International Journal of Engineering Innovations & Technology

ISSN:

Research Article

Improving YOLOv11 Performance in Underwater Fish Detection via Evolutionary Tuner and Selective Kernel Attention

Hua-Ching Chen1, Hsuan-Ming Feng2,

1School of Information Engineering, Xiamen Ocean Vocational College, Fujian, China
2Department of Computer Science and Information Engineering, National Quemoy University, Kinmen, Ta

Received: 2025 Dec 05 **Accepted:** 2025 Dec 11 **Published:** 2025 Dec 16

Abstract

This study aims to enhance the accuracy of underwater fish detection by proposing a dual-enhanced YOLOv11 model. The approach leverages two key improvements: First, a selective kernel attention (SKA) mechanism is incorporated into the YOLOv11 architecture to enable dynamic selection of multi-scale convolution kernels, improving adaptability to various target sizes. Second, an evolutionary tuner (ET) is employed for hyperparameter optimization to refine model performance further. The proposed model achieves significant gains over the baseline, with improvements of 2.06% in mean average precision (mAP)@0.5 and 6.30% in mAP@0.5:0.95, attaining final scores of 98.629% and 86.933%, respectively. The dual-enhanced model demonstrates superior accuracy and robustness in complex underwater environments, ultimately achieving a precision of 99.069% and a recall of 95.968%.

Keywords: *evolutionary tuner, hyperparameter optimization, image feature extraction, selective kernel attention*

1. Introduction

Global climate change and intensified human activities are placing unprecedented pressure on marine ecosystems. Issues such as rising sea levels, ocean acidification, and biodiversity decline are becoming increasingly severe, posing a significant threat to the health of these ecosystems [1]. Given the profound impact of marine ecosystems on human survival and economic development, the international community has significantly increased its focus on effective ecological monitoring. In this context, the accurate and efficient detection and identification of underwater fish have become a crucial component of marine resource management and environmental protection. This technology is vital for understanding ecosystem health and has a wide range of applications, including marine fishing, aquaculture, resource surveys, ecological protection, and underwater robot navigation [2]

Automated underwater fish monitoring is a challenging task due to the complex and dynamic nature of the aquatic environment. Factors such as low visibility, varying lighting conditions, and water turbidity complicate image acquisition and processing [3]. The morphological diversity among different fish species further complicates the detection and identification process. While advancements in deep learning and artificial intelligence have significantly improved detection accuracy, traditional methods and even some advanced models are often hindered by a lack of precision, efficiency, and robustness in real-world scenarios [4-5]. Conventional monitoring methods have proven inadequate as the requirements for monitoring accuracy and efficiency have increased with
This study addresses these challenges by proposing a dual-enhanced YOLOv11 model. The primary objective of this study is to significantly improve the accuracy and robustness of underwater fish detection by integrating two key enhancements.
First, a novel selective kernel attention (SKA) mechanism is incorporated to enable the model to dynamically adjust to multiscale features, thereby improving its adaptability to various fish sizes. Second, an evolutionary tuner (ET) is utilized to optimize
critical hyperparameters, further refining the model's performance [6]. The synergy of these two enhancements yields superior
results compared to existing state-of-the-art methods, providing new technical support for marine ecological monitoring. The
innovations of this study are twofold:

(1) The structural adjustment of the proposed SKA mechanism, combined with an ET for hyperparameter self-adjustment,
successfully improves underwater fish detection and recognition capabilities, verifying its effectiveness in enhancing
accuracy and efficiency.

(2) The analysis of experimental results yields a new

echnical approach that outperforms many related identification methods in multi-category fish detection applications, enabling a deeper exploration of marine ecological data and furnishing conservation efforts with a more scientific basis.
This paper is structured as follows: Section 2 reviews relevant literature on underwater object detection and attention mechanisms. Section 3 details the proposed dual-enhanced YOLOv11 model, including its architecture and training methodology. Section 4 presents the experimental setup, results, and a comprehensive performance analysis. Finally, Section 5 concludes the study and outlines directions for future work..

2. Related Works

The increasing complexity of underwater environments and the rich biodiversity they harbor have garnered significant attention from international researchers in recent years. Among the diverse research areas within computer vision, underwater fish detection and recognition has emerged as a prominent field, particularly with the advancements in deep learning techniques. In the realm of international research, object detection models based on convolutional neural networks (CNNs), mainly the you only look once (YOLO) series, have found widespread applications in underwater fish detection and recognition tasks. YOLOv11, the latest iteration in this series, has become an ideal choice for underwater biological identification missions due to its rapid and accurate performance.

Researchers have further enhanced the model's performance in complex underwater environments by introducing selfadjusting hyperparameter designs. For instance, Yang et al. [7] utilized the YOLO model with Mamba-C2f to capture richer gradient flow and long-/h////////range dependencies, thereby enhancing underwater fish detection. Kaur and Vijay [8] applied the YOLO nmodel to identify underwater fish images, improving the accuracy and efficiency of detection. Qin et al. [9] proposed a high-precision underwater fish identification method based on YOLOv8-FASG, which significantly improved the accuracy and recall rates of fish detection. Experimental results demonstrated that this method achieved a 1.3% and 6.1% increase in mean average precision (mAP)@0.5 and mAP@0.5:0.95 metrics, respectively, and a 1.6% and 3.5% improvement in precision and recall rates, providing more reliable fish swarm identification capabilities for underwater robots.

2.1. Image feature extraction

Image feature extraction is a process of acquiring key information from images to support subsequent analysis and processing. Deep learning, a machine learning (ML) technique that relies on neural networks, can automatically learn and extract high-level, abstract features from data. These two approaches have broad applications in various fields, such as facial recognition, object detection, and medical image analysis, where feature extraction and deep learning technologies play crucial roles [10].

Current underwater fish recognition systems face specific performance challenges. Firstly, their effectiveness is often limited by factors such as image quality, algorithm complexity, and computational resources. Secondly, in complex underwater environments, these systems exhibit weak adaptability, and the recognition accuracy of biological targets requires further improvement. To enhance the performance of underwater fish recognition systems, in-depth enhancements are required in algorithm optimization, data processing, and system design to adapt to the ever-changing underwater conditions and improve recognition accuracy [11]. Analysis of existing underwater fish recognition systems reveals that these systems can achieve a certain degree of biological identification under conditions of high-quality images and low-complexity algorithms.

However, their adaptability and recognition accuracy in complex underwater environments remain insufficient. To improve system performance, efforts can be made in the following aspects: optimizing algorithms by employing more advanced deep learning techniques to enhance recognition accuracy and adaptability; performing data preprocessing to improve image quality and reduce noise interference; and redesigning the system architecture to make it more flexible, thereby improving the system's robustness and adaptability [12].

YOLOv11 has played an essential role in underwater fish detection and recognition tasks. Complex and variable underwater environments, with issues such as uneven illumination, complex backgrounds, and similar fish morphology, pose severe challenges to target detection [13]. YOLOv11 effectively extracts key features of underwater images by improving the CNN structure. The model employs multi-scale feature fusion and the SKA mechanism, enhancing the recognition ability of fish edges and textures to subsequently improve detection accuracy and robustness [14]. Self-adjusting hyperparameter optimization further enhances the model's adaptability to various underwater conditions, enabling it to accurately identify different fish species, such as rays, groupers, and angelfish [15]. This feature extraction strategy significantly improves the model's detection ability for small targets and complex backgrounds, while effectively reducing the false positive rate.
YOLOv11 has broad application prospects in fields such as underwater ecological monitoring and marine biodiversity research.

2.2. The state-of-the-art and visualization of underwater image detection

In recent years, deep learning techniques, particularly CNNs, have demonstrated immense potential in the field of underwater image analysis. Numerous researchers have successfully utilized CNNs to identify underwater fish species, achieving remarkable results. Villon et al. [16] pioneered the use of CNNs for identifying low-resolution or blurred fish images, achieving an accuracy of 94.9%, laying the foundation for research in this field. Jalal et al. [17] employed deep learning techniques to achieve high-precision classification of fish species. Despite the absence of an explicit mAP value in the paper, high-precision classification tasks typically require the model to possess a high mAP.

The PLGAT (Plectropomus leopardus recognition using Global Attention mechanism and transfer learning) model proposed by Liu et al. [18] achieved an mAP of approximately 97% in the leopard coral grouper

recognition task, indicating the model's high accuracy in recognizing this species. Wu et al. [19] proposed an improved YOLOv5-fish model for underwater blurred images, achieving an mAP of 97.6% and a detection speed of 84 frames per second (FPS), demonstrating the model's strong robustness in handling complex underwater scenes. However, despite the significant progress made, existing research still faces several challenges, such as the complexity of underwater environments, the instability of lighting conditions, and the diversity of targets, all of which place higher demands on the model's performance. Moreover, small target detection, target occlusion, and multi-scale target detection remain pressing issues that need to be addressed.

Deep learning has demonstrated a strong ability to solve complex problems; however, its performance largely depends on the choice of hyperparameters. Manually adjusting hyperparameters is both time-consuming and inefficient; therefore, researchers have begun to explore the use of evolutionary algorithms e.g., such as genetic algorithm (GA) and particle swarm optimization (PSO), for hyperparameter optimization. These algorithms simulate the biological evolution process, searching for optimal solutions in the hyperparameter space through operations such as selection, crossover, and mutation. Evolutionary algorithms possess global search capabilities, effectively avoiding being trapped in local optimal solutions, and can handle high-dimensional, non-linear hyperparameter spaces.

Several studies have successfully applied evolutionary algorithms to optimize the hyperparameters of deep learning models, significantly improving their performance in tasks such as image classification and object detection. For example, Xu et al. [20] proposed a fast algorithm combining a gradient descent algorithm and an evolutionary learning algorithm, which can effectively solve other neural networks with better characteristics and higher accuracy. ML techniques have been actively explored for electricity theft detection. In particular, methods based on Support Vector Machines (SVM) have been investigated by Abro et al. [21] and have demonstrated promising detection performance. To address the challenges inherent in existing underwater fish target detection models in complex underwater environments, such as small target detection, occlusion, and lighting changes, this study proposes an improved model based on YOLOv11.

This model incorporates an ET method to optimize the hyperparameters of the YOLO model, enabling fine-grained adjustments to key parameters to achieve optimal model configuration. Simultaneously, an improved SKA mechanism is integrated to dynamically adjust the kernel function based on different types and sizes of fish, thereby further enhancing multiscale feature extraction. This empowers the model to adapt to objects of varying sizes, consequently improving target localization and classification accuracy. In addition, through carefully designed data augmentation strategies and loss function implementations, such as Focal Loss and Intersection over Union (IoU) Loss, the model's robustness in complex underwater environments is further enhanced. Experimental results demonstrate that the improved model has achieved significant performance improvements in underwater fish target detection tasks, offering new insights and methods for related research.

3. Research Methods

This chapter presents the proposed YOLOv11-based hybrid improvement framework designed to enhance underwater fish detection performance. Given the inherent challenges in feature extraction arising from the low contrast and high blurriness of underwater imagery, coupled with the high sensitivity of model training to hyperparameters, the proposed methodology incorporates a dual-optimization strategy. Technically, this strategy comprises the SKA

mechanism for architectural
enhancement and the ET for optimization strategy refinement. First, this chapter elucidates how SKA dynamically adjusts
receptive fields (RF) to enhance recognition capabilities for multi-scale and small-scale targets. Subsequently, it details how
the ET employs evolutionary algorithms to adaptively search for the global optimal solution within the complex
hyperparameter space, ensuring stable training convergence and superior generalization ability. This chapter comprehensively
demonstrates the scientific basis and technical efficacy of this hybrid mechanism in tackling the challenges of complex
underwater environments.

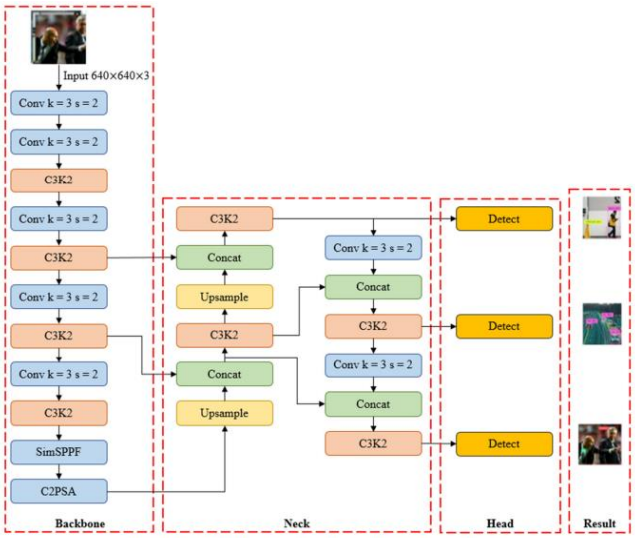


Fig. 1 YOLOv11 architecture (Backbone, Neck, and Head)

3.1. YOLO structure

The YOLO series of models has consistently demonstrated robust performance in object detection tasks, achieving a
favorable balance between speed and accuracy through iterative refinements. The latest iteration, YOLOv11, incorporates
several architectural innovations centered on the cross-stage partial with kernel size 2 (C3K2) block, spatial pyramid feature
fusion (SPFF) module, and cross-stage partial with spatial attention version 2 (C2PSA) block. These components significantly
enhance the model's capacity to process spatial information while maintaining high-speed inference capabilities. YOLOv11
demonstrates versatility, supporting a broad range of tasks including object detection (YOLOv11-detect), instance
segmentation (YOLOv11-seg), pose estimation (YOLOv11-pose), oriented object detection (YOLOv11-obb), and
classification (YOLOv11-cls) [22]. Fig. 1 illustrates the schematic architecture of the YOLOv11 algorithm:

3.2. Structured design of the improved YOLOv11 mechanism

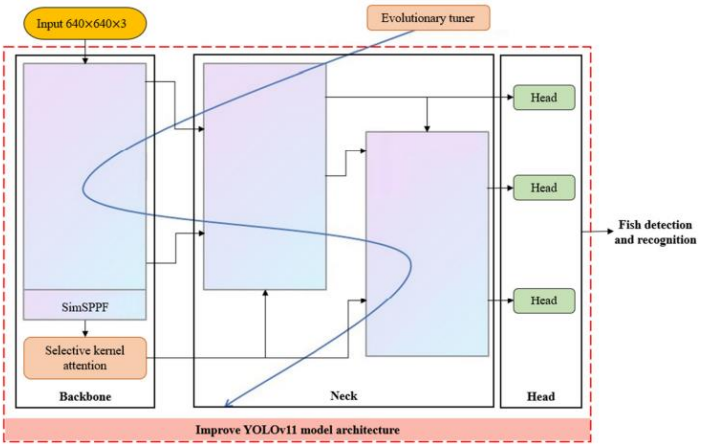
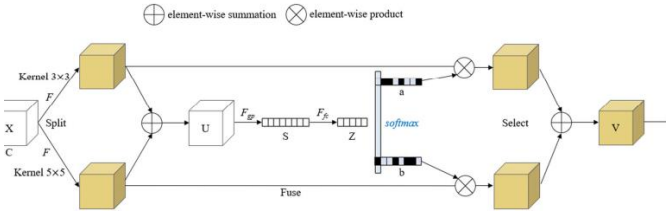


Fig. 2 illustrates the design of the YOLOv11 optimization structure and parameter adjustment process. This study proposes an integrated system architecture that combines the SKA mechanism for structural optimization with the ET method
for training strategy optimization. The ET is employed to optimize the hyperparameter space of the SKA-YOLOv11

framework, ensuring that the model converges efficiently and stably even when SKA dynamically selects convolutional kernels.

The scientific merit of this hybrid framework is derived from the structural improvement of SKA, elevating the model's
potential performance upper bound, while ET ensures that the model can efficiently reach and stably maintain this threshold,
thereby achieving an optimal balance between robustness and generalization in underwater detection.

The SKA mechanism proposed in this study utilizes a selective kernel (SK) approach to enhance the YOLOv11 structure,
enabling neurons to adaptively adjust their RF size across multiple kernels with varying sizes. Fig. 3 depicts the operational
procedure [23].



The SKA module consists of four primary components, as detailed below:

(1) Split: The input feature map is divided into two branches, which are subsequently convolved with 3x3 and 5x5 kernels,
respectively. These two branches capture features at different scales.

(2) Fuse: The output feature maps from the two branches are fused. The fusion is performed via

element-wise addition

(3) Attention:
Global average pooling (GAP): GAP is applied to the

fused feature map to obtain a channel-wise feature vector.
Fully connected layer and activation: The output of GAP is passed through a fully connected layer and an activation function,
such as the rectified linear unit (ReLU), to obtain two feature vectors, representing the importance of the 3×3 and 5×5
convolution kernels, respectively.

Softmax: A softmax operation is applied to these two feature vectors to obtain two probability values, reflecting the selection
probabilities for the 3×3 and 5×5 convolution kernels.
Scale: The output of softmax is multiplied element-wise with the outputs of the two branches to obtain weighted feature maps.
(4) Select: The weighted feature maps are added element-wise to obtain the final output feature map.

SKA is a dynamic selection mechanism for attention in CNNs. SKA enables each neuron to modulate its RF size based
on the multi-scale features of the input information and adaptively select convolution kernels of varying scales. Specifically,
SKA achieves this adaptability through a building block approach using SK units. Each SK unit comprises multiple branches
with different kernel sizes, enabling the extraction of features at various scales. Rather than simply concatenating or averaging
these branch features, SKA employs a mechanism that leverages the information within these branches to guide a softmax
attention mechanism. This attention mechanism facilitates a weighted fusion of the multiple branches with different kernel
sizes, resulting in a composite neuron with an adaptive RF size that is dynamically determined. By stacking various SK units,

a deep network with SK capabilities, known as SKNet, can be constructed. Subsequently, SKA aggregates the feature maps
from these units through weighted averaging, producing the final feature representation [23]

The SKA mechanism, by dynamically adjusting the RF size of neurons, enhances the model's capacity to detect multiscale objects and improves its adaptability in complex scenarios. The core technique underlying SKA finds primary
applications in areas involving multi-scale convolution, feature aggregation, and attention mechanisms. For instance, in object

detection tasks, SKA assists the model in more effectively identifying targets of varying sizes, resulting in improved detection
accuracy and robustness. In image classification tasks, SKA enables the model to capture multi-scale information more
effectively, thereby enhancing classification accuracy. The computational details of the SKA

mechanism are mathematically
formulated in the following equations:

(1) Multi-scale convolution: Assume that the input feature map is ∈

××, where is the number of channels, and and

are the height and width of the feature map, respectively. Convolution operations of different scales are applied to the input
feature map to generate multiple feature maps:
Y W X for i K

here, denotes the i-th convolution kernel (or filter) with a distinct scale, Yi represents the i-th output feature map after the convolution operation, and X is the input feature map. The index i ranges from 1 to K, where K is the total number of convolution kernels of different scales. ∈

××, and each output feature map Yi is a tensor with the same dimensions as the input feature map after convolution.

(2) Feature aggregation: Weighted fusion of feature maps at different scales is performed to generate the final feature map

$$Z = \sum_{i=1}^{K} \alpha_i \cdot Y_i$$

αi denotes the weighting coefficient for each feature map, where it is learned through an attention mechanism, and
∈

××
represents the final fused feature map.

(3) Attention mechanism: The attention mechanism uses the global information of the input feature map to generate attention
weights. Assuming that the input feature map is X, the global feature vector is obtained through the GAP operation:

$$GAP(X) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j}$$

The global feature vector is processed by a fully connected layer to generate weighted coefficients of different scales:

$$\alpha_i = \sigma(W_i \cdot GAP(X)), \ for \ i = 1, 2, \ldots, K$$

Here, σ is the activation function (such as sigmoid or softmax), and Wi is the weight of the fully connected layer.
Steps to Apply the SKA mechanism in YOLOv11.
(1) Input feature map: Obtain the output feature map X from YOLOv11's Backbone.
(2) Multi-scale convolution: Apply convolution operations of different scales to X to generate multi-scale feature maps Yi.
(3) Calculate attention weight: Generate weighting coefficients for each scale through GAP and fully connected layers αi.
(4) Feature aggregation: Fuse multi-scale feature maps based on the weighted coefficients to generate the final feature map
Z.
(5) Output feature map: Input Z into the Neck part of YOLOv11 for further processing.
In YOLOv11, both C2PSA and SKA are attention modules that enhance network performance. C2PSA focuses on efficient local feature extraction, while SKA improves adaptability to multi-scale targets by dynamically selecting

convolutional kernels. The utilization of multi-scale convolutions and feature aggregation enhances detection flexibility and
generalization in complex scenes. Integrating SKA into each feature extraction layer optimizes the capability of YOLOv11 to
detect targets of varying sizes. Concerning the SKA mechanism, the innovation of this study is centered on its targeted
deployment. The proposed architecture integrates the SKA mechanism as a dedicated component to replace the original cross-

stage partial attention (CSPA) module in the YOLOv11 backbone network. This replacement is a targeted enhancement to
address the challenges of pronounced variations in underwater image quality and large multi-scale target differences, thereby
equipping the model with more flexible RF dynamics, which is essential for improving the robustness of underwater detection.

3.2. Evolutionary tuner optimization process

YOLOv11, as an advanced object detection model, exhibits performance that is highly dependent on the appropriate configuration of its hyperparameters. The adaptive hyperparameter setting provided by the ET optimization mechanism plays a critical role in the training process of YOLOv11. By systematically adjusting hyperparameters such as the learning rate, batch size, and network architecture, significant improvements can be achieved in detection accuracy, inference speed, and generalization ability. Hyperparameter settings directly influence the model's learning process, including convergence speed and final performance [14].

To identify optimal hyperparameter combinations, researchers commonly employ methods such as grid search, random search, or Bayesian optimization. Furthermore, data augmentation techniques, including image flipping, rotation, and noise addition, can effectively enhance the model's adaptability to diverse scenarios [24]. However, the dimensionality of the hyperparameter space renders manual tuning challenging and labor-intensive. Therefore, a comprehensive understanding of YOLOv11's internal mechanisms, coupled with visualization tools and empirical experience, is essential for efficient hyperparameter tuning.

The ET adaptive hyperparameter settings within YOLOv11 are contingent upon the specific task and dataset. Generally, hyperparameters can be classified into model architecture hyperparameters, optimizer hyperparameters, training hyperparameters, and data augmentation hyperparameters. Model architecture hyperparameters comprise the selection of the backbone network, the design of the feature pyramid network, and the configuration of the prediction head. Optimizer parameters, such as learning rate, momentum, and weight decay, directly impact the model's training process. Training hyperparameters, including batch size, iteration count, and learning rate decay strategies, govern the model's training efficiency and generalization ability. Finally, data augmentation hyperparameters enhance the model's robustness by expanding data diversity [14].

Hyperparameter optimization is an iterative process that seeks to find the optimal set of hyperparameters, maximizing the model's performance in the target detection task. An excessive learning rate can incur model instability, whereas an excessively low learning rate can result in slow training. Batch size impacts the model's training stability and efficiency. Furthermore, training epochs, network architecture, and data augmentation strategies are also critical determinants that influence model performance [25].

In the YOLOv11 implementation within this study, a GA is employed for hyperparameter optimization. This algorithm mimics the biological evolution process, generating new individuals via mutation operations that randomly perturb existing hyperparameters. These individuals are then selected based on model performance, ultimately yielding an optimized set of hyperparameters that enhances the probability of identifying a near-optimal solution. The ET adaptive hyperparameter setting, after an initial training phase of 300 iterations, demonstrates an adaptive training behavior, as illustrated in Fig. 4.

As illustrated in Fig. 4, the discrete markers denote the fitness values of the hyperparameter combinations generated by the ET optimization mechanism at each completed iteration. The dashed line depicts the smoothed trend, enabling a clearer visualization of fitness value changes across iterations. The markers located near the bottom of the graph correspond to randomly generated hyperparameter combinations produced by the GA at the initial stage of each iteration, for which the fitness values obtained after model training are minimal and close to zero. Based on Fig. 4, the following observations are

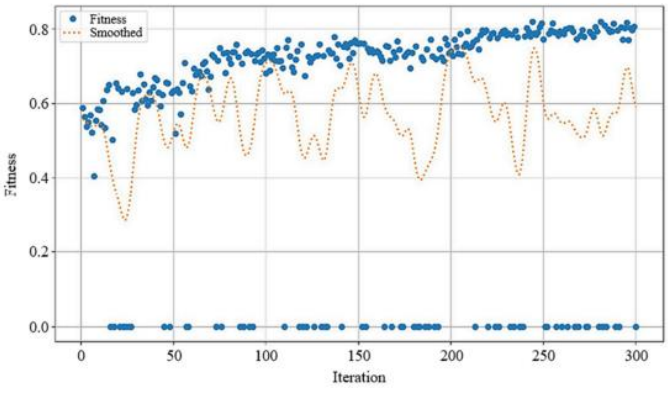
made:

(1) Early-stage fluctuations (0–50 iterations): In the initial training stage, because the hyperparameter combinations generated
by the ET optimization mechanism are random, the model's performance fluctuates significantly, and the fitness values
exhibit pronounced oscillations

(2) Mid-stage gradual stabilization (50–200 iterations): As the number of iterations increases, the GA continuously optimizes
the hyperparameter combinations, leading to a gradual improvement in model performance, with the fitness values tending
to stabilize. Notably, the distribution of individual data points becomes more concentrated, and the amplitude of the
smoothed trend curve decreases.

(3) Late stage convergence (200–300 iterations): In the final training stage, the hyperparameter combinations generated by
the ET optimization mechanism gradually converge toward a stable solution. Consequently, the rate of performance

improvement slows, and the fitness values exhibit minimal variation. The distribution of individual data points remains
concentrated at a relatively high level, while the corresponding smoothed tend curve stabilizes.



As shown in Fig. 4, the optimal solution (i.e., the point with the highest fitness value) is obtained around the 250th iteration
rather than near the final iteration (the 300th). This is attributed to the global search characteristic of the GA within the
hyperparameter space, which prevents the search from being limited to local optima. By simulating

operations such as selection,
crossover, and mutation in the biological evolution process, the GA explores the entire hyperparameter space. Consequently,
the optimal solution may emerge at any stage of the search process. Upon completion of the optimization, the ET generates a
series of files, including best.pt (model weight file), best_hyperparameters.yaml (best hyperparameter configuration file), and
tune_results.csv (training result record file)

The procedure employed by the GA in hyperparameter optimization within the ET mechanism is outlined below:
(1) Population initialization: A set of candidate solutions (i.e., hyperparameter combinations) is randomly generated, denoted
as the initial population P(0). Each candidate solution Xi is represented as a vector:

$$X_i = \{h_1, h_2, \ldots, h_n\}$$

where hi represents the i-th hyperparameter.
(2) Fitness evaluation: Individual x_i in the population is evaluated, and its fitness value $f(x_i)$ is calculated to quantify the quality
of the solution. The fitness function is defined based on the model's performance indicators:

$$f(x_i) = \text{Performance}(x_i)$$

where Performance(xi) represents the model's performance with the hyperparameter combination x_i.

(3) Selection: Individuals are selected based on their fitness values, with those possessing superior fitness exhibiting a higher
probability of being chosen for reproduction. The probability formula for roulette wheel selection is formulated as:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

where Pi is the probability of selecting individual xi, and the summation is over all individuals in the population.

(4) Crossover: Selected individuals undergo crossover operations to generate new offspring. Assuming two parent individuals
X1 and X2, the crossover process producing offspring

can be expressed as:

$$X_1 = \{x_{1,1}, x_{1,2}, \ldots, x_{1,k}, x_{2,k+1}, \ldots, x_{2,n}\}$$
$$X_2 = \{x_{2,1}, x_{2,2}, \ldots, x_{2,k}, x_{1,k+1}, \ldots, x_{1,n}\}$$

where k is the intersection point.

(5) Mutation: A mutation operation is performed on the generated offspring to randomly perturb the values of specific genes
(hyperparameters) to increase the diversity of the population. The mutation operation can be expressed as:

X X random perturbation

(6) Population update: The offspring individuals are incorporated into the population, and the next-generation population is
selected based on fitness values. The aforementioned iterative process is repeated until the termination conditions are met
(such as reaching the maximum number of iterations or achieving a predetermined fitness value).

P t selection from P t Offsparing t (1) ( ), ( ) + = { }

where P(t) denotes the population of the t-th generation; Offspring(t) represents the new offspring generated by the t-th
generation population through crossover and mutation operations, and P(t + 1) is the next-generation population formed by
selecting individuals with higher fitness values from both the t-th generation population and the newly generated offspring.

(7) Iteration cycle: Through multiple iterations, the GA continuously performs selection, crossover, and mutation operations,
gradually approaching the optimal solution. The following formula represents the iterative process of the GA:

where, GA(P(t)) is the GA operation including selection, crossover, and mutation, which are

performed on the t-th generation
population P(t) to generate the (t + 1)-th generation population P(t + 1).

The default parameter values (e.g., lr0: 0.01, lrf: 0.01, momentum: 0.937) of the YOLOv11 architecture are adjusted after
300 iterations of training and testing. The ET optimization mechanism then generates optimized fitness values for
hyperparameter adjustment (e.g., lr0: 0.00198, lrf: 0.01686, momentum: 0.83041). The optimized hyperparameter values

obtained via the ET optimization mechanism are used as the parameter settings for Experiments 3 and 4.

4. Experiment Description and Result Analysis

This study was conducted on a Windows platform equipped with an Intel® Core™ i7-13700 processor and an NVIDIA
GeForce RTX 4070 graphics card. Four experimental groups were configured, including the YOLOv11 base model, YOLOv11
integrated with the SKA mechanism, YOLOv11 coupled with hyperparameter optimization, and YOLOv11 integrated with
both the SKA mechanism and hyperparameter optimization. Subsequently, comparative experiments were conducted on these
combined improvements of YOLOv11.

Using YOLOv11 as the base model, a system for underwater fish detection was developed by introducing data
augmentation, a feature pyramid network, and an improved SKA module. The pre-processed underwater images were first
subjected to data augmentation and then input into the YOLOv11 backbone network to extract deep features. The feature
pyramid network is employed to fuse multi-scale features. The experimental dataset was sourced from a publicly available
underwater fish dataset on Roboflow, comprising 1,690 images annotated with eight fish species. YOLOv11 performs object
detection by partitioning an image into multiple grid cells and predicting bounding boxes and class probabilities for each cell.

The model adopts an end-to-end training strategy, utilizing IoU loss for bounding box regression and cross-entropy loss for
classification.

The experimental design employed k-fold cross-validation with evaluation metrics encompassing precision, recall,
mAP@0.5, and mAP@0.5:0.95. A 95% confidence interval was calculated based on results from three experiments conducted
under identical conditions using different random seeds to ensure the stability and reproducibility of the model's quantitative
results. Moreover, the dataset employed data augmentation strategies comprising random flipping, rotation, brightness/contrast
jittering, Gaussian blur, and synthetic occlusion to

enhance sample diversity and mitigate biases caused by the limited dataset.
Lastly, the applicability of the present study's results is constrained to the relevant subsets of Roboflow.
Experiment 1 YOLOv11_origin detection

Experiment 1 utilized the default model parameters of YOLOv11 for training and testing on the underwater fish dataset.
The experimental results are presented in Table 1. In terms of performance evaluation, the model achieved a precision of
0.95856, a recall of 0.90678, an mAP@0.5 of 0.96637, and an mAP@0.5:0.95 of 0.81779. The model's Giga Floating-Point
Operations per Second (GFLOPs) were 6.3, and the frame rate (FPS) was 64.94, indicating that an average of approximately
64.94 images could be processed per second

Table 1 YOLOv11_origin loss rate & performance analysis

Item	train box_loss	train cls_loss	train dfl_loss	val box_loss	val cls_loss	val dfl_loss
loss rate	0.51996	0.35269	0.89456	0.80223	0.476	1.04032
Item	Precision	Recall	mAP@0.5	mAP@0.5:0.95	GFLOPs	FPS
Performance	0.95856	0.90678	0.96637	0.81779	6.3	64.94

Experiment 2 YOLOv11 with SKA mechanism

Experiment 2 employed the improved YOLOv11 model integrated with the SKA mechanism. This model was applied to
the underwater fish recognition task and subjected to training and testing. The experimental results are presented in Table 2.

In terms of performance evaluation, the model achieved a precision of 0.97518, a recall of 0.93637, an mAP@0.5 of 0.9792,
and an mAP@0.5:0.95 of 0.85531. The model's GFLOPs were 15.1, and the FPS was 53.12, indicating that an average of
approximately 53.12 images could be processed per second.

Table 2 YOLOv11_SKA loss rate & performance analysis

Item	train box_loss	train cls_loss	train dfl_loss	val box_loss	val cls_loss	val dfl_loss
loss rate	0.41202	0.28856	0.85324	0.71907	0.40693	1.02635
Item	Precision	Recall	mAP@0.5	mAP@0.5:0.95	GFLOPs	FPS
Performance	0.97518	0.93637	0.9792	0.85531	15.1	53.12

Experiment 3 YOLOv11_origin+ET

Experiment 3 utilized the base YOLOv11 model coupled with the ET parameter optimization method for training and
testing to perform underwater fish recognition. The experimental results are presented in Table 3. In terms of performance
evaluation, the model achieved a precision of 0.96224, a recall of 0.97034, an mAP@0.5 of

0.98551, and an mAP@0.5:0.95
of 0.86903. The model's GFLOPs were 6.3, and the FPS was 74.07, indicating that an average of approximately 74.07 images
could be processed per second.

Table 3 YOLOv11_origin+ET loss rate & performance analysis

Item	train box_loss	train cls_loss	train dfl_loss	val box_loss	val cls_loss	val dfl_loss
loss rate	0.4034	0.39197	1.02284	0.84671	0.6506	1.32329
Item	Precision	Recall	mAP@0.5	mAP@0.5:0.95	GFLOPs	FPS
Performance	0.96224	0.93034	0.97551	0.86903	6.3	74.47

Experiment 4 YOLOv11_SKA+ET

Experiment 4 applied the YOLOv11 model, incorporating both the SKA-based architectural modification and the ETimproved hyperparameter optimization, to the underwater fish recognition task. The experimental results are shown in Table
4. In terms of performance evaluation, the model achieved a precision of 0.99069, a recall of 0.95968, an mAP@0.5 of 0.98629,
and an mAP@0.5:0.95 of 0.86933. The model's GFLOPs were 15.0, and the FPS was 58.56, indicating that an average of
approximately 58.56 images could be processed per second.

Table 4 YOLOv11_SKA+ET loss rate & performance analysis

Item	train box_loss	train cls_loss	train dfl_loss	val box_loss	val cls_loss	val dfl_loss
loss rate	0.32581	0.32889	0.99756	0.76278	0.58998	1.29803
Item	Precision	Recall	mAP@0.5	mAP@0.5:0.95	GFLOPs	FPS
Performance	0.99069	0.95968	0.98629	0.86933	15.0	58.56

Quantitative analysis reveals that, after introducing the

SKA+ET mechanism, an increase in GFLOPs and a decrease in
FPS were observed. However, given the low visibility characteristics of underwater environments, prioritizing speed over
accuracy could result in mislocalization or missed detections, potentially leading to more severe failures. Moreover, the
proposed SKA+ET model achieves an FPS of 58.56, which surpasses the 30 FPS standard typically required for real-time
underwater video streams. Overall, the experimental results indicate that this study meets the requirements for near-real-time
underwater applications while strongly validating its feasibility for future underwater drone deployment.

Experiment 5 Performance comparison

(1) Performance comparison of four approaches:

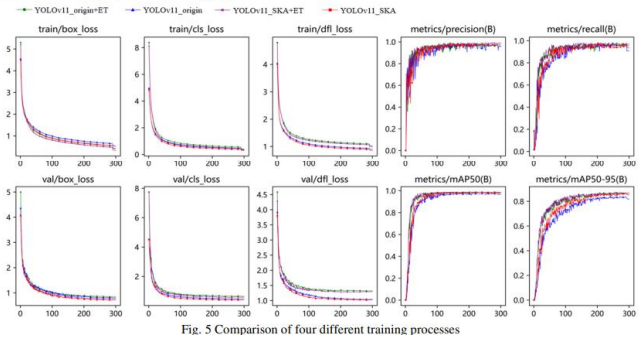


Fig. 5 Comparison of four different training processes

Finally, a comprehensive analysis of the results from the four experiments was conducted. Fig. 5 illustrates the trends of
the loss functions during the training process for the four configurations. The results show that the YOLOv11_SKA model
achieved the best performance in terms of both training loss and validation loss. Regarding performance metrics, the

YOLOv11_SKA+ET model attained the highest precision, recall, mAP@0.5, and mAP@0.5:0.95. Overall, the
YOLOv11_SKA+ET model outperformed the other three models. These results indicate that the SKA mechanism and hyperparameter optimization played a positive role in improving model performance. It is also noted that the characteristics of
different datasets may influence the performance of different models

Table 5 presents a comparative analysis of the loss rates and performance metrics derived from the training process of the
four distinct methods

Loss function: The training loss of YOLO11_SKA was the lowest, indicating that this model achieved the best convergence
during training and exhibited the lowest risk of overfitting. The validation loss was also the lowest for YOLOv11_SKA,
demonstrating strong generalization capability on the validation set.

Performance metrics: YOLOv11_SKA+ET achieved the highest mAP@0.5 and mAP@0.5:0.95 values, signifying the model's
highest accuracy in object detection. Regarding precision, YOLOv11_SKA+ET exhibited the highest precision, reflecting a
minimal false positive rate. Regarding recall,

YOLOv11_SKA+ET demonstrated a comparatively higher recall rate, suggesting
the model's enhanced ability to detect a greater number of targets.

Model complexity and speed: The GFLOPs of YOLOv11_SKA and YOLOv11_SKA+ET were significantly higher than those
of the other two models, indicating greater model complexity. Regarding image processing speed, YOLOv11_origin exhibited
the highest FPS. The FPS of YOLOv11_SKA and YOLOv11_SKA+ET were relatively lower, suggesting that the
incorporation of the SKA mechanism increases the model's computational load, thereby reducing inference speed.

Table 5 Comparison of churn & effectiveness analysis of four different models

Item	train box_loss	train cls_loss	train dfl_loss	val box_loss	val cls_loss	val dfl_loss
YOLOv11_origin	0.51996	0.35269	0.89456	0.80223	0.476	1.04032
YOLOv11_origin+ET	0.4034	0.39197	1.02284	0.84671	0.6506	1.32329
YOLOv11_SKA	0.41202	0.28856	0.85324	0.71907	0.40693	1.02635
YOLOv11_SKA+ET	0.32581	0.32889	0.99756	0.76278	0.58998	1.29803
Item	Precision	Recall	mAP@0.5	mAP@0.5:0.95	GFLOPs	FPS
YOLOv11_origin	0.95856	0.90678	0.96637	0.81779	6.3	64.94
YOLOv11_origin+ET	0.96224	0.93034	0.97551	0.86903	6.3	74.07
YOLOv11_SKA	0.97518	0.93637	0.9792	0.85531	15.1	53.12
YOLOv11_SKA+ET	0.99069	0.95968	0.98629	0.86933	15.0	58.56

Fig. 6 illustrates the progression of the mAP@0.5 metric during the training process for the YOLOv11 model under four
distinct configurations

YOLOv11_origin: The mAP@0.5 of the original YOLOv11 model increased rapidly during the early training phase. However,
it soon reached a plateau, indicating limitations in the model's learning capacity. The model achieved an mAP@0.5 of 0.96637,
which, while satisfactory, serves as a baseline for subsequent comparisons.

YOLOv11_origin+ET: Based on the original model, hyperparameter optimization was implemented to identify an optimal set
of hyperparameters for enhanced model performance. As shown in Fig. 6, this optimization strategy yielded a significant
improvement in model performance. The peak mAP@0.5 exceeded that of the original model, and the convergence rate was
accelerated. Specifically, the mAP@0.5 reached 0.97551, representing a 0.914% increase compared to the baseline model.
These results validate the efficacy of ET in hyperparameter optimization.

YOLOv11_SKA: The SKA mechanism was integrated into the original model to enhance its ability to focus on target regions,
thereby improving detection accuracy. The integration of the SKA mechanism increased mAP@0.5, although the magnitude
of improvement was less pronounced than that observed with hyperparameter optimization. The model achieved an mAP@0.5
of 0.9792, representing a 1.283% improvement over the baseline model. This result confirms the positive impact of the SKA
mechanism on model performance.

YOLOv11_SKA+ET: A comprehensive optimization of the original YOLOv11 model was achieved by combining hyperparameter optimization enabled by ET and the architectural adjustment provided by the SKA mechanism. As shown in
Fig. 6, this combined approach yielded the best results, exhibiting the highest peak mAP@0.5 and enhanced generalization
capabilities. The model achieved an mAP@0.5 of 0.98629, the highest among the four configurations, representing a 1.992%
improvement over the baseline model. These results underscore the synergistic effect of combining hyperparameter
optimization and the SKA mechanism in achieving optimal model performance.

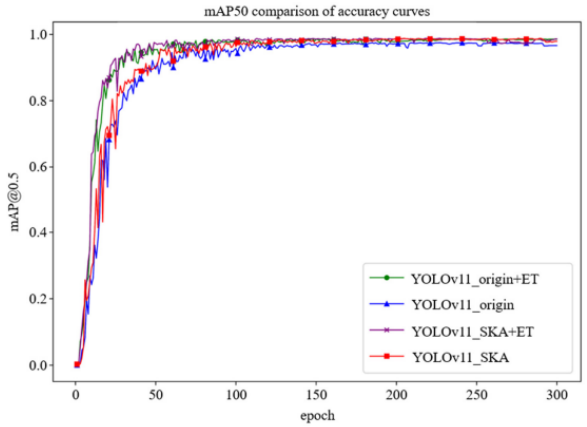


Fig. 6 Comparison of mAP@0.5 indicators in four different ways

Fig. 7 presents the trajectory of the mAP at IoU thresholds ranging from 50% to 95% (mAP@0.5:0.95) during the training
process for the YOLOv11 model under four distinct configurations. The mAP@0.5:0.95 metric is a crucial indicator for
evaluating the performance of object detection models. It represents the average precision calculated across IoU thresholds
from 50% to 95%. Compared to mAP@0.5, this metric provides a more comprehensive assessment of the model's performance
across various IoU thresholds and is particularly sensitive to the detection effectiveness of small objects. A comparative
analysis of the four configurations is presented below:

YOLOv11_origin: The mAP@0.5:0.95 of the original YOLOv11 model increased rapidly during the early training phase.
However, it soon reached a plateau, indicating limitations in the model's learning capacity. The model achieved an

mAP@0.5:0.95 of 0.81779, which, while adequate, serves as a baseline for subsequent comparisons.
YOLOv11_origin+ET: Based on the original model, hyperparameter optimization was implemented using the ET mechanism.
This approach was designed to identify an optimal set of hyperparameters to enhance model performance. As shown in Fig. 7,
ET-driven hyperparameter optimization resulted in a significant improvement. The peak mAP@0.5:0.95 exceeded that of the
original model, and the convergence rate was accelerated. Specifically, the mAP@0.5:0.95 reached 0.85903, representing a
4.124% increase compared to the baseline YOLOv11 model. These results validate the effectiveness of hyperparameter
optimization in enhancing model performance.

YOLOv11_SKA: The SKA mechanism was integrated into the original model to enhance its ability to focus on target regions,
thereby improving detection accuracy. The introduction of the SKA mechanism increased mAP@0.5:0.95, although the
magnitude of improvement was less pronounced than that observed with hyperparameter optimization. The model achieved
an mAP@0.5:0.95 of 0.85531, representing a 3.752% improvement over the baseline model. This result confirms the positive
impact of the SKA mechanism on model performance.

YOLOv11_SKA+ET: A comprehensive optimization of the original YOLOv11 model was achieved by combining ET-driven
hyperparameter optimization with the architectural adjustment provided by the SKA mechanism. As shown in Fig. 7, this

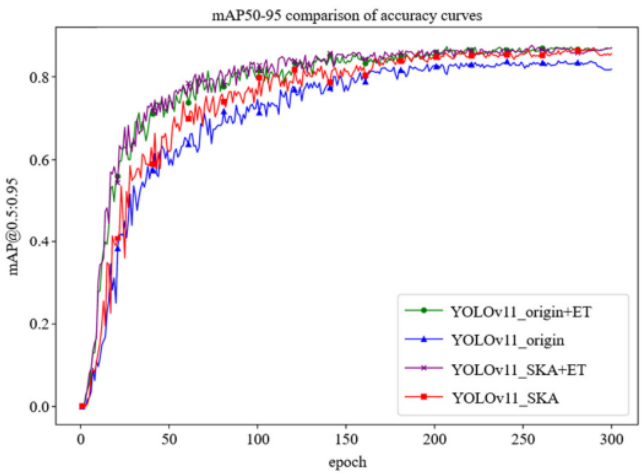


Fig. 7 Comparison of mAP@0.5:0.95 indicators in four different ways

Fig. 8 illustrates the inference predictions, highlighting the significant advantages of the YOLOv11_SKA+ET model in
terms of practical detection performance. Qualitative evaluation through comparison with three mainstream detection models
demonstrates that the YOLOv11_SKA+ET model exhibits superior adaptability across diverse detection tasks within complex
underwater environments, providing enhanced detection performance specifically for low-resolution small underwater fish
targets, partially occluded regions, and varying illumination conditions.

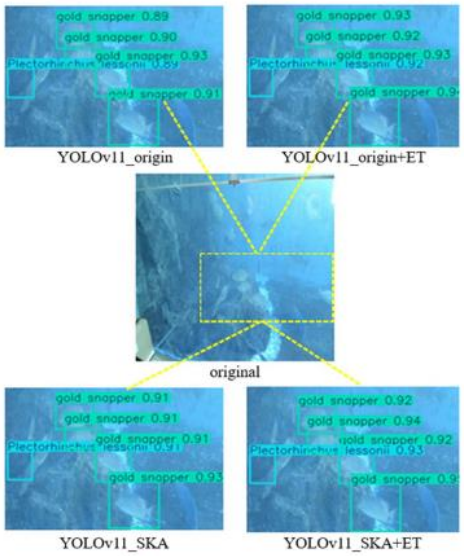


Fig. 8 Comparison of the detection effects of four different methods

combined approach yielded the best results, exhibiting the highest peak mAP@0.5:0.95 and enhanced generalization
capabilities. The model achieved an mAP@0.5:0.95 of 0.86933, the highest among the four configurations, representing a
5.154% improvement over the baseline model. These results highlight the synergistic effect of combining hyperparameter
optimization and the SKA mechanism in achieving optimal model performance

(3) Performance comparison with other methods:

This study introduces an improved YOLOv11 model designed to enhance underwater fish detection

performance through
the incorporation of ET hyperparameter optimization and architectural adjustments using the SKA mechanism. The proposed
model achieved a mAP of 98.629%, significantly surpassing existing baselines. Notably, substantial improvements were
achieved in small object detection, occlusion handling, and robustness under varying illumination conditions, demonstrating
the advantages of the proposed approach. A performance comparison of different methods for underwater fish detection is
presented in Table 6.

Table 6 Performance comparison of different methods in underwater fish detection

Method	mAP
Villon et al. [16]	94.90%
Jalal et al. [17]	91.64% and 79.8%
Liu et al. [18]	97%
Wu et al. [19]	97.6%
Rathi et al. [26]	96.29%
Wibowo and Utama [27]	90.50%
The proposed method (SKA+ET)	98.63%

Compared to previous related studies, the main advantages and contributions of the proposed method include:
Enhanced small object detection capability: The YOLOv11 model, incorporating the SKA mechanism, is more effective at
focusing on small target regions, thereby improving overall detection accuracy. This mechanism, which dynamically adjusts
convolution kernels of different sizes, enables the model to precisely capture subtle features of small objects, enhancing small
object detection performance. Compared to traditional attention mechanisms, SKA places greater emphasis on small objectrelated contextual information, reducing background noise interference and further improving detection accuracy.
More robust occlusion handling: The SKA mechanism, in conjunction with the ET hyperparameter optimization mechanism
and a tailored data augmentation strategy, enables the model to more accurately predict the position and size of occluded or
overlapping targets. The data augmentation strategy simulates diverse occlusion scenarios, enhancing the model's robustness
and generalization ability in real-world scenes.
Improved illumination invariance: The ET hyperparameter optimization mechanism effectively enhances the model's
adaptability to varying illumination conditions. By automatically searching for the optimal hyperparameter combination, the
model demonstrates superior performance when processing underwater images with significant illumination variations,
reducing the impact of lighting factors on detection results.

4. Conclusion

This study successfully enhanced the performance of an underwater fish detection model by introducing a dualimprovement strategy to the YOLOv11 architecture. The research integrated an SKA mechanism for dynamic feature channel
weighting and employed an ET for fine-grained hyperparameter optimization. The synergistic effect of these two
enhancements resulted in a model with superior accuracy and robust detection capabilities in complex underwater
environments. The key findings and conclusions of this study are as follows:

(1) Significant performance gains: The improved model demonstrated substantial improvements over the original YOLOv11
model across all key metrics. Specifically, gains of 2.06% in mAP@0.5 and 6.30% in mAP@0.5:0.95 were observed with
final scores reaching 98.629% and 86.933%, respectively. Precision and recall also experienced increases of 3.35% and
5.83%, reaching final values of 99.069% and 95.968%.
(2) Effectiveness of the SKA mechanism: The SKA mechanism effectively enhances the model's ability to focus on target
regions by adaptively adjusting feature channel weights, thereby improving overall detection accuracy.

(3) Trade-off with computational complexity: While the SKA mechanism successfully elevated accuracy, it also led to an
increase in the model's computational complexity (GFLOPs), resulting in a slight reduction in inference speed (FPS).
(4) Importance of Hyperparameter Optimization: The use of an ET for hyperparameter optimization proved crucial in further
improving the model's performance and mitigating the

trade-off between accuracy and speed.

In summary, given the aforementioned improvements, the proposed dual-enhanced YOLOv11 model offers a highly
competitive and effective solution for underwater fish detection, with promising applications for marine ecological monitoring.
Future work will involve systematic validation using larger and more diverse public datasets (e.g., Fish4Knowledge,
LifeCLEF), as well as the exploration of semi-supervised learning and domain adaptation strategies to improve performance
on rare species and challenging long-tail data, thereby further evaluating and enhancing the model's generalization ability. In
addition, research on domain adaptation will be expanded to address image variations across different aquatic environments
(such as freshwater and deep sea). Finally, the integration of multi-modal sensing will be explored, enabling fusion with other
sensor data such as sonar and depth sensors to improve robustness in complex detection scenarios.

Acknowledgments

This work was supported by the Scientific Research Foundation for Advanced Talents of Xiamen Ocean Vocational
College (grant number: KYG202205).

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] C. M. Crain, B. S. Halpern, M. W. Beck, and C. V. Kappel, "Understanding and Managing Human Threats to the
Coastal Marine Environment," Annals of the New York Academy of Sciences, vol. 1162, no. 1, pp. 39-62, 2009.
[2] S. P. González-Sabbagh and A. Robles-Kelly, "A Survey on Underwater Computer Vision," ACM Computing Surveys,
vol. 55, no. 13s, pp. 1-39, 2023.
[3] R. H. Thurstan, S. Brockington, and C. M. Roberts, "The Effects of 118 Years of Industrial Fishing on UK Bottom
Trawl Fisheries," Nature Communications, vol. 1, article no. 15, 2010.
[4] Y. Wang, J. Zhang, Y. Cao, and Z. Wang, "A Deep CNN Method for Underwater Image Enhancement," IEEE
International Conference on Image Processing, pp. 1382-1386, 2017.
[5] S. Mittal, S. Srivastava, and J. P. Jayanth, "A Survey of Deep Learning Techniques for Underwater Image Classification," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 10, pp. 6968-6982, 2023.
[6] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, et al., "An Underwater Image Enhancement Benchmark Dataset and
Beyond," IEEE Transactions on Image Processing, vol. 29, pp. 4376-4389, 2019.
[7] C. Yang, J. Xiang, X. Li, and Y. Xie, "FishDet-YOLO: Enhanced Underwater Fish Detection with Richer Gradient

Flow and Long-Range Dependency Capture through Mamba-C2f," Electronics, vol. 13, no. 18, article no. 3780, 2024.
[8] M. Kaur and S. Vijay, "Deep Learning with Invariant Feature Based Species Classification in Underwater Environments," Multimedia Tools and Applications, vol. 83, no. 7, pp. 19587-19608, 2024.
[9] X. Qin, C. Yu, B. Liu, and Z. Zhang, "YOLO8-FASG: A High-Accuracy Fish Identification Method for Underwater
Robotic System," IEEE Access, vol. 12, pp. 73354-73362, 2024.
[10] Y. H. Liu, "Feature Extraction and Image Recognition with Convolutional Neural Networks," Journal of Physics:
Conference Series, vol. 1087, no. 6, article no. 062032, 2018.
[11] S. Anwar, C. Li, and F. Porikli, "Deep Underwater Image Enhancement,"
https://doi.org/10.48550/arXiv.1807.03528,

2018.
[12] M. A. Iqbal, Z. Wang, Z. A. Ali, and S. Riaz, "Automatic Fish Species Classification Using Deep Convolutional Neural
Networks," Wireless Personal Communications, vol. 116, no. 2, pp. 1043-1053, 2021.
[13] X. Chen, P. Zhang, L. Quan, C. Yi, and C. Lu, "Underwater Image Enhancement Based on Deep Learning and Image
Formation Model,"
https://doi.org/10.48550/arXiv.2101.00991, 2021.