduction in parameters can diminish the model's capacity to capture intricate, fine-grained details necessary for precise detections in such demanding scenarios. Potential strategies to mitigate these performance drops include employing multi-scale feature extraction, integrating attention mechanisms to enhance feature representation, and selectively reintroducing targeted complexity to key network layers to maintain efficiency while improving precision. These strategies can help balance model efficiency with the accuracy required for high-resolution or detailed object detection tasks, highlighting promising avenues for future research.

References

1. Zhong H, Xiao L, Wang H, et al. LiFSO-Net: A lightweight feature screening optimization network for complex-scale flat metal defect detection[J]. Knowledge-Based Systems, 2024, 304: 112520.
2. Chen T, Todo Y, Zhang Z, et al. A learning orientation detection system and its application to grayscale images[J]. Knowledge-Based Systems, 2025, 310: 112901.
3. Li Z, Dong Y, Shen L, et al. Development and challenges of object detection: A survey[J]. Neurocomputing, 2024, 598: 128102.

4. Mahaur B, Mishra K K. Small-object detection based on YOLOv5 in autonomous driving systems[J]. Pattern Recognition Letters, 2023, 168: 115-122.
5. Mahaur B, Mishra K K, Kumar A. An improved lightweight small object detection framework applied to real-time autonomous driving[J]. Expert Systems with Applications, 2023, 234: 121036.
6. Jia X, Tong Y, Qiao H, et al. Fast and accurate object detector for autonomous driving based on improved YOLOv5[J]. Scientific reports, 2023, 13(1): 9711.
7. Jain S. Deepseanet: Improving underwater object detection using efficientdet[C]//2024 4th International Conference on Applied Artificial Intelligence (ICAPAI). IEEE, 2024: 1-11.
8. Ma B, Hua Z, Wen Y, et al. Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments[J]. Artificial Intelligence in Agriculture, 2024, 11: 70-82.
9. Maalouf A, Jadhav N, Jatavallabhula K M, et al. Follow anything: Open-set detection, tracking, and following in real-time[J]. IEEE Robotics and Automation Letters, 2024, 9(4): 3283-3290.
10. Zhou Q, Shi H, Xiang W, et al. DPNet: Dual-path network for real-time object detection with lightweight attention[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024.
11. Hussain M. Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo[J]. IEEE access, 2024, 12: 42816-42833.
12. Situ Z, Teng S, Liao X, et al. Real-time sewer defect detection based on YOLO network, transfer learning, and channel pruning algorithm[J]. Journal of Civil Structural Health Monitoring, 2024, 14(1): 41-57.
13. Liu X, Wang T, Yang J, et al. MPQ-YOLO: Ultra low mixed-precision quantization of YOLO for edge devices deployment[J]. Neurocomputing, 2024, 574: 127210.
14. Liu B, Jiang W. DFKD: Dynamic Focused Knowledge Distillation Approach for Insulator Defect Detection[J]. IEEE Transactions on Instrumentation and Measurement, 2024.
15. Liu J, Li Y, Chen Z. LRFE-YOLO: Low-Rank Feature Enhancement for YOLO in Few-Shot Industrial Meter Fine Needle Detection[C]//2024 9th International Symposium on Computer and Information Processing Technology (ISCIPT). IEEE, 2024: 449-454.
16. Thakur A, Kumar R, Bhadoria R S. YOLO-NAS Based Deep Learning Approach for Breast Lesion Identification[C]//2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI). IEEE, 2024, 2: 1-4.
17. Pandey N N, Pati A, Maurya R. DriSm_YNet: a breakthrough in real-time recognition of driver smoking behavior using YOLO-NAS[J]. Neural Computing and Applications, 2024, 36(29): 18413-18432.
18. Lanza E, Di Angelantonio S, Gosti G, et al. A recurrent neural network model of C. elegans responses to aversive stimuli[J]. Neurocomputing, 2021, 430: 1-13.
19. Dan Y, Sun C, Li H, et al. Adaptive spiking neuron with population coding for a residual spiking neural network[J]. Applied Intelligence, 2025, 55(4): 288.
20. Lin C, Qiao Y, Pan Y. Bio-inspired interactive feedback neural networks for edge detection[J]. Applied Intelligence, 2023, 53(12): 16226-16245.
21. Chen T, Kobayashi Y, Yan C, et al. A learning artificial visual system and its application to orientation detection[J]. Applied Intelligence, 2025, 55(4): 310.
22. Park C, Kim J S. Caenorhabditis elegans connectomes of both sexes as image classifiers[J]. Experimental Neurobiology, 2023, 32(2): 102.
23. Bardozzo F, Terlizzi A, Lió P, et al. ElegansNet: a brief scientific report and initial experiments[J]. arXiv preprint arXiv:2304.13538, 2023.

24. Currier T A, Pang M M, Clandinin T R. Visual processing in the fly, from photoreceptors to behavior[J]. Genetics, 2023, 224(2): iyad064.
25. Bentley B, Branicky R, Barnes C L, et al. The multilayer connectome of Caenorhabditis elegans[J]. PLoS computational biology, 2016, 12(12): e1005283.
26. Jin X, Yu W, Chen D W, et al. DFD-NAS: General deepfake detection via efficient neural architecture search[J]. Neurocomputing, 2025, 619: 129129.
27. Lu S, Hu Y, Yang L, et al. PHD-NAS: Preserving helpful data to promote Neural Architecture Search[J]. Neurocomputing, 2024, 587: 127646.
28. Kumar Y, Kumar P. Comparative study of YOLOv8 and YOLO-NAS for agriculture application[C]//2024 11th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, 2024: 72-77.
29. Lai H, Liu B, Kan H Y, et al. YOLOv8-lite: An Interpretable Lightweight Object Detector for Real-Time UAV Detection[C]//2023 9th International Conference on Computer and Communications (ICCC). IEEE, 2023: 1707-1713.
30. Di Y, Phung S L, Van Den Berg J, et al. TP-YOLO: A lightweight attention-based architecture for tiny pest detection[C]//2023 IEEE International Conference on Image Processing (ICIP). IEEE, 2023: 3394-3398.
31. Kumari S, Chakravarthy V S. Biologically inspired image classifier based on saccadic eye movement design for convolutional neural networks[J]. Neurocomputing, 2022, 513: 294-317.
32. Wülfing J M, Kumar S S, Boedecker J, et al. Adaptive long-term control of biological neural networks with deep reinforcement learning[J]. Neurocomputing, 2019, 342: 66-74.
33. Manglani H R, Fountain-Zaragoza S, Shankar A, et al. Employing connectome-based models to predict working memory in multiple sclerosis[J]. Brain Connectivity, 2022, 12(6): 502-514.
34. Bardozzo F, Terlizzi A, Simoncini C, et al. Elegans-AI: How the connectome of a living organism could model artificial neural networks[J]. Neurocomputing, 2024, 584: 127598.
35. Roberts N, Yap D A, Prabhu V U. Deep connectomics networks: Neural network architectures inspired by neuronal networks[J]. arXiv preprint arXiv:1912.08986, 2019.
36. Su D, Chen L, Du X, et al. Constructing convolutional neural network by utilizing nematode connectome: A brain-inspired method[J]. Applied Soft Computing, 2023, 149: 110992.
37. Lechner M, Hasani R, Amini A, et al. Neural circuit policies enabling auditable autonomy[J]. Nature Machine Intelligence, 2020, 2(10): 642-652.
38. Razzaq W, Hongwei M. Neural circuit policies imposing visual perceptual autonomy[J]. Neural Processing Letters, 2023, 55(7): 9101-9116.
39. Wang X, Liu C, Zhao M, et al. An Artificial Neural Network for Image Classification Inspired by the Aversive Olfactory Learning Neural Circuit in Caenorhabditis elegans[J]. Advanced Science, 2025, 12(7): 2410637.
40. Paszke A. Pytorch: An imperative style, high-performance deep learning library[J]. arXiv preprint arXiv:1912.01703, 2019.