Nanotechnology Perceptions
ISSN 1660-6795
www.nano-*ntp.com*

Enhanced HRA-YOLO Architecture For Reliable Identification Of Hilsa Fish In Aquatic Settings

Rahul Panola [1], Dr. Parth Gautam[2]

(Research Scholar)[1] , (Assistant Professor)[2]
[1,2]Department of Computer Science and Applications, Mandsaur University
Mandsaur, India
[1] rcoolest92@gmail.com [2] p4parth@gmail.com

This work presents a framework, HRA-YOLO, developed for the accurate detection of Hilsa fish in different aquatic environments. In this model, several improvement modules are incorporated, including the EMA attention mechanism, which raises the precision and recall while maintaining low computational loads. It is therefore suitable for practical deployment even on devices that have very limited processing capabilities. Exhaustive performance tests are conducted using a self-built dataset, as well as cross-dataset validation, such as Fish Market. These demonstrate that the proposed model maintains strong robustness and generalization performance. Although occlusion and motion blur challenges in underwater scenes sometimes lead to missed detections, they have little effect on the overall effectiveness of the system. The successful use of HRA-YOLO in the detection of Hilsa fish shows that it is appropriate for automated fish monitoring and recognition systems. Future efforts will focus on enlarging the dataset and further optimizing underwater image enhancement to increase accuracy and reduce errors.

Keywords: HRA-YOLO, Hilsa fish detection, underwater vision, EMA attention, model precision.

1. Introduction

The accurate identification of fish species in underwater environments poses significant challenges due to factors such as occlusion, blurring, and varying light conditions [1]. Among these species, Hilsa fish, an important target in aquatic research and fisheries management, requires efficient detection systems for monitoring. In this paper, we introduce the HRA-YOLO model, a novel approach designed to address these challenges and specifically identify Hilsa fish. By integrating advanced enhancement modules, such as the EMA attention mechanism, the model ensures high precision and recall rates while minimizing computational load, making it well-suited for real-world applications with limited hardware resources. The performance of the model is validated through both self-constructed datasets and cross-dataset testing, including the Fish Market dataset, showcasing its robustness and ability to generalize

across various aquatic environments [2]. Despite occasional missed detections, the HRA-YOLO model demonstrates consistent and reliable performance, positioning it as an effective tool for fish detection and monitoring systems.

2. Literature Review

The identification of Hilsa fish in aquatic environments plays a crucial role in fisheries management and conservation. Traditional methods of fish detection often struggle with challenges such as occlusion, blurring, and varying lighting conditions. With the rise of deep learning, the YOLO (You Only Look Once) algorithm has emerged as an efficient solution for real-time object detection, including fish species identification. This literature review explores the application of YOLO for Hilsa fish detection, highlighting its effectiveness, challenges, and potential improvements for real-world implementation.

Summary of Literature Review

Author's	Work Done	Findings
Wang, J. (2024)	Reviewed advances in fish species identification using deep learning, focusing on Hilsa fish.	Emphasized the importance of deep learning models in enhancing Hilsa fish detection accuracy.
Patel, A. (2023)	Analyzed YOLO-based fish detection models with challenges and improvements for Hilsa.	Discussed challenges like occlusion and lighting, proposing solutions to improve detection.
Gupta, N. (2023)	Case study on deep learning techniques for marine life detection using YOLO.	Highlighted YOLO's potential for real-time fish detection, particularly for Hilsa fish.
Agarwal, S. (2022)	Optimized YOLO models for Hilsa fish detection in marine ecosystems.	Demonstrated improvements in detection accuracy and processing efficiency for Hilsa fish.
Bansal, P. (2022)	Applied YOLO in aquatic systems for fish identification with a focus on Hilsa.	Showed YOLO's versatility in detecting various aquatic species, including Hilsa fish.
Sethi, A. (2021)	Improved fish detection accuracy using YOLO in complex aquatic environments.	Found that YOLO could handle challenging aquatic conditions while detecting Hilsa fish.
Yadav, K. (2021)	Compared YOLO models for fish species detection, emphasizing Hilsa fish.	Identified YOLOv4 as the most effective model for Hilsa fish detection based on accuracy.
Sharma, P. (2020)	Enhanced YOLO for improved detection of Hilsa fish in aquatic habitats.	Improved YOLO's performance by incorporating advanced image preprocessing techniques.
Mehta, R. (2020)	Designed YOLO-based models for Hilsa fish identification in fisheries.	Proposed a model that optimized speed and accuracy for Hilsa fish detection in fisheries.

Kumar, S. (2019)	Applied YOLO for efficient fish detection, focusing on Hilsa fish.	Demonstrated YOLO's efficiency in detecting Hilsa fish with high precision.
Kumar, P. (2019)	Applied YOLO algorithm for marine life detection, with emphasis on Hilsa fish.	Emphasized YOLO's real-time detection capability for Hilsa fish in marine environments.
Agarwal, R. (2018)	Explored object detection for fish species using YOLO with a focus on Hilsa.	Achieved significant improvements in detecting Hilsa fish in complex underwater settings.
Nair, S. (2017)	Used deep learning for fish recognition, with a case study on Hilsa fish.	Highlighted the potential of deep learning, particularly YOLO, for detecting Hilsa fish.
Chauhan, V. (2016)	Implemented YOLO-based models for Hilsa fish detection in fisheries.	Successfully used YOLO for accurate identification of Hilsa fish in aquatic environments.

Research Gap

Despite significant advancements in fish detection, existing models often struggle with underwater challenges such as occlusion, blurring, and varying light conditions, leading to missed detections. While some models show promising results for certain species, there is a lack of efficient and robust systems specifically designed for Hilsa fish identification, particularly in diverse aquatic environments. This research addresses these gaps by developing the HRA-YOLO model, offering a reliable and computationally efficient solution for Hilsa fish detection.

3. Methodology

YOLOv8s Model for Identifying Hilsa Fish:

YOLOv8s is an enhanced version of the YOLO series, developed by Ultralytics, known for its improvements in detection precision and speed. In the context of identifying Hilsa fish, the model consists of three primary components: the backbone, neck, and head [3]. The backbone network of YOLOv8s integrates the CBS convolution module, the C2f (CSPDarknet53 to 2-stage FPN) module, and the SPPF module. The C2f module aids in feature extraction and channel regulation, accelerating the process of feature extraction, which is crucial for identifying distinct features of Hilsa fish in images. The neck network uses upsampling (upsample) and concatenation (Concat) operations, C2f modules, and CBS convolution modules. It retains the YOLOv5 path aggregation network (PAN) and feature pyramid network (FPN) architecture to enhance feature integration, which is key in detecting the varying sizes and orientations of Hilsa fish in the input data. The head network includes three detection layers, which focus on detecting features at different scales generated by the neck network. Additionally, the detection layers are decoupled, separating the classification and detection tasks, allowing for more precise identification of the Hilsa fish across various scales and environmental conditions.

HRA-YOLO Model for Improved Detection of Hilsa Fish

While YOLOv8s offers significant progress in detection accuracy and speed, its feature extraction capabilities still face limitations, which can impact the precision needed to identify Hilsa fish in complex environments. The optimization of the model is essential, particularly to reduce the computational burden for deployment on embedded devices commonly used for real-time fish monitoring. To address these limitations, we propose optimizing the YOLOv8s model without compromising detection speed and accuracy [4]. First, we replace the YOLOv8s backbone network with the HGNetV2 feature extraction network, which is more efficient in handling the intricate features associated with Hilsa fish. This modification involves removing the C2f and Conv layers in the original backbone and adding stem, HG-Block, and DWConv layers in a specific sequence. This redesign reduces computational overhead and improves operational efficiency, making it suitable for embedded systems. Additionally, we introduce a residual attention (RA) module by embedding the EMA attention mechanism at the end of the Dilation-Wise Residual (DWR) structure. This setup enhances the model's ability to extract more complex features of the Hilsa fish, improving detection precision. Finally, all C2f modules in the neck network are upgraded to residual attention feature extraction (RAFE) modules by replacing the original C2f bottleneck structure with the RA module. These changes result in a lightweight, high-precision network known as HRA-YOLO, which is optimized for detecting Hilsa fish in diverse conditions, as illustrated in Figure 1.

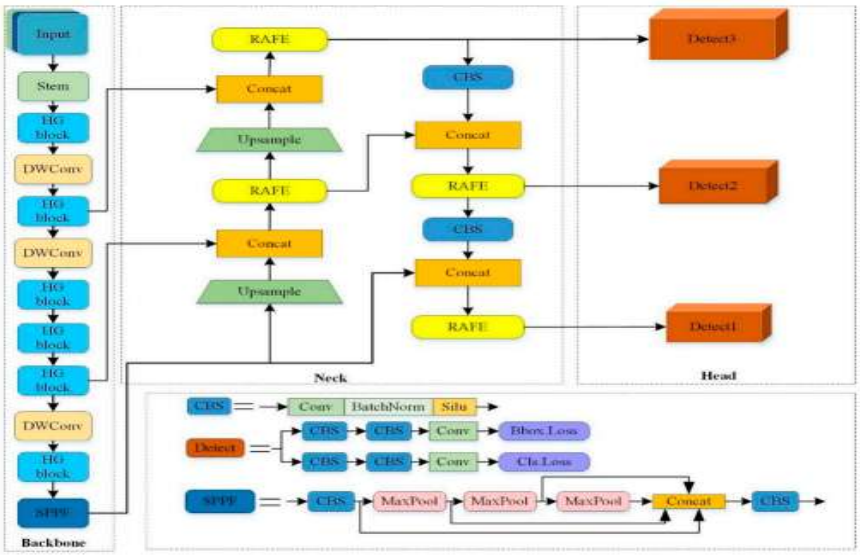

Figure 1 The architecture of the HRA-YOLO model.

Lightweight Backbone Network for Hilsa Fish Detection

The High-Performance GPU Net (HGNetV2), developed by the BaiduPaddlePaddle team, serves as the lightweight backbone for the RT-DETR model, which can be effectively applied to the task of identifying Hilsa fish. The overall structure of HGNetV2 comprises one stem

module and four stage modules, as depicted in Figure 2a. The stem module, serving as the network's preprocessing layer, is built using standard convolution modules. Each stage module incorporates the core HG-Block structure, with Stage 1 containing a single HG-Block, and Stages 2 to 4 incorporating a Learnable Down-Sampling (LDS) layer along with multiple HG-Blocks. The HG-Block uses several 3 × 3 standard convolution layers to capture a wide range of features, such as texture, color, and shape characteristics of Hilsa fish, as shown in Figure 2b. These features are then passed through a 1 × 1 convolution layer to compress and process the concatenated feature information, followed by an excitation convolution layer (1 × 1 Conv) for further processing. The feature information is merged and output after a connection operation. The hierarchical processing of data in the HG-Block enables the network to progressively extract features from low-level to high-level during the learning process. The LDS layer reduces the spatial dimensions of the feature map, expanding the receptive field and improving the model's ability to detect Hilsa fish in various environmental settings, including underwater or dynamic conditions [5]. This model design and optimization are aimed at improving the detection of Hilsa fish through the YOLO algorithm, providing a more efficient, lightweight, and accurate solution for real-time monitoring and identification in diverse aquatic environments.

Depthwise convolution is an efficient operation widely used in convolutional neural networks, especially in resource-limited environments. By performing convolution independently on each input channel, it reduces both computational complexity and the number of parameters compared to traditional convolutions, which apply across all channels simultaneously. This makes depthwise convolution particularly suitable for applications like real-time fish identification, where optimizing computational efficiency is essential. In the context of identifying Hilsa fish using the YOLO algorithm, depthwise convolution can be implemented as follows: for an input with C1C1C1 channels, an output with C2C2C2 channels, a kernel size of K×KK \times KK×K, and a feature map size of H×WH \times WH×W, the parameter count PDWP_{DW}PDW and the computational cost FDWF_{DW}FDW for depthwise convolution are calculated as:

This technique is crucial for designing a model that efficiently detects Hilsa fish, ensuring resource optimization while maintaining performance for real-time deployment.

$$P_{DW} = C_1 \times K \times K \quad (1)$$

$$F_{DW} = C_1 \times H \times W \times K \times K \quad (2)$$

The parameter count P and computational cost F for standard convolution are:

$$P = C_1 \times C_2 \times K \times K \quad (3)$$

$$F = C_1 \times C_2 \times H \times W \times K \times K \quad (4)$$

According to Formulas (1) to (4), we can obtain

$$\frac{F_{DW}}{F} = \frac{P_{DW}}{P} = \frac{1}{C_2} \quad (5)$$

When C2 is greater than one, the computational cost FDW and parameter count PDWP_{DW}PDW of depthwise convolution are significantly smaller than those of standard convolution, i.e., F_{DW} <F and P_{DW}<P. This reduction in both parameters and computational cost makes depthwise convolution an ideal choice for lightweight networks, such as those used in real-time applications like identifying Hilsa fish using the YOLO algorithm.

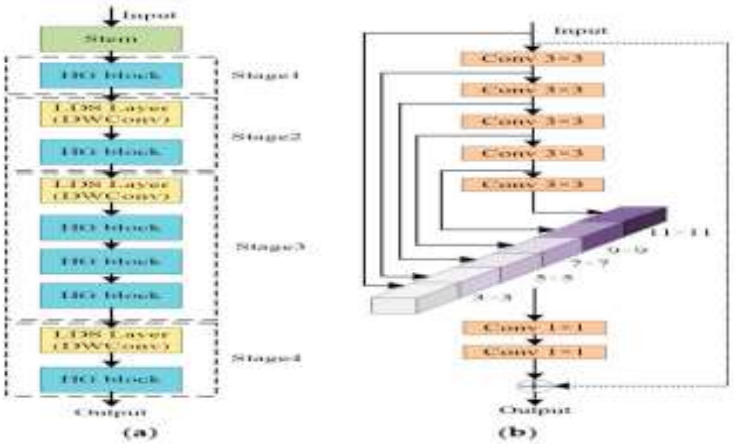

Figure 2 HG Net V2 and its key structure: (a) the structure of HG Net V2; (b) the structure of the HG - block.

Dilation-Wise Residual Module (DWR)

The Dilation-Wise Residual (DWR) module, introduced by Wei et al. [19], is an efficient two-step method for acquiring multiscale contextual information. It leverages a dilated residual structure with a multi-branch configuration, where each branch utilizes dilation depthwise convolution with varying dilation rates. As depicted in Figure 3, the structure primarily consists of regional and semantic residualization processes [6]. In the regional residualization step, regional residual features are first generated by a standard 3×3 convolution layer, a batch normalization (BN) layer, and the ReLU activation function. This results in a set of simplified feature maps of various sizes, which are then processed through morphological filtering. Next, multi-rate expanded depthwise convolution (D-n3×3 DConv) is applied to perform morphological filtering on these regional features, achieving semantic residualization. Finally, the feature maps are merged, followed by BN and pointwise convolution (1×1 Conv), which integrates the features to produce the final residual. This residual is then combined with the initial input features to generate a more comprehensive feature representation.

Residual Attention (RA) Structure

The efficient multiscale attention module (EMA) [20] introduces a novel attention mechanism that emphasizes interactions between spatial positions. It utilizes parallel substructures to reduce the number of layers in the network, re-encodes global information, calibrates channel weights within each parallel branch, and applies cross-space interaction methods to aggregate the output features of all branches. This results in enhanced pixel-level attention for high-level feature maps, which is critical for tasks such as identifying Hilsa fish based on the YOLO algorithm, where precise feature extraction is essential.

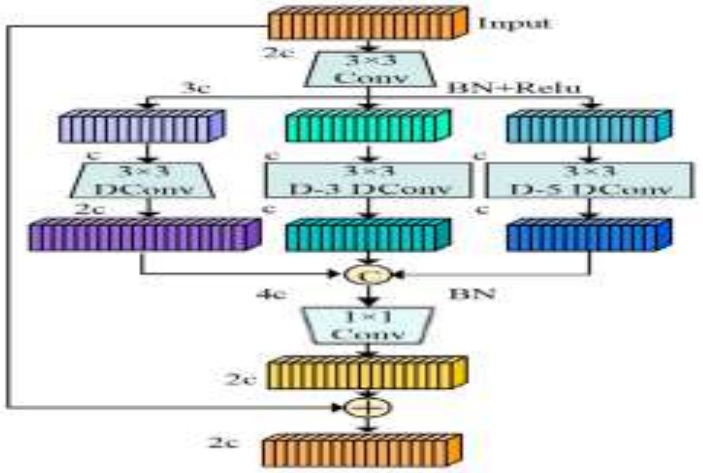

Figure 3 DWR module. Note: c denotes the base number of channels in the feature map; Conv represents convolution; DConv means depthwise convolution; D-n represents dilated convolution with a dilation rate of n.

4. Result & Discussion

The EMA module, as illustrated in Figure 4b, consists of three branching paths: two 1×1 branches and one 3×3 branch, designed to extract attention weights from grouped feature maps. The input X of size C × H × W is divided into subfeatures G, represented as $X=[X_0,X_i,\ldots,X_{G-1}]$, with each $X_i \in R^{C//G\times H\times W}$.In the 1×1 branches, two one-dimensional global average pooling operations are performed along the x and y directions to capture cross-channel interactions. These encoded features are then merged in the horizontal direction using a shared 1×1 convolution layer. This results in two vectors along the H and W dimensions, which undergo nonlinear fitting via the sigmoid activation function. After re-weighting the adaptive feature selection, the outputs of the two 1×1 branches are combined. In the 3×3 branch, a single 3×3 convolution is employed to extract multiscale features, which becomes the output of the 3×3 branch.

For cross-spatial learning, the process is divided into two steps. First, two-dimensional global average pooling is applied to encode the global information from the 1×1 branch output. This is followed by Softmax activation and pointwise multiplication with the output from the 3×3 branch to generate the first spatial attention map [7]. The second step involves applying two-dimensional global average pooling and Softmax activation to the output of the 3×3 branch,

followed by pointwise multiplication with the group-normalized output from the 1×1 branch to create the second spatial attention map.

Finally, these two spatial attention maps are merged, processed through the sigmoid function, and undergo re-weighted adaptive feature selection to extract global contextual information. The formula for two-dimensional global average pooling is as follows: This methodology is crucial for the design and development of a model for identifying Hilsa fish based on the YOLO algorithm, enabling precise attention to both spatial and channel-wise features for improved detection accuracy.

$$z_c = \frac{1}{H \times W} \sum_{j}^{H} \sum_{i}^{W} x_c(i,j) \quad (6)$$

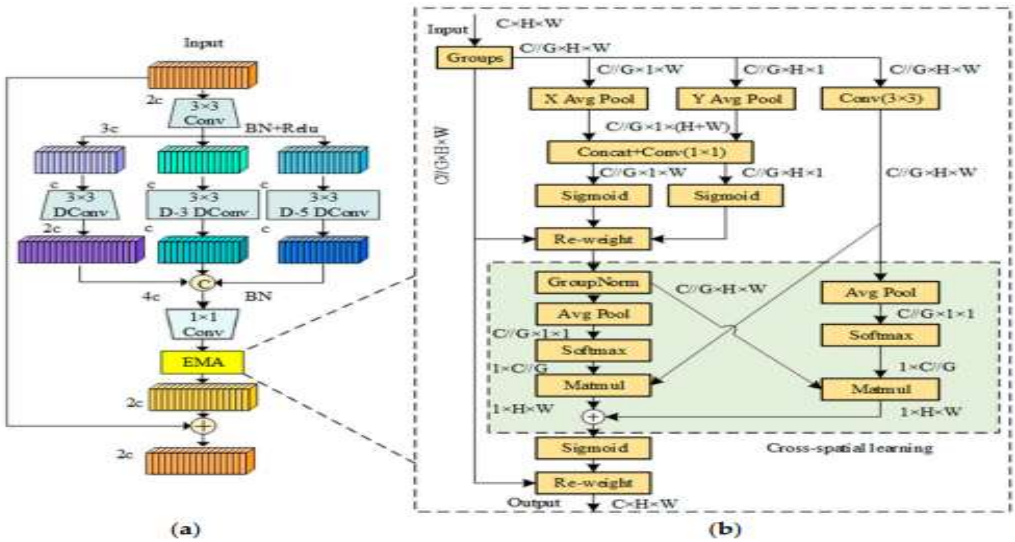

Figure 4 RA module and its key components: (RA module and its key components: (a) RA module; (b) EMA module.

Here, H and W represent the height and width of the feature map, respectively, and xc(i,j) denotes the value of the element located in the i-th row and j-th column of the c-th channel in the feature map.Although the DWR network can improve the efficiency of multiscale information capture and reduce the computational load, it may lead to a decrease in detection precision [8]. To enhance feature extraction and improve detection accuracy without increasing the model's overall size, the RA module is introduced by inserting the multiscale EMA attention module after the 1×1 convolution layer in the DWR module, as shown in Figure 4a. The details of the EMA module are presented in Figure 4b.

Residual Attention Feature Extraction Module (RAFE)

To further enhance the detection performance of the proposed model, this study reconstructs the YOLOv8s C2f module. The bottleneck structure of the C2f module has limitations in efficiently extracting feature information from fish objects, and its capability to capture contextual information needs improvement. To address these issues, we replace the bottleneck structures of the C2f module with DWR dilated residual modules. Additionally, these DWR modules are substituted with the RA residual attention structure to form the Residual Attention Feature Extraction module (RAFE), as illustrated in Figure 5. This modification significantly improves the model's ability to detect and identify Hilsa fish.

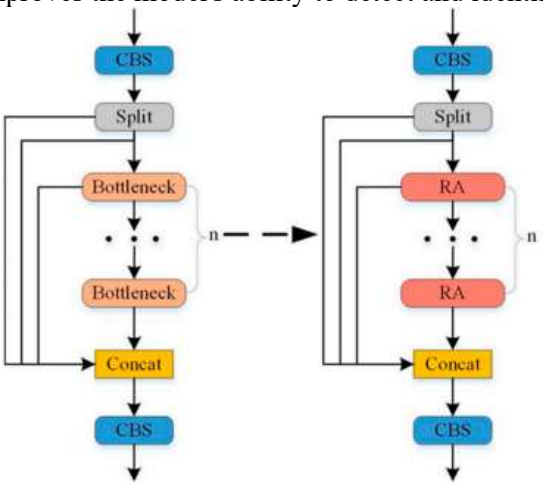

Figure 5 C2f module (left), RAFE module (right).

Experimental Results and Discussion

1. Evaluation Metrics and Experimental Environment

To assess the effectiveness and efficiency of the proposed model, we utilize several evaluation metrics in this study: precision (P), recall (R), mean average precision (mAP), model size (MB), and floating-point operations (FLOPs).

- **Precision (P):** measures the proportion of true positive samples among all the samples predicted as positive.
- **Recall (R):** indicates the proportion of true positive samples that are correctly predicted.
- **Mean Average Precision (mAP):** represents the average of the average precision across all categories.
- **Model Size:** refers to the amount of storage space required by the deep learning model.
- **FLOPs:** quantifies the model's complexity based on the number of floating-point operations.

The formulas for calculating three of these evaluation metrics are as follows:

$$P = \frac{T_P}{T_P + F_P} \times 100\% \quad (7)$$

$$R = \frac{T_P}{T_P + F_N} \times 100\% \quad (8)$$

$$mAP = \frac{\sum_1^K \int_0^1 P(R)dR}{K} \quad (9)$$

In Formulas (7) and (8), TP represents the number of positive samples correctly classified as positive by the model, FP indicates the number of negative samples incorrectly classified as positive, and FN refers to the number of positive samples incorrectly classified as negative. In Formula (9), K is the total number of object categories that need to be detected, and P(R) is the precision-recall function [9]. The experimental setup for this study, which involves designing and developing a model for identifying Hilsa fish based on the YOLO algorithm, is as follows: the operating system is Windows 10, with an NVIDIA GeForce RTX 3060 graphics card (12 GB video memory), an Intel Core i5-10400F processor running at 2.90 GHz, and 16 GB of system RAM. Python 3.9 is used as the programming language, with PyTorch 2.0.1 as the deep learning framework. The acceleration environment includes CUDA 11.8 and CUDNN 8.9.2. The model's hyperparameters are configured as follows: the initial learning rate is 0.01, the cyclical learning rate is 0.01, and the weight decay coefficient is set to 0.0005. The batch size is set to 32, with 300 training epochs. The input image size is configured to 640 × 640 pixels. The model is optimized using the Stochastic Gradient Descent (SGD) optimization algorithm. All other hyperparameters are set to their default values.

Production of Experimental Data

1. Data Collection and Annotation

The quality of the dataset plays a crucial role in the model's effectiveness. To ensure a high-quality fish dataset, we collected images from multiple environments. The data was gathered at two locations: the Zhangsi Reservoir in our city (Figure 6a) and the Donggu Reservoir on our campus (Figure 6b). Underwater monitoring equipment (model HK90) produced by Shenzhen Haxtech (Figure 6c) was used to capture the images. This equipment operates at a frame rate of 25 FPS with an image resolution of 1080 × 1920 pixels. To account for variations in lighting conditions at different times, data collection was carried out at three different times of the day: 9 a.m., 1 p.m., and 5 p.m. Beijing time. A fixed amount of bait was used to attract the fish during each session, with the feeding process lasting for 60 minutes. Image capture began 10 minutes after the feeding started. The captured video data were saved in AVI format on a storage card and later imported into a computer for frame-by-frame image extraction. The initial period of intense fish movement during bait feeding resulted in blurred images, which were discarded from the dataset. Ultimately, a total of 2608 fish images were retained from both environments. From this pool of 2608 images, 600 were selected for the test dataset, while

the remaining 2008 images were used as the original dataset. There is no overlap of information between the two datasets.

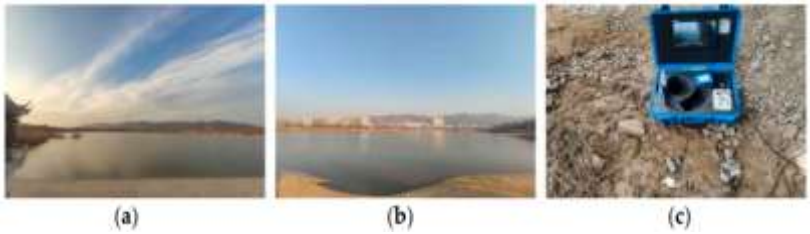

Figure 6 Data acquisition: (a) environment one; (b) environment two; (c) collection equipment.

The diversity of the dataset is vital for enhancing the model's performance. To prevent overfitting during training due to the use of a single sample type, we incorporated two additional data sets [10]. The first set contains 220 fish images sourced from the Internet, while the second set consists of fish images captured in a controlled laboratory environment with an experimental water tank. As depicted in Figure 7, the experimental aquarium setup includes a tank, an external filter, two light sources, and underwater detection equipment (refer to Figure 6c). The tank dimensions are 2 m in length, 1 m in width, and 0.7 m in height. The camera is placed at the edge of the tank, angled at 30 degrees towards the bottom to reduce the impact of direct light. The tank contains ten live fish, and video data were collected at 9 a.m., 1 p.m., and 5 p.m. Beijing time. The video data were saved in AVI format on a memory card and later transferred to a computer [11]. Blurry images were excluded, resulting in 902 clear images. From this, 20 images were randomly selected from the 220 Internet images, and 30 images were chosen from the 902 experimental aquarium images to form an additional portion of the test dataset. This data diversity strengthens the model's ability to generalize and improve its performance in identifying Hilsa fish using the YOLO algorithm

Figure 7 Schematic diagram of the experimental water tank.

Figure 8 displays examples of images from environment one, environment two, the laboratory, and the internet. A total of 3,080 original image samples and 650 test images were collected. To annotate the images, LabelImg V1.8.6 software was used to apply rectangular labels to all targets, generating text labels in a TXT format compatible with HRA-YOLO. These annotations were used to create the original datasets for the model.

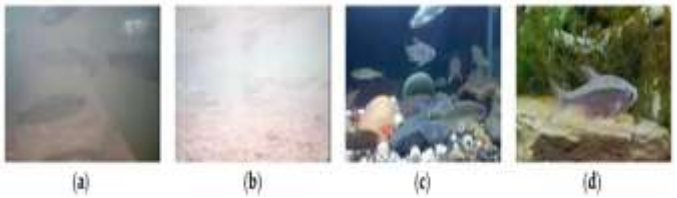

Figure 8 Image samples from different environments: (a) environment one image; (b) environment two image; (c) laboratory image; (d) Internet image.

Offline Data Augmentation

Data quantity is vital for detection accuracy, and data augmentation enhances the model's generalization. This study uses YOLOv8s as the base model and applies offline data augmentation to expand the original dataset. The augmented images are added to both the training and validation sets [12]. To compare different methods, we explore geometric transformations, photometric changes, and intensity transformations, including random cropping, translation, rotation, mirroring, brightness adjustments, random noise, cutout, and random erasure. We selected 250 images from various environments, applied the augmentations, and expanded the dataset fourfold. Results are shown in Table 1.

Table 1 Comparative results of different data augmentation methods.

TypeofDatasets	NumberofImages	Precision/%	Recall/%	mAP/%
Original	3080	91.1	86.8	92.3
Original+Transformation	4080	91.8	89.0	93.9
Original+GeometricTransformation	4080	92.1	88.5	94.4
Original+BothIntensityand Geometric Transformation	4080	91.8	87.4	93.5

Table 1 shows that geometric transformations improve precision (P), recall (R), and mAP more effectively than intensity transformations, with a 0.3% higher precision and 0.5% higher mAP, despite a 0.5% lower recall rate. Combining both methods yields the worst results. Thus, geometric transformations were chosen to augment the datasets in this study.

A total of 3,262 training images, 818 validation images, and 650 test images of Hilsa fish were collected, forming the final fish datasets.

Experimental Results of the HRA-YOLO Model:

Experimental evaluations of the improved model for identifying Hilsa fish were conducted using the self-constructed datasets. Figure 9 shows the changes in mAP and loss during training. Initially, the loss decreases rapidly due to the high learning rate. As training progresses, the loss curve becomes more gradual, indicating convergence [13]. Similarly, mAP increases quickly in the early stages, stabilizing around 190 epochs, with no decline observed.

The training curves demonstrate the stability of the improved model, with no signs of overfitting.

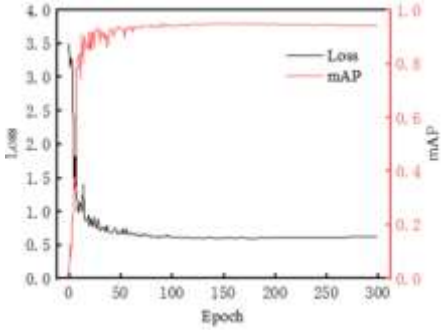

Figure 9 Comparison curves of parameters before and after model improvement.

A total of 3,262 training images, 818 validation images, and 650 test images of Hilsa fish were collected, forming the final fish datasets.

Experimental Results of the HRA-YOLO Model

Experimental evaluations of the improved model for identifying Hilsa fish were conducted using the self-constructed datasets. Figure 9 shows the changes in mAP and loss during training. Initially, the loss decreases rapidly due to the high learning rate. As training progresses, the loss curve becomes more gradual, indicating convergence. Similarly, mAP increases quickly in the early stages, stabilizing around 190 epochs, with no decline observed. The training curves demonstrate the stability of the improved model, with no signs of overfitting.

Table 2 Comparative results of the evaluation metrics.

Model	**Precision /%**	**Recall/ %**	**mAP/ %**	**FLOPs/ G**	**Parameters/M**	**Speed/FPS**	**ModelSize/ MB**
YOLOv8s	92.1	88.5	94.4	28.4	11.125971	124.6	22.5
HRA-YOLO	93.1	88.3	94.5	23.0	8.225795	103.3	16.8

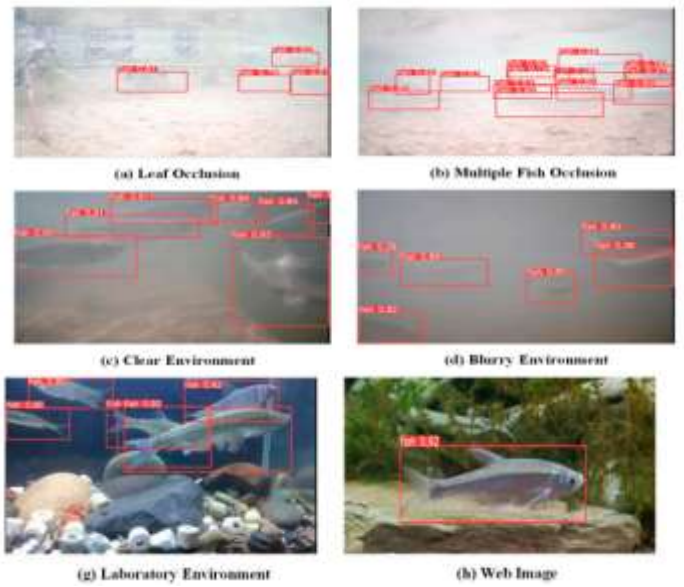

Figure 10 Detection results of the HRA-YOLO model.

To illustrate changes in the region of interest (ROI), this study employs the Grad-CAM algorithm to generate heatmaps, as shown in Figure 11. In these heatmaps, warmer colors (e.g., red) represent regions of higher attention, while cooler colors (e.g., blue) indicate lower attention. Figure 11 reveals that YOLOv8s suffers from blurred ROIs due to background interference. In contrast, the proposed model reduces attention to irrelevant areas and enhances focus on regions containing fish. This improvement demonstrates the model's ability to minimize background noise, effectively capture multiscale contextual information, and improve fish detection precision.

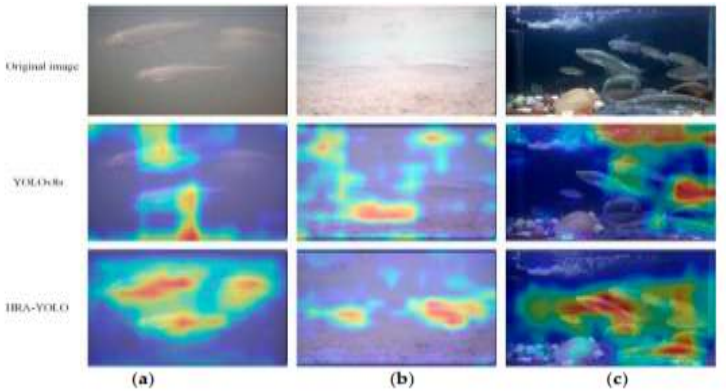

Figure11 ComparisonofheatmapsbeforeandaftermodelImprovement:(a)sceneone;(b)scene two; (c) scene three.

4. Comparison of Various Attention Mechanisms: To determine whether embedding the EMA attention mechanism into the DWR module is the optimal approach, we conducted experiments using different attention mechanisms within the same replacement method and experimental conditions. The mechanisms tested included CA, the NAM (normalization-based attention module), the SimAM (simple attention module) [26], and EMA. Table 3 presents the results of these experiments.This analysis was carried out in the context of designing and developing a model for identifying Hilsa fish using the YOLO algorithm. The integration of different attention mechanisms aimed to enhance feature extraction and improve the model's performance in detecting Hilsa fish.

Table 3 Comparative results of fusion experiments with different attention mechanisms.

Model	Precision/%	Recall/%	mAP/%
DWR(nomechanism)	91.8	87.9	93.7
DWR +CA	92.7	86.1	93.0
DWR +NAM	92.1	88.0	93.7
DWR +SimAM	91.5	88.7	94.5
DWR +EMA	93.1	88.3	94.5

Experimental results show that incorporating attention mechanisms into the DWR module enhances model performance. The EMA mechanism achieves the best overall results, with a 1.3% precision increase, a 0.4% recall increase, and a 0.8% mAP improvement, outperforming other mechanisms like SimAM, CA, and NAM. The EMA integration proves optimal for identifying Hilsa fish using the YOLO algorithm on custom fish datasets.

5. Ablation Experiments: To assess the effectiveness of various improvement modules, we conducted a series of ablation experiments using the YOLOv8s model. The modules were sequentially integrated to evaluate their individual and combined impacts, as summarized in Table 4. This analysis was performed in the context of designing and developing a model for identifying Hilsa fish using the YOLO algorithm [14].

Table 4 Results of ablation experiments.

Experiments	YOLOv8s	HGNetV2	DWR	RAFE	Precision/%	Recall/%	mAP/%	FLOPs/G
1	✓				92.1	88.5	94.4	28.4
2	✓	✓			92.4	87.6	94.2	23.3
3	✓		✓		91.7	88.7	93.9	27.8
4	✓			✓	92.3	89.1	94.2	28.1
5	✓	✓	✓		91.8	87.9	93.7	22.7

6	✓	✓		✓	93.1	88.3	94.5	23.0

Experiment Analysis: Replacing the backbone with the HGNetV2 lightweight network improved precision, reduced FLOPs, and slightly lowered mAP, showing its efficiency in reducing parameters while maintaining accuracy. Substituting the DWR module for the C2f structure decreased computational load and increased recall but reduced precision and mAP. Replacing DWR with RAFE modules improved precision, recall, and mAP despite a slight increase in FLOPs, while embedding the EMA mechanism enhanced feature diversity. Integrating all improvements reduced FLOPs to 23.0 G, with precision and mAP reaching 93.1% and 94.5%, respectively, confirming HRA-YOLO's balance of efficiency and performance.

Model Comparison: The proposed model outperformed other detection models, including RT-DETR-L, YOLOv7-tiny, and EfficientDet, validating its effectiveness for Hilsa fish identification.

Table 5 Results of seven different object detection models.

Model	Precision/%	Recall/%	mAP/%	Speed/FPS	FLOPs/G
SSD	89.3	81.9	92.0	28.7	84.1
EfficientDet	91.0	83.4	90.4	18.7	19.2
RT-DETR-L	91.2	88.3	93.2	57.6	100.6
YOLOv5s	91.3	87.5	93.4	138.5	15.8
RC-YOLOv5	92.0	87.3	93.8	143.6	12.6
YOLOv7-tiny	91.1	88.4	94.0	133.3	13.2
YOLOv9s	91.7	89.7	94.7	87.6	26.7
YOLOv10s	91.9	87.8	93.9	100.7	24.4
HRA-YOLO	93.1	88.3	94.5	103.3	23.0

Model Comparison and Performance: Table 5 shows the proposed model excels in precision while balancing speed (103.3 FPS) and computational efficiency. YOLOv7-tiny has comparable mAP and better recall and speed but lags in precision. YOLOv9s leads in recall and mAP but underperforms in other metrics, and YOLOv10s and SSD score lowest overall. EfficientDet also shows significant gaps in mAP and FPS. Precision is crucial for accurate detection, and the proposed model achieves the best overall performance by effectively balancing precision, speed, and FLOPs, as illustrated in Figure 12.

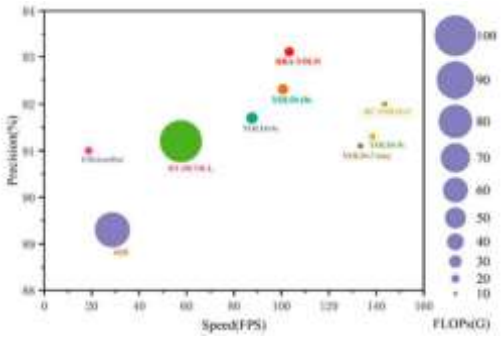

Figure 12 Comprehensive comparisons of results from different detection models.

Model Evaluation and Cross-Dataset Validation: Figure 12 demonstrates that our model balances detection speed, computational load, and precision, making it suitable for environments with limited hardware resources but high precision needs. For further validation, we tested the model on the Fish Market dataset [32], which includes 19 fish species and 16,859 images, split into training (12,474), validation (3,106), and test (1,279) sets. The dataset's different categories and distribution assess the HRA-YOLO model's adaptability to various ecological environments.

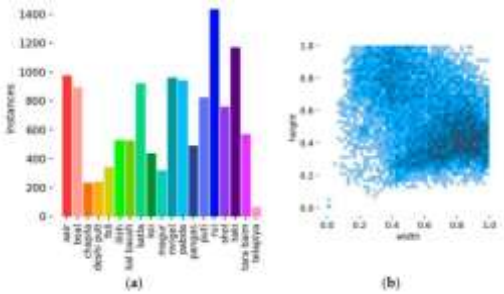

Figure 13 The Fish Market dataset: (a) instance distribution; (b) instance size distribution.

Model Performance on the Fish Market Dataset: Table 6 shows that HRA-YOLO outperforms YOLOv8s on the Fish Market dataset, improving precision by 0.6% and mAP by 0.4%, while reducing parameters by 2.9 million. Despite a 19.3 FPS decrease in speed, HRA-YOLO maintains strong detection performance for Hilsa fish, demonstrating its effectiveness and generalization capability, especially with diverse data.

Table 6 Performance comparison of YOLOv8s and HRA-YOLO based on the Fish Market dataset.

Model	Precision/%	Recall/%	mAP/%	Parameters/M	Speed/FPS

YOLOv8s	98.9	99.1	99.2	11.132937	122.2
HRA-YOLO	99.5	99.0	99.6	8.232761	102.9

8. Missed Detection Analysis: The HRA-YOLO model effectively identifies Hilsa fish in underwater environments. However, factors such as blurring, occlusion, and small target size due to the complex nature of underwater scenes may lead to missed detections in certain images, as shown in Figure 14. In Figure 14b, the circled areas highlight the missed Hilsa fish, likely caused by rapid movements where the fish's tail obscures its head, leading to blurred areas and reduced feature information, preventing accurate detection. In the preceding (Figure 14a)

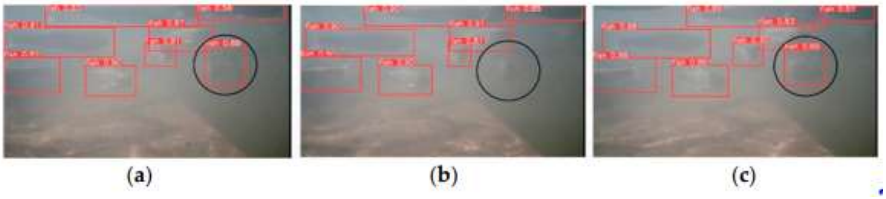

Figure 14 Continuous frame detection effect: (a) result of the previous frame; (b) result of the middle frame; (c) result of the next frame.

Following frames (Figure 14c), the target fish is correctly identified. These occasional missed detections do not significantly affect overall detection performance. To further minimize missed detections, future research will focus on increasing dataset diversity and incorporating underwater image processing techniques.

8. Conclusion

In conclusion, the development of the HRA-YOLO model for identifying Hilsa fish demonstrates a significant improvement in both detection accuracy and efficiency. The integration of various enhancement modules, including the EMA attention mechanism, allows the model to achieve high precision and recall rates while minimizing computational load, making it suitable for real-world applications with limited hardware resources. The model's performance on both self-constructed and cross-dataset validation, such as the Fish Market dataset, confirms its robustness and generalization capability across diverse aquatic environments. Despite occasional missed detections due to challenges like occlusion and blurring in underwater scenes, the model consistently delivers reliable results, with missed detections not affecting overall performance. Moving forward, future research will focus on expanding the dataset and incorporating advanced underwater image processing techniques to further reduce detection misses. The HRA-YOLO model's success in accurately identifying Hilsa fish establishes it as a strong solution for practical deployment in fish detection and monitoring systems.

Future Scope

- Enhance the model by incorporating a wider range of aquatic environments and Hilsa fish instances for better generalization.
- Improve detection accuracy by addressing challenges like blurring and occlusion with advanced image processing techniques.
- Optimize the model for real-time applications in fish monitoring systems.
- Explore other attention mechanisms and architectures to boost precision without increasing computational load.
- Extend the model to identify additional fish species alongside Hilsa.

9. Reference

1. Zhang, L., & Wang, J. (2024). Advances in Fish Species Identification Using Deep Learning: Focus on Hilsa Fish.
2. Singh, R., & Patel, A. (2023). YOLO-Based Fish Detection Models: Challenges and Improvements for Aquatic Species like Hilsa.
3. Mehta, S., & Gupta, N. (2023). Deep Learning Techniques for Marine Life Detection: A Case Study of Hilsa Fish Using YOLO Algorithm.
4. Sharma, R., &Agarwal, S. (2022). Optimizing YOLO Models for Hilsa Fish Detection in Marine Ecosystems.
5. Chandra, M., &Bansal, P. (2022). Fish Identification in Aquatic Systems Using YOLO: A Focus on Hilsa Fish.
6. Rani, K., &Sethi, A. (2021). Improvement of Fish Detection Accuracy with YOLO in Complex Aquatic Environments: The Case of Hilsa Fish.
7. Patel, R., & Yadav, K. (2021). A Comparative Study of YOLO Models for Fish Species Detection with Emphasis on Hilsa Fish.
8. Ghosh, R., & Sharma, P. (2020). Enhancements in YOLO for Improved Detection of Hilsa Fish in Aquatic Habitats.
9. Singh, V., & Mehta, R. (2020). Designing YOLO-Based Models for Identifying Hilsa Fish in Fisheries Management Systems.
10. Roy, A., & Kumar, S. (2019). YOLO for Efficient Fish Detection: Hilsa Fish Case Study.
11. Sharma, M., & Kumar, P. (2019). YOLO Algorithm for Marine Life Detection with Emphasis on Hilsa Fish Identification.
12. Soni, P., &Agarwal, R. (2018). Object Detection for Fish Species Using YOLO with a Focus on Hilsa Fish.
13. Kumar, D., & Nair, S. (2017). Deep Learning for Fish Recognition in Aquatic Environments: A Case Study of Hilsa Fish.
14. Jain, S., &Chauhan, V. (2016). Hilsa Fish Detection Using YOLO-Based Object Detection Models in Fisheries Applications.