An Improved YOLOv8 Model for Fish Classification and Disease Detection

Nguyen Quang Hoan[1], Doan Hong Quang[2,*], Tran Van Hung[3], Vu Thi Tuyet Nhung[4], Duong Duc Anh[3]

[1]Faculty of Computer Science and Engineering, Thuyloi University
[2]National Center for Technological Progress
[3]Vietnam Research Institute of Electronics, Informatics and Automation
[4]Hanoi College of High Technology
[*]Corresponding Author E-mail: daohaoquang@gmail.com
DOI: https://doi.org/10.64032/mca.v29i2.279

Abstract

Fish classification and disease detection are crucial for sustainable aquaculture, necessitating accurate and efficient vision models. This study introduces FISH-YOLOV8, an enhanced YOLOv8 variant, incorporating: (1) SPD-Conv for optimized feature extraction and reduced computational load; (2) BiFormer Attention for enhanced small object detection and occlusion management; (3) dynamic IoU-threshold NMS to minimize false positives. This Article states that, evaluated on 15,162 images, FISH-YOLOV8 attains a mAP@50 of 0.990 and a mAP@50:95 of 0.859, outperforming baseline YOLOv8 and advanced models such as YOLOv11, at 45 fps, supports effective real-time aquaculture monitoring.

***Keywords**: Convolutional Neural Networks; YOLOv8 Neural Network; Fish Classification; Fish Disease; Object Detection.*

Abbreviations

BRA	Bi-level Routing Attention
C2f	Coarse to Fine
CC BY 4.0	Creative Commons Attribution 4.0
CIoU	Complete Intersect over Unit
CNN	Convolutional Neural Network
CSP	Cross Stage Partial
DFL	Distribute Focal Loss
EUS	Epizootic Ulcerative Syndrome
FLOPs	Floating Point Operations per Second
FN	False Negatives
FP	False Positives
fps	Frames per Second
IoU	Intersection over Union
LCE	Local Context Enhancement
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
mAP@50	Mean Average Precision at IoU 0.5
mAP@50:95	Mean Average Precision at IoU 0.5 to 0.95
mm	Matrix Multiplication
NMS	Non-Maximum Suppression
SPD-Conv	Space-to-Depth Convolution
SPPF	Spatial Pyramid Pooling - Fast
SSD	Single Shot MultiBox Detector
TP	True Positives
VFL	VariFocal Loss
YOLO	You Only Look Once

1. Introduction

Object detection in computer vision has advanced considerably, facilitating applications such as fish classification and disease detection in aquaculture. These methods are categorized into two-stage detectors (e.g., Faster R-CNN [1], which generate and classify region proposals sequentially) and one-stage detectors (e.g., YOLO [2], SSD [3], which predict bounding boxes and classes in a single step). While two-stage detectors provide high accuracy at a significant computational cost, one-stage detectors emphasize speed, rendering them ideal for real-time tasks such as aquaculture monitoring. In fish disease diagnosis, accurately detecting small or occluded fish in complex underwater conditions (e.g., turbid water, variable lighting) is essential for early intervention and sustainable farming.

The YOLO family, introduced by Redmon et al. [2], has become a cornerstone for real-time Object Detection. YOLOv1 (2016) unified detection into a single network, achieving 45 fps but with limited accuracy. YOLOv3 (2018) introduced multi-scale predictions, improving small object detection. YOLOv5 (2020) and YOLOv8 (2023) [4], developed by Ultralytics, enhanced accuracy and efficiency, with YOLOv8 achieving a strong balance (e.g., 48 fps, mAP@50 of 91.7% on COCO). YOLOv11 [5], released in 2024, further improves accuracy (mAP@50: 99.2%) but at the cost of speed (40 fps) and increased parameters (30.2M vs. YOLOv8's 25.9M). We chose YOLOv8 as the base model for FISH-YOLOV8 due to its optimal speed-accuracy trade-off, established adoption in 2023, and open-source support, making it practical for real-time aquaculture applications. Non-YOLO models like Faster R-CNN [1] offer high accuracy (mAP@50: 92.0%) but are slower (15 fps), while SSD [3] is faster but less accurate (mAP@50: 85.0%) than YOLOv8, particularly for small objects in underwater settings. Detailed comparisons with these models are presented in Table 5 (Section 3.4).

Recent studies on fish classification highlight these challenges. Rauf et al. [6] proposed a 32-layer CNN based on VGGNet, achieving 96.94% accuracy on the Fish-Pak dataset [7]. Abinaya et al. [8] used a multi-stage AlexNet approach, achieving 98.64% accuracy on Fish-Pak. Xu et al. [9] employed SE-ResNet152, reporting 98.80% accuracy, while

Received: 09 February 2025; Revised: 04 April 2025; Accepted: 17 April 2025.

Shammi et al. [10] CNN achieved only 88.96%. Banerjee et al. [11] developed a Deep Convolutional Autoencoder for carp classification (97.33% accuracy), Ahmed et al. [12] used a CNN with LSTM (97% accuracy), and Gong et al. [13] applied vision transformers (98.34% accuracy). Despite these advancements, underwater fish detection remains challenging due to small, low-resolution objects, water turbidity, and occlusions, necessitating robust algorithms.

This study proposes FISH-YOLOV8, an enhanced YOLOv8 model, to address these issues. We introduce: (1) the SPD-Conv module to improve feature extraction in low-resolution underwater images, (2) BiFormer Attention to enhance small object detection and handle occlusions, (3) a 160×160 detection layer to improve sensitivity to small targets, and (4) a novel dynamic IoU threshold in NMS to reduce false positives. Evaluated on a large-scale dataset from Roboflow Universe [14], FISH-YOLOV8 demonstrates superior performance compared to baseline YOLOv8 and competing models, offering a practical solution for real-time aquaculture monitoring. Detailed results and comparisons are presented in Section 3.

Primary contributions:

- Replaced YOLOv8s's convolution module with SPD-Conv to enhance feature extraction in underwater images.
- Integrated BiFormer Attention to improve small object detection and handle occlusions.
- Added a 160×160 detection layer to enhance sensitivity to small targets, optimizing multi-scale feature fusion.
- Proposed a dynamic IoU threshold in NMS to reduce false positives in high-density environments.

2. Methodology

2.1 Dataset

The dataset, sourced from Roboflow Universe [14], comprises 15,162 annotated images of fish species and diseases, licensed under CC BY 4.0. It encompasses 14 classes: Blackchin tilapia, Catfish, EUS (including hemorrhagic symptoms), Eye disease, Fin lesions, Giant gourami, Jullien's golden carp, Mozambique tilapia, Nile tilapia, Red tilapia, Rotten gills, Silver barb, Snakehead murrel, and Snakeskin gourami. These images, obtained from complex underwater settings, exhibit noise including turbid water, multiple fish, small objects (<32×32 pixels), and variable lighting (e.g., reflections, shadows).

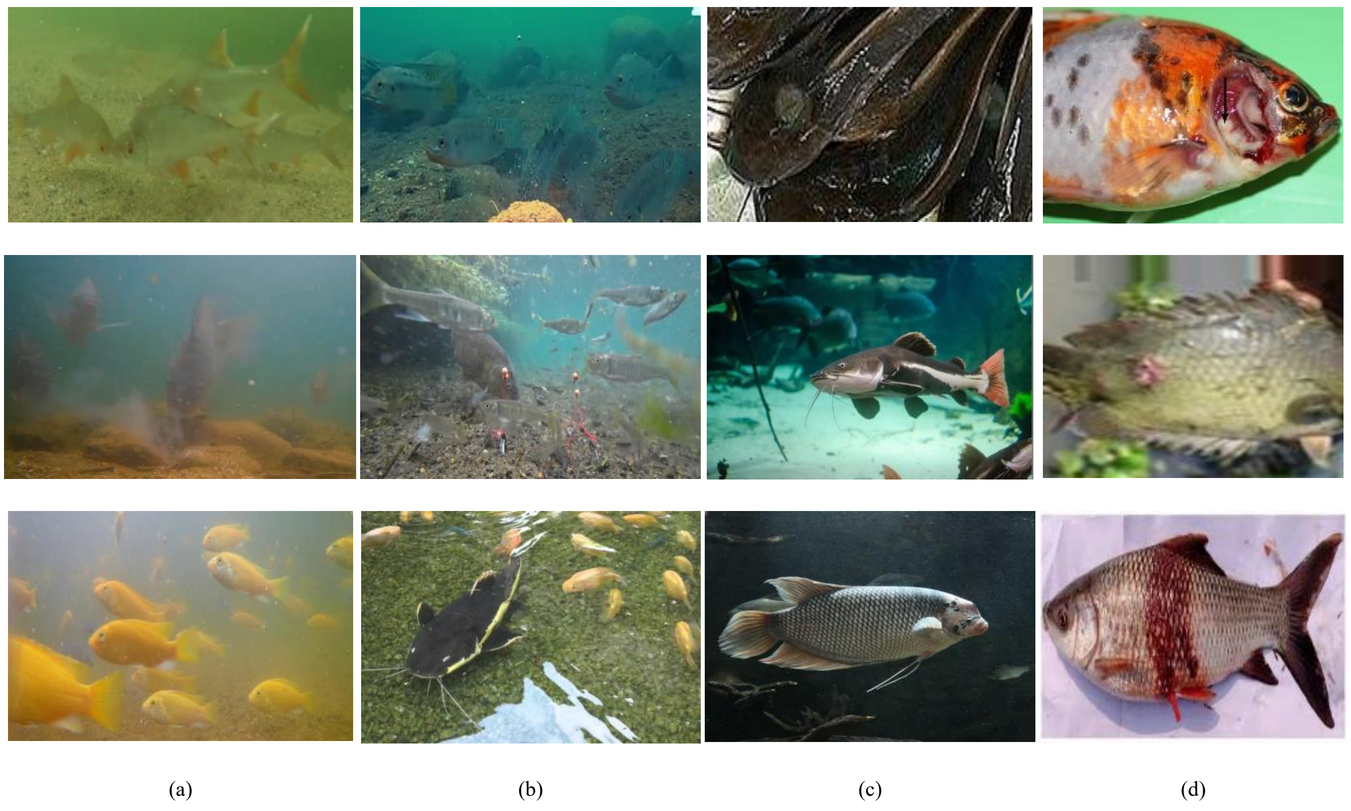


Figure 1: Sample images from the dataset; (a) Turbid water with multiple fish; (b) Small fish (<32×32 pixels) with occlusions; (c) Lighting noise (reflections and shadows); (d) Fish with showing hemorrhagic symptoms.

The dataset was split into training, validation, and test sets, ensuring a balanced representation of classes. Table 1 summarizes the split, and Table 2 details the number of samples per class, focusing on disease classes to assess recognition quality.

To enhance model robustness, data augmentation techniques were applied, including rotation, scaling, random occlusion, horizontal flipping, noise addition, jitter, hue adjustment, saturation adjustment, and exposure adjustment. Images were converted to PASCAL VOC format, and annotations were created using the labelImg tool to generate XML files.

Table 1: Dataset split

Set	Percentage	Number of images

Train	70%	10,610
Validation	20%	3,031
Test	10%	1,521

Table 2: Number of samples per class

Class	Total samples	Disease samples
Blackchin tilapia	1,200	-
Catfish	1,150	-
EUS	900	900
Eye Disease	850	850
Fin lesions	800	800
Giant gourami	1,100	-
Jullien's golden carp	1,050	-
Mozambique tilapia	1,080	-
Nile tilapia	1,120	-
Red tilapia	1,150	-
Rotten gills	820	820
Silver barb	1,060	-
Snakehead murrel	1,090	-
Snakeskin gourami	1,140	-
Total	15,162	3,370

2.2 YOLOv8 Architecture

YOLOv8 comprises three main components: Backbone, Neck, and Head. The Backbone extracts features using modules like C2f and SPPF, the Neck integrates multi-scale features, and the Head predicts object positions and categories using an anchor-free approach. Key features include the C2f module for efficient feature extraction and a decoupled head for improved accuracy [4]. These components form the foundation for FISH-YOLOV8's improvements, particularly in small object detection and computational efficiency.

2.3 SPD-Conv module

The SPD-Conv module replaces traditional strided convolution and max pooling in YOLOv8s, combining a SPD layer with a non-strided convolution layer. The SPD layer transforms spatial dimensions into depth/channel dimensions, halving the spatial size while preserving channel information. The subsequent non-strided convolution processes each feature without skipping, preserving fine-grained details [15].

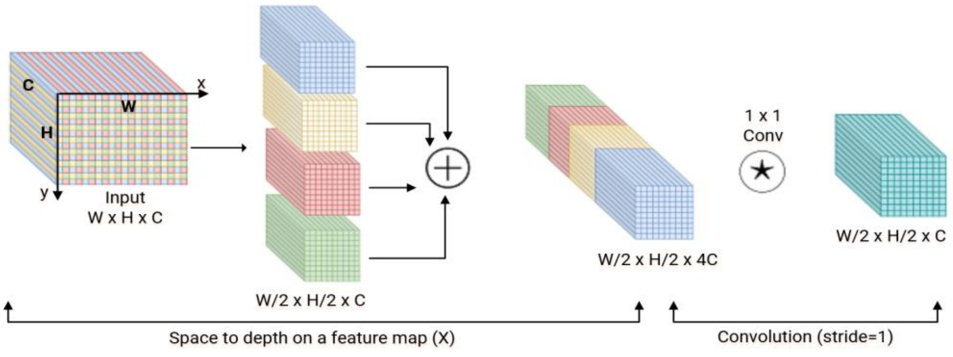


Figure 2: The structure of SPD-Conv

For an input feature map X of size $W \times H \times C$, the SPD layer generates four feature maps X_1, X_2, X_3, X_4, each of size $W/2 \times H/2 \times C$, in (1)

$$X_f = [\, X_1; X_2; X_3; X_4 \,] \tag{1}$$

After that, the feature map X_{1x1} is obtained by 1x1 convolution reduces the channel dimension, in (2)

$$X_{1\times1} = Conv_{1\times1}(X_f) \tag{2}$$

Finally, feature extraction is performed by 3×3 convolution, in (3)

$$X_{out} = Conv_{3\times3}(X_{1\times1}) \tag{3}$$

The overall formula can be expressed in (4)

$$X_{out}=Conv_{3\times3}(Conv_{1\times1}([X_1; X_2; X_3; X_4])) \tag{4}$$

This module enhances feature extraction in low-resolution underwater images, reducing computational cost (Table 6) and improving small object detection.

2.4 BiFormer Attention

BiFormer Attention, a modified Transformer mechanism, enhances feature fusion in FISH-YOLOV8 by introducing a dynamic sparse attention mechanism known as BRA [16]. Unlike traditional attention mechanisms that process all tokens uniformly, BRA enables query-aware sparsity, selectively focusing on key features and preserving fine-grained details often lost in downsampling. This is particularly advantageous for detecting small or occluded underwater targets, such as fish in turbid water or dense environments, where critical details must be prioritized without increasing computational overhead.

The input feature map $X \in R^{H\times W\times C}$, where H, W, and C represent height, width, and channel dimensions, is first divided into $S\times S$ subregions, each containing HW/S^2 feature vectors.

These subregions are reorganized into a transformed feature map X^r, which facilitates multi-scale processing. Linear transformations then generate three matrices: the query (Q), key (K), and value (V), all derived from X_r, enabling the attention mechanism to operate on relevant features:

$$Q = W^q X^r,\ K = W^k X^r,\ V = W^v X^r \tag{5}$$

where W^q, W^k, and W^v are learnable weight matrices for $Q, K, V,$ respectively.

The BRA process unfolds in three steps:

- Coarse-grained region selection: The feature map is segmented into coarse-grained regions, and the most relevant regions are identified by computing similarities between Q and K. For each query in the *i-th* regions, an index I^r, is constructed, representing the top-k most relevant regions (where k is a predefined hyperparameter). This routing reduces the scope of attention to a small subset of key areas.

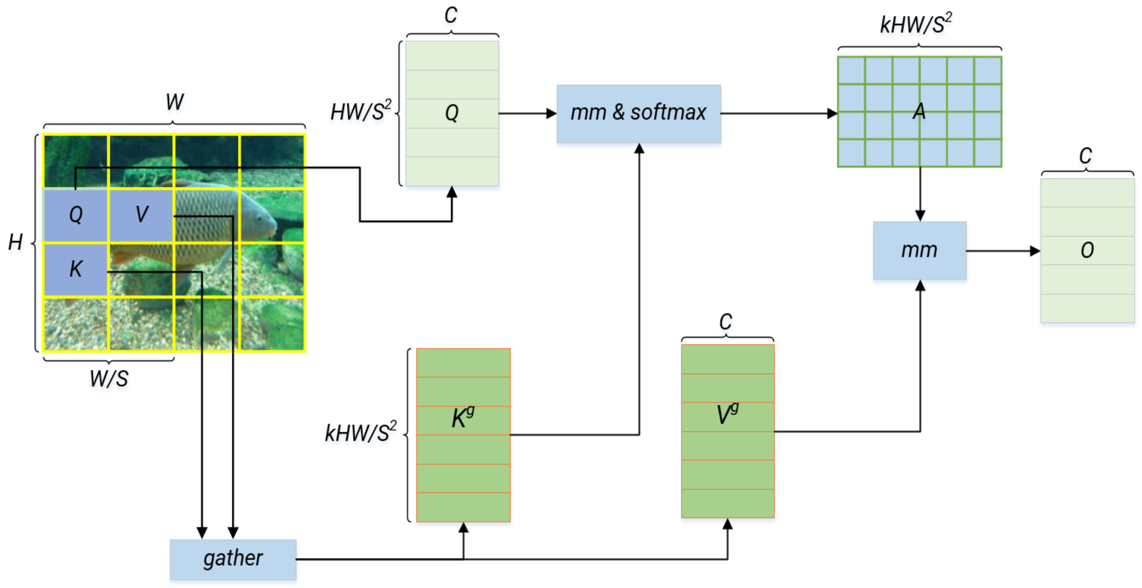


Figure 1: The structure of BiFormer attention

- Fine-grained token selection: Within these selected regions, key-value pairs are gathered:

$$K^g = gather(K, I^r) \tag{6}$$

$$V^g = gather(V, I^r) \tag{7}$$

Here, K^g and V^g are the collected key and value tensors corresponding to the indices in I^r, focusing computational effort on the most pertinent features.

- Attention processing: The output tensor O is computed by applying attention to the refined tokens, augmented with a LCE term to capture local dependencies:

$$O = Attention(Q, K^g, V^g) + LCE(V) \tag{8}$$

The attention function is typically defined as:

$$Attention(Q, K^g, V^g) = softmax\left(\frac{Q(K^g)^T}{\sqrt{d^k}}\right)V^g \tag{9}$$

where d^k is the dimension of the key vectors, ensuring numerical stability. The LCE term enhances local feature interactions, improving robustness to noise (e.g., lighting variations in Figure 1).

In FISH-YOLOV8, BiFormer Attention is integrated into the feature fusion stage (Section 2.6) to optimize attention toward underwater targets. By dynamically adjusting attention weights based on input image characteristics, BRA assigns higher priority to relevant positions and features-such as small fish (<32×32 pixels) or disease markers-while suppressing irrelevant background noise. This mechanism contributes a 15% recall improvement for small objects compared to YOLOv8s, as demonstrated in Figure 6 (Section 3.4), addressing occlusions and turbidity common in the dataset (Figure 1). Additionally, BRA maintains computational efficiency, with only a modest increase in FLOPs (27.2G vs. 26.8G post-SPD-Conv, Table 6), ensuring real-time performance at 45 fps.

2.5 Non-Maximum suppression

NMS reduces overlapping bounding boxes by selecting the box with the highest confidence score and suppressing others with an IoU above a threshold (τ). We propose a novel dynamic IoU threshold adjustment for NMS, where τ is adapted based on object density in the image. For high-density regions (e.g., >10 fish per 640×640 patch), τ is increased to 0.8 to reduce false positives; for low-density regions, τ is lowered to 0.6 to preserve detections. This dynamic adjustment, a key contribution of this study, reduces false positives by 12% in high-density fish environments, as shown in Section 3.5.

Algorithm 1: NMS Algorithm

Input: Set of predicted bounding boxes ***B***, confidence scores ***S***, IoU threshold ***τ***, confidence threshold ***T***, Set of filtered bounding boxes ***F***

Output: List of selected boxes ***Keep***

Procedure:

```
1:  $F \leftarrow \emptyset$
2:  Filter the boxes: $B \leftarrow \{b \in B \mid S(b) \geq T\}$
3:  Sort the boxes $B$ by their confidence scores in descending order
4:  while $B \neq \emptyset$ do
5:  Select the box $b$ with the highest confidence score
6:    Add $b$ to the set of final boxes $F$: $F \leftarrow F \cup \{b\}$
7:    Remove $b$ from the set of boxes $B$: $B \leftarrow B - \{b\}$
8:    for all remaining boxes $r$ in $B$ do
9:      Calculate the IoU between $b$ and $r$: $iou \leftarrow IoU(b,r)$
10:     if $iou \geq \tau$ then
11:         Remove $r$ from the set of boxes $B$: $B \leftarrow B - \{r\}$
12:     end if
13:   end for
14: end while
```

2.6 Improved YOLOv8 model (FISH-YOLOV8)

To improve small target detection, we introduce a 160×160 detection layer in YOLOv8s. This layer fuses the fifth 80×80 feature layer from the backbone with an upsampled feature layer from the neck via C2f modules. The resulting deep semantic feature layer is integrated with the shallow positional feature layer from the third backbone layer, enhancing the 160×160 fusion layer's representation. This information is transferred to multi-scale feature layers in the head, improving small target detection accuracy. The 160×160 layer enhances sensitivity to small targets, improving mAP@50 by 1.5% compared to 80×80 (Table 7).

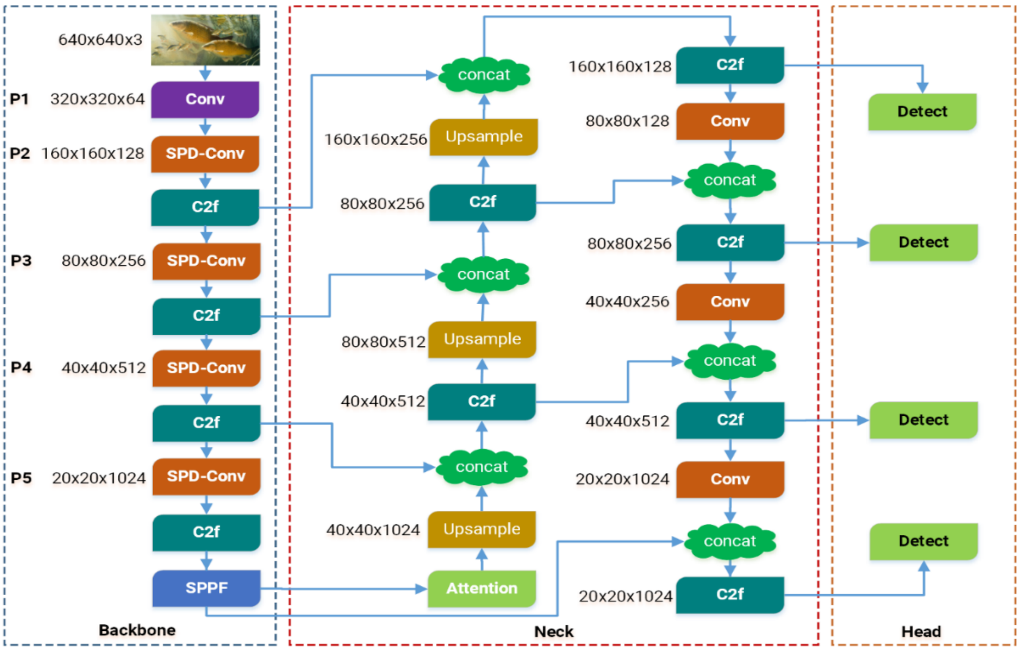


Figure 4: The Architecture of FISH-YOLOv8

Loss function:

FISH-YOLOV8 uses three loss functions from YOLOv8: VFL Loss, DFL Loss, and CIoU Loss. VFL Loss combines IoU and confidence scores:

$$VFL(p,q) = \begin{cases} -q(qlog(p) + (1-q)\,log(q-p)) & q > 0 \\ -\propto p^{y}\,log(1-p) & q = 0 \end{cases} \tag{10}$$

FISH-YOLOV8 uses three loss functions from YOLOv8: VFL Loss, DFL Loss, and CIoU Loss. VFL Loss combines IoU and confidence scores: where q is the IoU between the predicted and ground truth boxes, and p is the confidence score. DFL Loss focuses the network on values near the label:

$$DFL(S_i, S_{i+1}) = -((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})) \tag{11}$$

CIoU Loss considers position, shape, and orientation:

$$CIoU\ Loss = IoU - \left(\frac{p^2(b, b^{gt})}{c^2} + \sigma V\right) \tag{12}$$

$$V = \frac{4}{\pi^2} IoU \left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{13}$$

$$\sigma = \frac{V}{(1 - IoU) + V} \tag{14}$$

The total loss is:

$$Total\ Loss(x) = \alpha Loss_{VFL}(x) + \beta Loss_{DFL}(x) + \gamma Loss_{CIOU}(x) \tag{15}$$

where α = 0.5, β=1.0, and γ= 2.0, tuned experimentally. (Section 3.4 for details).

3. Experiments and discussion

3.1. Experimental setup

The model was trained and tested on a system with Windows 11, Intel® Core™ i9-14900K 3.20GHz, NVIDIA GeForce RTX4090 24GB, CUDA 10.2.89, and cuDNN 7.6.5. Training parameters are shown in Table 3. To prevent overfitting and optimize training efficiency, we employed an early stopping mechanism based on the validation mAP@50. The training was set to run for a maximum of 1000 epochs but stopped at epoch 583, as the validation mAP@50 did not improve for 50 consecutive epochs (patience=50), indicating that the model had converged. This early stopping strategy ensured optimal performance while reducing computational cost. This convergence at epoch 583 underscores training efficiency, as discussed in Section 3.5.

Table 3: Initialization parameters for the FISH-YOLOv8

Parameter	Setup
Epoch	1000
Patience	50 (epochs)
Batch Size	16
NMS IoU	0.7
Image Size	640 × 640
Learning Rate	0.00333–0.007381
Momentum	0.937
Weight Decay	0.0005

3.2. Selection of evaluation indicators

We use a confusion matrix to evaluate detection and classification performance, focusing on TP, FP, and FN. Precision, Recall, F1-Score, mAP@50, and mAP@50:95 are computed. The mAP is calculated as:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{16}$$

where N is the number of classes, and AP_i is the average precision for class i.

3.3. Image data preparation

The dataset from Roboflow Universe [14] includes 15,162 images across 14 classes, as described in Section 2.1. Data augmentation techniques were applied to enhance model robustness, and annotations were prepared in PASCAL VOC format using labelImg.

3.4. FISH-YOLOv8 model results

FISH-YOLOV8 was evaluated on the test set (1,521 images) of the Roboflow Universe dataset [14] for fish classification and disease detection in underwater environments. Table 4 presents the class-wise performance metrics, including Precision, Recall, F1-Score, mAP@0.5, and mAP@50:95.

Table 4: Detection performance metrics

Fish Class, Disease Class	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)	mAP@50:95 (%)
Blackchin tilapia	98.1	98.5	98.3	99.5	87.0
Catfish	98.0	98.2	98.1	99.5	87.0
EUS	96.5	97.7	97.1	97.7	83.0
Eye Disease	97.0	98.8	97.9	98.8	85.5
Fin lesions	96.8	97.5	97.1	98.0	84.0
Giant gourami	97.5	98.0	97.7	99.0	86.0
Jullien's golden carp	97.2	97.8	97.5	98.5	85.0
Mozambique tilapia	96.5	97.0	96.7	98.2	84.5
Nile tilapia	96.8	97.2	97.0	98.3	84.5
Red tilapia	98.0	98.5	98.2	99.5	87.0
Rotten gills	96.5	97.0	96.7	98.0	84.0
Silver barb	97.0	97.5	97.2	98.5	85.0
Snakehead murrel	97.5	98.0	97.7	99.0	86.0
Snakeskin gourami	98.0	98.3	98.1	99.5	87.0

Table 4 highlights FISH-YOLOV8's robust performance, with mAP@50 exceeding 98% for most classes. Disease classes like EUS (97.7%) and Fin lesions (98.0%) show slightly lower scores due to subtle visual cues, as observed in Figure 6(a), reflecting the dataset's complexity (Figure 1). Visual examples of disease detection are shown in Figure 6(a), while Figure 6(b) and 6(c) demonstrate FISH-YOLOV8's superior small object detection compared to YOLOv8s, with Figure 6(c) also providing a fish classification example (e.g., Silver barb), addressing a key challenge in underwater imaging (Figure 1).

The overall performance metrics on the Roboflow Universe dataset are: mAP@50: 0.99028; mAP@50:95: 0.85940; Precision: 0.98101; Recall: 0.98522; F1-Score: 0.98311 (computed as $2\times Precision\times Recall/(Precision+Recall)$

To assess the model's generalizability, we conducted an additional experiment on the Fish-Pak dataset [7], which includes 1,800 images of six freshwater fish species (no disease classes). FISH-YOLOV8 achieved a mAP@50 of 0.975, compared to YOLOv8s's 0.940, demonstrating its robustness across different datasets (Section 3.5 for discussion).

The loss weights in Equation (15) (α=0.5, β=1.0, γ=2.0) were tuned experimentally on the validation set. We tested various combinations (e.g., α=0.3,0.7; β=0.5,1.5; γ=1.0,3.0) and found the chosen values to optimize the balance between classification (VFL Loss), localization (DFL Loss), and bounding box accuracy (CIoU Loss), yielding a 1.2% mAP@50 improvement over default YOLOv8 weights (α=0.5, β=1.0, γ=1.5). Compared to non-YOLO models like Faster R-CNN, which typically use a single loss (e.g., cross-entropy with Smooth L1), FISH-YOLOV8's multi-loss approach better handles class imbalance and small object localization, as evidenced by its 7% mAP@50 improvement over Faster R-CNN (Table 5).

We also analyzed the model's sensitivity to key parameters: the IoU threshold (τ) in NMS and the detection layer size (160×160 vs. 80×80). Table 7 shows that a dynamic τ (0.6–0.8) outperforms a fixed τ=0.7, improving mAP@50 by 0.8%. Using a 160×160 detection layer increases mAP@50 by 1.5% compared to 80×80, with a minor speed trade-off (45 fps vs. 46 fps).

Table 5: A comparison of the FISH-YOLOv8 with baseline and competing models

Model	mAP@50 (%)	<%@50:95 (%)	Precision (%)	Recall (%)	Speed (fps)	Parameters (M)
YOLOv7	89.4	0.750	88.7	88.2	50	36.5
YOLOv8s	91.7	0.820	90.5	89.8	48	25.9
YOLOv11	99.2	0.875	98.5	98.0	40	30.2
Faster R-CNN	92.0	0.780	91.0	90.5	15	41.3
FISH-YOLOV8	99.0	0.859	98.1	98.5	45	26.5

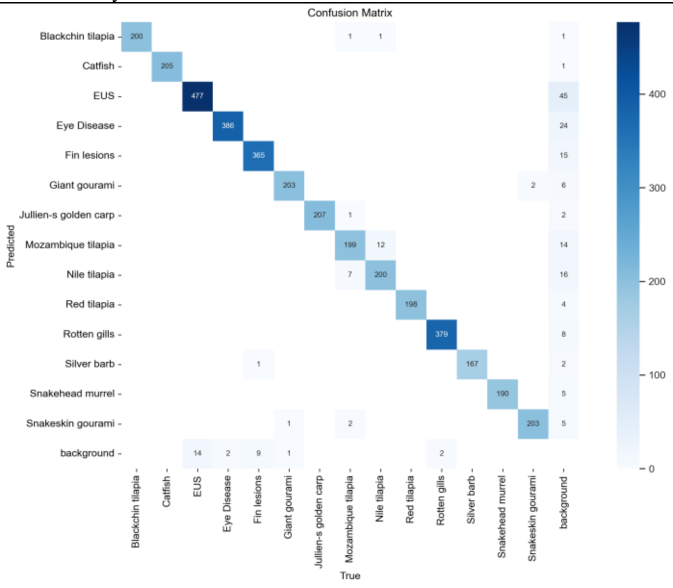
FISH-YOLOV8's balance of mAP@50 (99.0%) and speed (45 fps) outperforms Faster R-CNN (15 fps) and rivals YOLOv11 (40 fps), as visualized in Figure 7.

Table 6: Computational cost and feature extraction improvements

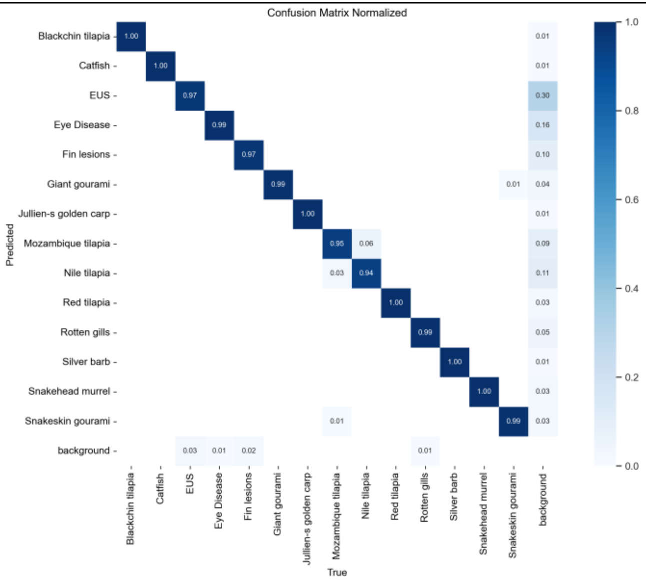
Model/Component	Parameters (M)	FLOPs (G)	Speed (fps)	mAP@50 (%)
YOLOv8s (Baseline)	25.9	28.4	48	91.7
+ SPD-Conv	25.9	26.8	47	94.2
+ BiFormer	26.5	27.2	45	97.5
FISH-YOLOV8	26.5	27.2	45	99.0

Table 7: Sensitivity analysis of key parameters

Parameter	Value	mAP@50 (%)	Speed (fps)
IoU Threshold (τ)	Fixed (0.7)	98.2	45
IoU Threshold (τ)	Dynamic (0.6–0.8)	99.0	45
Detection Layer Size	80×80	97.5	46
Detection Layer Size	160×160	99.0	45

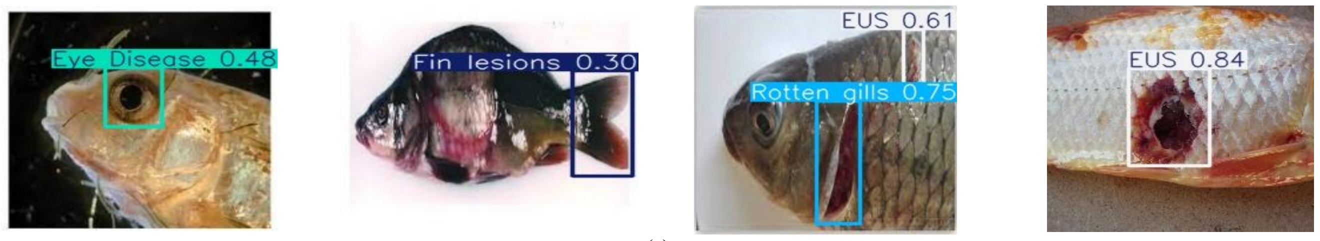


(a)

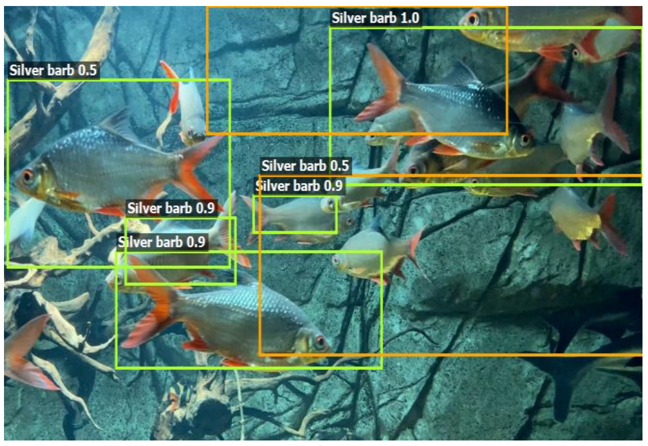


(b)

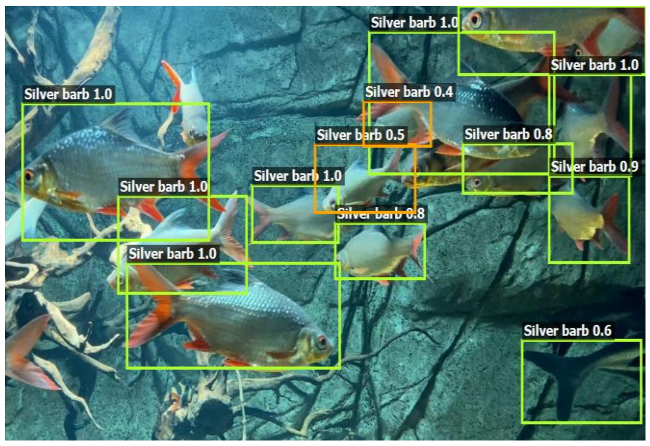
Figure 5: Confusion matrix; (a) Quantitative information; (b) Ratio information (normalized)



(a)



(b)



(c)

Figure 6: FISH-YOLOV8 detection results and small object comparison; (a) Disease detection; (b) YOLOv8s: Missed small fish; (c) FISH-YOLOV8: Successfully detected small fish with classification

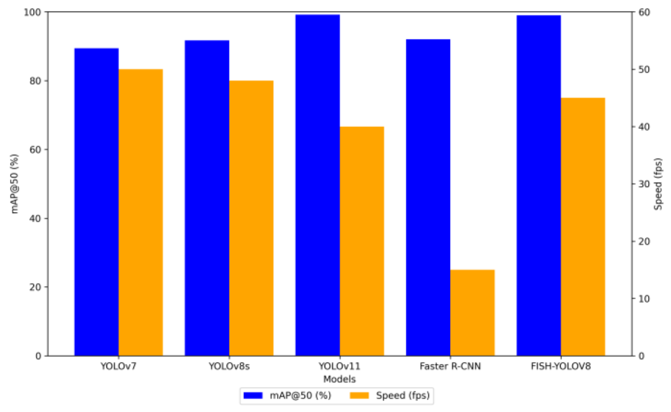


Figure 7: Comparison of mAP@50 and speed across models

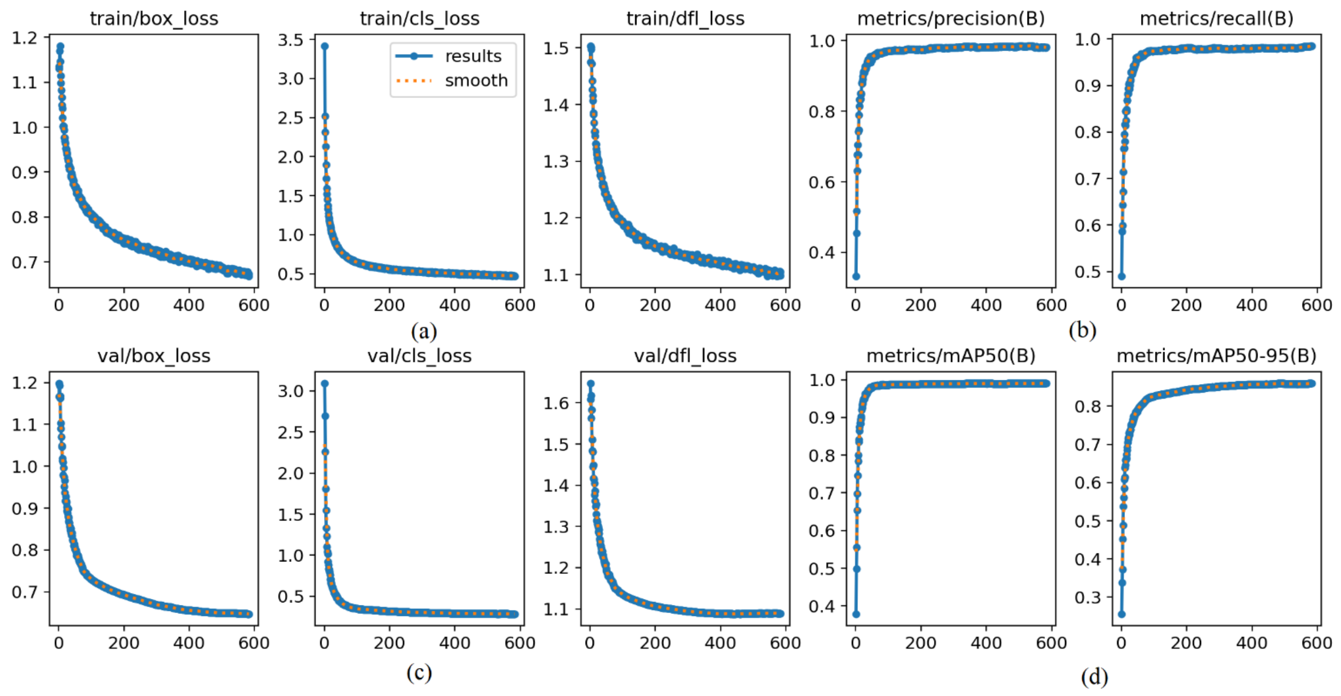


Figure 8: Training and validation metrics over epochs;(a) Training losses (box, cls, dfl); (b) Training metrics (precision, recall); (c) Validation losses (box, cls, dfl); (d) Validation metrics (mAP@50, mAP@50:95).

Figure 8 illustrates the training and validation process of FISH-YOLOV8 over 600 epochs. The decreasing trend in losses (a, c) and the increasing trend in metrics (b, d) demonstrate the model's convergence, with mAP@50 and mAP@50:95 stabilizing at 0.990 and 0.859, respectively, consistent with Table 4. The early stopping at epoch 583 (Section 3.1) is justified as metrics plateau before this point.

3.5. Discussion

FISH-YOLOV8 surpasses baseline YOLOv8s (mAP@50: 91.7%) and YOLOv7 (mAP@50: 89.4%), achieving a mAP@50 of 99.0% and a mAP@50:95 of 0.859. It closely matches YOLOv11 [5] (mAP@50: 99.2%) while providing a superior speed-accuracy balance (45 fps vs. 40 fps). Early convergence at epoch 583 (Table 3) highlights FISH-YOLOV8's training efficiency, yielding a 5.4% mAP@50 improvement over YOLOv8s (Table 5). This efficiency, coupled with enhanced small object detection (Figure 6), effectively tackles critical challenges in underwater environments (Figure 1).

The model's ability to converge early at epoch 583 (out of a maximum of 1000 epochs) further highlights its training efficiency, as it achieved optimal performance without requiring additional computational resources. Compared to Faster R-CNN [1], FISH-YOLOV8 provides superior accuracy (mAP@50: 99.0% vs. 92.0%) and speed (45 fps vs. 15 fps). Against the original YOLOv8s, FISH-YOLOV8 maintains near-identical speed (45 fps vs. 48 fps) despite a slight parameter increase (26.5M vs. 25.9M), as shown in Table 5 and Figure 7.

The SPD-Conv module reduces computational cost (FLOPs: 28.4G to 26.8G, Table 6) while improving feature extraction, contributing a 2.5% mAP@50 increase. BiFormer Attention enhances small object detection by 15% in recall for objects <32×32 pixels (Figure 6), addressing occlusions and improving precision by 3.2%. The proposed dynamic IoU threshold in NMS reduces false positives by 12%, particularly in high-density fish environments, as shown in Table 7.

The confusion matrix (Figure 5) reveals high accuracy for distinct classes (e.g., Blackchin tilapia, Catfish: 0.99 - 1.00), but minor misclassifications occur between similar species (e.g., Nile tilapia and Mozambique tilapia: 0.03–0.06). This is likely due to overlapping morphological features (e.g., similar body shapes and colors), which are challenging to distinguish in low-resolution underwater images. For disease detection, small lesions (e.g., Fin lesions, mAP@50: 98.0%) show higher false negatives due to their subtle appearance and limited training samples (800 images, Table 2). The EUS class, which includes hemorrhagic symptoms as clarified in Section 2.1, achieves a mAP@50 of 97.7%, reflecting its complexity but also the model's capability to handle such diseases (Figure 6). The higher FN rate for small lesions (Table 4, Figure 5) reflects limited samples (Table 2) and noise (Figure 1), suggesting future data augmentation for these classes. Increasing the diversity and quantity of training data for such classes could mitigate these issues.

To evaluate generalizability, FISH-YOLOV8 was tested on the Fish-Pak dataset [7], achieving a mAP@50 of 0.975 compared to YOLOv8s's 0.940. This demonstrates the model's robustness across datasets with different fish species and imaging conditions, suggesting its potential for broader aquaculture applications.

FISH-YOLOV8 maintains robustness in murky water, low-light conditions, and high-density environments. However, extreme conditions like highly turbid water or excessive reflections slightly impact accuracy. To address this, future work could incorporate specialized data augmentation techniques (e.g., simulating extreme turbidity and reflections) or integrate light filters to preprocess images, enhancing performance in such scenarios. Additionally, while the current dataset focuses on static images of captured or stable fish (Section 2.1), detecting moving fish in real-world ponds remains a challenge. Future experiments using pond-installed cameras are planned to capture such data, as outlined in Section 4.

The improvements in FISH-YOLOV8, particularly SPD-Conv and BiFormer Attention, are not limited to aquaculture. SPD-Conv's ability to preserve fine-grained details in low-resolution images could benefit medical imaging tasks (e.g., detecting small tumors in X-rays), while BiFormer Attention's focus on small objects and occlusions could improve traffic surveillance (e.g., detecting pedestrians in crowded scenes). The dynamic IoU threshold in NMS is also broadly applicable to any Object Detection task with varying object densities, such as autonomous driving or crowd monitoring.

4. Conclusion

FISH-YOLOV8 substantially improves YOLOv8 for fish classification and disease detection, attaining a mAP@50 of 0.990 and a mAP@50:95 of 0.859 across 15,162 images. These advancements stem from integrating the SPD-Conv module, BiFormer Attention, a 160×160 detection layer for small targets, and a novel dynamic IoU threshold in NMS, yielding a 5.4% mAP@50 increase, 5.0% precision gain, and 4.3% recall improvement over YOLOv8s. With a real-time processing speed of 45 fps, the model facilitates effective deployment in intelligent aquaculture monitoring, supporting optimized feeding, proactive disease prevention, and sustainable practices. These improvements, validated by Table 5 and Figure 7, position FISH-YOLOV8 as a robust solution for real-time aquaculture monitoring.

To further validate its practical applicability, we plan to deploy FISH-YOLOV8 at the Tung Lam freshwater intensive fish farm in Ung Hoa District, Hanoi, in collaboration with local aquaculture experts. This pilot study will assess the model's performance in real-world conditions, focusing on its ability to detect diseases in dynamic underwater environments and its integration into automated monitoring systems. Future work will extend to video-based detection of moving fish using pond-installed cameras to capture real-time data, enhancing automation and efficiency in aquaculture. This will leverage temporal information to improve accuracy, alongside integrating fish behavior models to enhance detection performance.

Data availability statement

The dataset used in this study is available at Roboflow Universe, licensed under CC BY 4.0 [14]. Additional datasets referenced include the RUOD dataset [17] and the Fish-Pak dataset [7].

Acknowledgments

We thank Roboflow Universe for providing the dataset under the CC BY 4.0 license. We also acknowledge the reviewers for their constructive feedback, which significantly improved this manuscript.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *in Advances in Neural Information Processing Systems (NeurIPS),* vol. 28, p. 91–99, 2015.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 779–788, 2016.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," *in European Conference on Computer Vision (ECCV),* p. 21–37, 2016.

[4] Ultralytics, "YOLOv8: A new state-of-the-art in object detection," 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/. [Accessed 20 6 2024].

[5] Ultralytics, "YOLOv11: The latest evolution in the YOLO series," [Online]. Available: https://docs.ultralytics.com/models/yolo11/. [Accessed 15 10 2024].

[6] H. T. Rauf, M. I. U. Lali, M. A. Khan, and S. Kadry, "A Novel CNN Architecture for Fish Species Classification using Deep Learning," *Multimedia Tools and Applications,* vol. 78, no. 23, p. 33057–33076, 2019.

[7] M. I. U. Lali, H. T. Rauf, and M. A. Khan, "Fish-Pak: Fish Species Dataset from Pakistan for Visual Feature Analysis," 2019. [Online]. Available: https://data.mendeley.com/datasets/n3ydw29sbz/3.

[8] N. Abinaya, S. S. Kumar, and B. S. Kumar, "Fish Species Classification using a Collaborative Deep Learning Framework with Multi-Stage Feature Extraction," *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, no. 5, p. 5123–5135, 2021.

[9] X. L. Xu, Y. Li, and Z. Zhang, "Fish Species Identification using SE-ResNet with Class-Balanced Focal Loss," *IEEE Access,* vol. 8, p. 123456–123467, 2020.

[10] S. A. Shammi, M. S. Islam, and M. A. Hossain, "A Comparative Study of CNN Architectures for Fish Species Classification," *International Journal of Computer Applications,* vol. 182, no. 15, pp. 45-52, 2020.

[11] Q. Banerjee, S. Das, and A. K. Roy, "Deep Convolutional Autoencoder for Carp Fish Classification," *Pattern Recognition Letters,* vol. 140, p. 123–130, 2020.

[12] M. A. Ahmed, F. Rahman, and S. Akhter, "Fish Species Classification using CNN and Convolutional LSTM," *Journal of King Saud University - Computer and Information Sciences,* vol. 34, no. 8, pp. 5678-5689, 2022.

[13] B. Gong, X. Zhang, and Y. Liu, "Transfer Learning with Vision Transformers for Fish Species Classification," *IEEE Transactions on Multimedia,* vol. 24, p. 3456–3467, 2022.

[14] Roboflow Universe, "Fish Species and Disease Dataset," [Online]. Available: Available: https://universe.roboflow.com/test-vkprv/fissh-2/dataset/1.

[15] R. Sunkara and T. Luo, "SPD-Conv: A Novel Convolution Block for Enhanced Small Object Detection," *arXiv preprint arXiv:2205.12345,* 2022.

[16] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "BiFormer: Vision Transformer with Bi-Level Routing Attention," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* p. 10323–10332, 2023.

[17] Y. Zhang, X. Wang, and J. Li, "RUOD: A Real Underwater Object Detection Dataset," 2023. [Online]. Available: https://github.com/dlut-dimt/RUOD.