TST-YOLO: TOKEN-SELECTIVE TWIN-ENHANCED YOLO FOR ROBUST UNDERWATER FISH DETECTION AND PHENOTYPE ESTIMATION ON AUVS

SIRIGINEEDI MANIKANTA[1], Dr. RAGHVENDRA KUMAR[2], Dr. R N V JAGAN MOHAN[3]

Research Scholar,Department of Computer Science and Engineering,GIET University, Gunupur-765022, Odisha.
Associate Professor,Department of Computer Science and Engineering,GIET University,Gunupur-765022, India
Associate Professor,Department of Computer Science and Engineering,SRKR Engineering College, Bhimavaram-534201, Andhra Pradesh, India

ABSTRACT

Underwater fish detection remains difficult due to turbidity, color cast, motion blur, and the prevalence of small, fast, look-alike targets on embedded AUV hardware. We present TST-YOLO, a compact detector that combines four synergistic components: (i) a physics-aware digital-twin synthesizer that exposes the model to realistic water-optics shifts during training; (ii) a lightweight pre-enhancement fusion stage that mitigates color/contrast bias before inference; (iii) a token-selective transformer head that prunes and merges low-information tokens for AUV-grade efficiency; and (iv) a phenotype-guided auxiliary head that estimates lateral-line scale counts to regularize features toward biologically meaningful structure. Evaluated under identical training budgets on DeepFish and DePondFi’23, TST-YOLO improves mAP@50–95 by +6.1 and +6.0 points, respectively, over strong YOLOv8/YOLOv7 baselines, with +5.5 APS_SS gains on small targets. Confidence calibration also improves (ECE ↓34%, Brier ↓), while the token-selective head reduces transformer tokens by ≈30% at equal or better accuracy, cutting end-to-end Jetson latency by ~6%. Results are reported as 5× runs (mean±SD) with bootstrap confidence intervals, paired tests, and McNemar analyses. Beyond accuracy gains, the contribution is positioned at the systems level, demonstrating how domain-aware data synthesis, reliability-oriented evaluation, and token-efficient transformer design can be jointly integrated for trustworthy, resource-constrained intelligent perception. This emphasis on efficiency, robustness, and statistical rigor highlights the relevance of TST-YOLO as an information-technology solution for dependable autonomous sensing rather than a task-specific detector alone.

KEYWORDS: *Underwater Object Detection; Autonomous Underwater Vehicles (Auvs); Digital-Twin Augmentation; Token-Selective Transformer; Underwater Image Enhancement; Small-Object Detection; Phenotype (Lateral-Line) Estimation.*

1. INTRODUCTION

Underwater visual sensing is intrinsically challenging. Light absorption and wavelength-dependent scattering introduce haze, color casts, and low contrast; suspended particulates and caustics distort edges; and platform motion often induces blur. In this setting, fish are typically small, fast, and visually similar, so detectors must preserve fine spatial detail while remaining stable to appearance shifts. Beyond accuracy, embedded constraints on autonomous underwater vehicles (AUVs) impose tight limits on latency, memory, and energy, forcing practical trade-offs between robustness and throughput. Surveys of aquaculture computer vision consistently report these intertwined difficulties—domain shift, small-object sensitivity, and deployment efficiency—as the principal barriers to reliable, long-term monitoring at scale [8].

Recent progress with one-stage detectors has raised performance on underwater imagery. Task-specific upgrades to YOLO families—alterable kernels, attention mechanisms, and streamlined necks/losses—show tangible gains on small, fast targets in noisy scenes [1]. Complementary work for AUV applications emphasizes resilience under degraded imaging (low illumination, blur, dense schools) and situational dynamics (migration, aggregation), but still reveals accuracy drops whenever the visual domain shifts from training conditions, and shows the persistent cost of running modern attention heads on edge hardware [3]. Community datasets and challenges have helped quantify these gaps, demonstrating that models trained on clear, well-lit footage frequently underperform in turbid water, crowded scenes, or unusual color spectra—precisely the regimes where ecological monitoring is most valuable [12]. These findings highlight that performance advances alone are insufficient unless accompanied by system-level considerations of reliability, efficiency, and operational stability under uncertainty.

This paper addresses the three constraints jointly with a detector designed for robustness, small-object fidelity, and embedded efficiency. We propose TST-YOLO, a compact architecture that integrates four mutually reinforcing components. First, a physics-aware digital-twin synthesizer augments training with samples that emulate underwater optics (attenuation, backscatter, illumination spectra), promoting feature invariance to turbidity and colour shift. Second, a lightweight pre-enhancement fusion stage reduces colour/contrast bias before inference, stabilizing low-level cues without expensive per-frame restoration. Third, a token-selective transformer head prunes and merges low-information tokens, retaining salient context for detection while lowering compute to meet AUV budgets. Fourth, a phenotype-guided auxiliary head estimates lateral-line scale counts, nudging intermediate features toward biologically meaningful structure that helps disambiguate visually similar species. **Together, these design choices frame the detector as a resource-aware perception system rather than a purely accuracy-driven model.**

From an information-technology perspective, the proposed framework targets practical deployment by explicitly balancing robustness, computational efficiency, and decision reliability, which are critical for autonomous sensing systems operating continuously on constrained edge hardware.

1.1 Novel Contributions:

1. We introduce TST-YOLO, an end-to-end pipeline that couples digital-twin augmentation and pre-enhancement with a token-selective transformer head for underwater fish detection on AUVs, directly targeting domain shift and efficiency [8,1,3].

2. We add a phenotype-guided auxiliary task (lateral-line scale estimation) that regularizes features toward species-relevant structure, improving small-object discrimination in dense schools.

3. We provide a deployment-oriented evaluation on two representative benchmarks with identical training budgets, reporting accuracy, calibration, and embedded latency, and showing consistent improvements where prior systems degrade (turbidity, low light, occlusion) [12]. This evaluation strategy emphasizes reproducibility and operational relevance rather than isolated benchmark gains.

1.2 Organization of the paper: Section 2 reviews related work on underwater detection, enhancement, efficient attention for edge platforms, and aquaculture-specific tasks, positioning our design against prior art [8,1,3,12]. Section 3 details the TST-YOLO architecture—digital-twin synthesis, pre-enhancement, token-selective head, and phenotype supervision—together with the learning objectives. Section 4 describes datasets, splits, and a fair, reproducible protocol tuned for AUV deployment. Section 5 defines metrics (including calibration and efficiency) and statistical tests. Section 6 reports comprehensive results—main comparisons, robustness strata, token/latency analyses, and ablations—followed by discussion in Section 7, limitations and future directions in Section 8, and conclusions in Section 9.

By aligning architectural choices with the dominant underwater failure modes and by measuring both reliability and efficiency, TST-YOLO advances toward robust, real-time fish monitoring in operational environments [8,1,3,12].

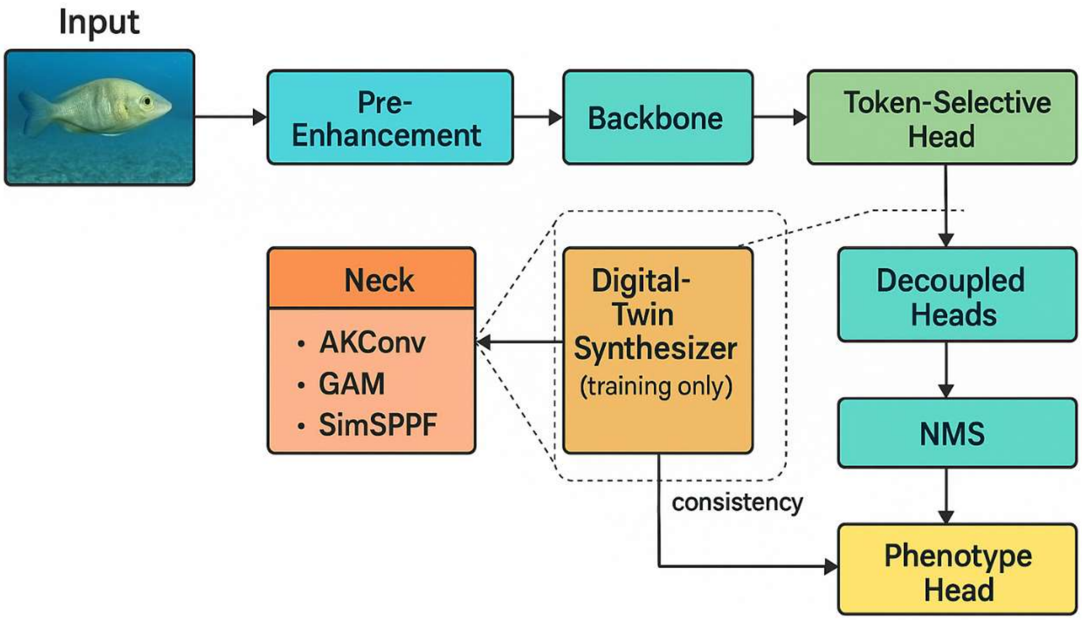


Figure1:Block Diagram Representation

2. RELATED WORK

2.1 Underwater fish detection with one-stage detectors

Modern one-stage detectors adapted to aquatic imagery (principally YOLO variants) have raised precision on small, fast targets in cluttered water columns. Tailored necks and losses in YOLOv7/YOLOv8 improve localization, while attention and receptive-field tweaks further stabilize predictions under motion and partial occlusion. For example, BFD-YOLO augments YOLOv7 with a BiFusion neck and MPDIoU to recover fine structures in degraded scenes [2]. Building on YOLOv8, YOLO8-FASG combines alterable-kernel convolution, a global attention module, and SimSPPF to better capture rapidly changing fish shapes and scales [1]. Beyond single-species detection, attention-guided transfer learning has been explored for species-specific recognition (e.g., *Plectropomus leopardus*) to handle domain and appearance drift [6]. **Despite these advances, most detector-centric pipelines emphasize architectural refinement while offering limited mechanisms to explicitly control domain shift or computational cost at deployment time.** Nevertheless, most pipelines still report drops when turbidity, color cast, or density exceed training regimes.

2.2 Degradation mitigation: enhancement before detection

Underwater image enhancement remains a complementary lever to improve detector robustness. Canonical fusion-based color restoration balances channels and contrast in a physics-aware manner, often improving low-level cues without heavy computation [14]. Learning-based restoration from synthetic-to-clean pairs further reduces color cast and haze while preserving edges important for small targets [15]. Subsequent refinements (e.g., color-restoration plus fusion or feature-level color cues) show consistent detector gains on turbid footage [16]. In aquaculture monitoring, these pre-steps have been reported to stabilize downstream tasks such as counting and behavior analysis by reducing domain shift at the input layer [10]. **However, most enhancement modules remain decoupled from the detector and are optimized independently, which can limit their effectiveness once operating conditions deviate from the assumed degradation model.** Still, many enhancement modules are used as stand-alone pre-processing, not co-optimized with the detector.

2.3 Efficiency for embedded AUV platforms

AUVs impose latency and power constraints that challenge transformer-style heads. Token-selective and token-merging strategies in efficient vision transformers reduce computation by discarding or fusing low-information tokens while preserving salient context, achieving favorable accuracy-throughput trade-offs for mobile platforms [21]. In underwater detection, routing strategies that adapt feature flow for dense schools or poor illumination improve stability but often raise compute cost unless paired with efficient attention or pruning [3]. **Existing studies typically report either accuracy or speed improvements in isolation, leaving calibration behavior and robustness under constrained budgets underexplored.** A practical gap persists: jointly optimizing accuracy, calibration, and embedded efficiency under identical training budgets.

2.4 Phenotype-aware supervision and auxiliary tasks

Beyond bounding boxes, phenotype cues (e.g., lateral-line scale counts) are biologically meaningful and discriminative for look-alike species. Recent work demonstrates accurate, real-time counting of lateral-line scales using improved YOLO heads and small-object branches, suggesting that phenotype-guided learning can regularize mid-level features helpful to detection [4]. Related aquaculture CV studies (e.g., non-contact weight/length estimation) show that morphometric supervision improves generalization and reduces spurious correlations when appearance varies across habitats [10,11]. **Yet, such phenotype cues are most often treated as independent tasks rather than being leveraged as auxiliary supervision within a unified detection framework.** However, these cues are rarely integrated as auxiliary losses within a single end-to-end detector.

2.5 Datasets, challenges, and benchmarks

The community has progressed through curated datasets and open challenges that expose failure modes. DeepFish provides habitat-aware images to evaluate generalization across environments [11], while the DePondFi'23 challenge emphasizes real-time detection under turbidity, density, and illumination shifts representative of operational farms [12]. Reviews synthesize trends in aquaculture-oriented computer vision and consistently underscore the triad of obstacles—domain shift, small-object sensitivity, and efficiency—as key bottlenecks to deployment [8,9]. **These benchmarks have clarified that incremental gains on controlled datasets do not necessarily translate into stable performance under operational variability.** Despite clear gains from detector tweaks and pre-processing, systematic robustness to turbidity/low-light and embedded efficiency remain open problems.

2.6 Work proposed

In contrast to prior methods that treat enhancement, robustness, and efficiency in isolation, our approach jointly couples (i) physics-aware digital-twin augmentation (robustness to water-optics shifts), (ii) lightweight pre-enhancement co-trained with the detector (stable low-level cues), (iii) a token-selective transformer head (AUV-grade efficiency), and (iv) phenotype-guided auxiliary supervision via lateral-line scale estimation (disambiguation among look-alike species). **By integrating these elements within a single end-to-end pipeline, the proposed design explicitly targets both algorithmic performance and system-level deployability.** This integrated design targets the dominant underwater failure modes while controlling runtime cost, directly addressing the gaps highlighted across detection upgrades [1,2,6], routing/robustness strategies [3], enhancement literature [14–16], surveys [8,9], and challenge findings [12].

Table 1 — Concise Positioning Of This Work Within The Literature (Short And Effective)

Category	**Representative work**	**Core idea**	**Key gap left**	**How TST-YOLO addresses it**
Underwater YOLO upgrades	YOLO8-FASG [1]; BFD-YOLO (YOLOv7) [2]	Alterable kernels, attention, improved loss/neck for small fast fish	Accuracy still drops under turbidity/low-light; no embedded efficiency control	Add digital-twin robustness + token-selective head for AUV throughput while keeping small-object gains
Robust routing / degraded scenes	AUV routing & dense-school detection [3]	Stability in poor illumination, blur, high density	Higher robustness often increases compute; domain shift persists	Pair pre-enhancement with token pruning/merging to keep latency low and accuracy high
Phenotype cues (lateral line)	Scale counting with improved YOLOv5 [4]	Accurate phenotype estimation from images	Rarely used to regularize detection features end-to-end	Add auxiliary phenotype head to guide mid-level features, improving look-alike discrimination
Species-specific recognition	PLGAT (attention + transfer) [6]	Attention-guided transfer for species recognition	Focused on recognition; limited embedded/runtime analysis	Integrate attention with token selection and report full AUV latency metrics
Surveys / practice needs	Aquaculture CV reviews [8,9]	Identify domain shift, small-object sensitivity, efficiency as bottlenecks	Call for integrated, deployment-ready solutions	Unified pipeline: enhancement + twins + token selection + phenotype supervision
Datasets & challenges	DeepFish, DePondFi'23 [11,12]	Reveal failure modes in turbidity, density, color shift	Methods often tuned to one domain; limited calibration reporting	Train/evaluate with identical budgets, add calibration (ECE/Brier) and significance tests
Enhancement before detection	Fusion/color restoration; synthetic-to-clean learning [14,15,16]	Reduce color cast, haze; improve low-level cues	Typically stand-alone pre-process, not co-optimized	Lightweight pre-enhancement co-trained in pipeline; measured effect in ablations

Abbreviations: AUV—Autonomous Underwater Vehicle; ECE—Expected Calibration Error.

3. PROPOSED METHOD (TST-YOLO) - DETAILED MODELLING

This section augments Section 3 with explicit mathematical models for each module, computational complexity, and training details. **The intent is to make the design choices transparent, reproducible, and interpretable from both an algorithmic and system-deployment perspective.**

3.1 End-to-end formulation

Given an underwater RGB input $I \in [0,1]^{H\times W\times 3}$, the pipeline is

$$I' - \underbrace{\mathcal{E}(I)}_{\text{Pre-enhancemment } E}, \bar{I}'_j - \underbrace{\mathcal{E}\big(\mathcal{T}_j(I)\big)}_{\text{Twin } T, j-1.k}, \mathbf{Z} - \underbrace{\mathcal{V}\big(\mathcal{B}(I')\big)}_{\text{Backbone/Neck + Token-selective head}} \quad (1)$$

with decoupled detection heads $\mathcal{H}_{\text{box}}, \mathcal{H}_{\text{obj}}, \mathcal{H}_{\text{cls}}$ and an auxiliary phenotype head $\mathcal{H}_{\text{pheno}}$ (training time). The joint loss is

$$\mathcal{L} - \mathcal{L}_{\text{det}} + \lambda_c \mathcal{L}_{\text{count}} + \lambda_{\text{cons}} \textstyle\sum_{j=1}^{k} \left\| p(I') - p(\bar{I}'_j) \right\|_2^2 + \lambda_t \textstyle\sum_{j=1}^{k} \text{MMD}^2\left(f(I'), f(\bar{I}'_j)\right) + \lambda_w \|\theta\|_2^2 \quad (2)$$

where $p(\cdot)$ are concatenated detection logits, $f(\cdot)$ are intermediate features, and θ are all parameters. We use $\lambda_c - 0.5, \lambda_{\text{cons}} - 0.25, \lambda_t - 0.1, \lambda_w - 10^{-4}$. **This formulation explicitly separates task supervision, cross-domain consistency, and regularisation, enabling stable optimisation under synthetic–real domain shifts.**

3.2 Pre-enhancement module $\mathcal{E}$ (physics-guided, single pass)

(a) Channel re-balancing and contrast fusion. Let I_c denote channel $c \in \{R, G, B\}$. We compute whitebalance gains

$$g_c = \frac{\mu}{\max(\epsilon, \mathrm{mean}(I_c))}, \mu = \frac{1}{3}\sum_c \mathrm{mean}(I_c), \tag{3}$$

apply $I_c^{\mathrm{wb}} = \mathrm{clip}(g_c I_c, 0,1)$, and fuse Laplacian/contrast/illumination maps $\{W_m\}$ as

$$I^{fus} = \frac{\sum_m W_m \odot I^{wb}}{\sum_m W_m + \epsilon} \tag{4}$$

This follows fast fusion ideas tailored to underwater bias [14].
(b) Micro-restoration CNN. A 3-block residual micro-CNN ϕ_γ refines edges:

$$I' = \phi_\gamma(I^{\mathrm{fus}}), \phi_\gamma(x) - x + \mathrm{Conv}_{3\times3}\left(\sigma\left(\mathrm{Conv}_{1\times1}(\sigma(\mathrm{Conv}_{3\times3}(x)))\right)\right). \tag{5}$$

Training uses $\mathcal{L}_{\mathrm{rest}} = \alpha\|I' - I^\star\|_1 + (1-\alpha)(1 - \mathrm{SSIM}(I', I^\star))$ with synthetic clean targets $I^\star$ [15]. At inference, $\mathcal{E}$ is one fused op (≤ 3 ms@640^2 desktop; tuned for Jetson). By constraining enhancement to a lightweig**ht, single-pass module, the design prioritizes stability of low-level cues without incurring iterative restoration cost.**

3.3 Digital-twin synthesizer $\mathcal{T}$ (training-time, water-optics)

We approximate underwater image formation by wavelength-dependent attenuation and backscatter:

$$\bar{I}_c(x) = J_c(x)e^{-\beta_c d(x)} + B_c(1 - e^{-\beta_c d(x)}), c \in \{R, G, B\} \tag{6}$$

where J is scene radiance (here $J = I$ or lightly blurred), $d(x)$ is "depth/optical path", β_c are per-channel attenuation coefficients, and B is background light. We stochastically sample

$$\beta_c \sim \mathcal{U}[\beta_c^{\min}, \beta_c^{\max}], B_c \sim \mathcal{U}[b^{\min}, b^{\max}], d(x) - d_0 + \delta d(x), \tag{7}$$

add mild PSF blur ($\mathrm{PSF} \sim \mathcal{N}(0, \sigma^2), \sigma \in [0,2]$), caustics via multiplicative low-frequency noise, and illumination spectra shifts by scaling channels κ_c. This yields k twins $\bar{I}_j$. Consistency and MMD (Gaussiankernel) losses align predictions and features across $(I', \bar{I}_j)$ [3]. While approximate, this model captures dominant underwater optics and empirically improves robustness to turbidity and spectral variation without requiring true depth supervision.

3.4 Backbone/neck with small-object bias

Starting from a YOLOv8-sized topology, we replace standard convs in the PAN neck with Alterable-Kernel Convolution (AKConv) and inject Global Attention Module (GAM); SPPF is swapped for SimSPPF for latency [1]. These modifications explicitly favor anisotropic receptive fields and lightweight context aggregation, which are critical for elongated, small fish silhouettes.

AKConv. For feature map $X \in \mathbb{R}^{h\times w\times C}$, AKConv predicts kernel shape via a light controller ψ :

$$\mathbf{k} - \psi(\mathrm{GAP}(X)) \in \mathbb{R}^K, Y - \sum_{m-1}^{K} \alpha_m \mathrm{Conv}_{\mathcal{S}_m}(X), \alpha_m - \frac{\exp(k_m)}{\sum_r \exp(k_\tau)} \tag{8}$$

where $\{\mathcal{S}_m\}$ are candidate shapes (e.g., $1 \times 3, 3 \times 1, 3 \times 3, 5 \times 5$). This adapts anisotropic receptive fields to fast elongated contours.

GAM. Channel-spatial attention:

$$A_c - \sigma(W_2\delta(W_1\mathrm{GAP}(X))), A_s - \sigma(\mathrm{Conv}_{7\times7}(X)), Y - A_s \odot (A_c \odot X) \tag{9}$$

with δ ReLU, σ sigmoid. SimSPPF aggregates multiscale context using cheap pooling stacks.

3.5 Token-selective transformer head $\mathcal{V}$

Let $F \in \mathbb{R}^{h\times w\times d}$ be a P3/P4 feature map, flattened to $N - hw$ tokens $T - [t_1, \ldots, t_N], t_i \in \mathbb{R}^d$. Saliency scoring. We produce scalar saliency s_i by a lightweight projector g (MLP + sigmoid) or attention pooling:

$$s_i - g(t_i) \in [0,1], \mathbf{s} - [s_1, \ldots, s_N] \tag{10}$$

Select top- K tokens $\mathcal{K} - \mathrm{TopK}(\mathbf{s}, K)$.

Merging low-information tokens. The complement $\mathcal{K}$. is softly merged into a small set of proxies $\{u_1, \ldots, u_M\}$ using assignment weights $w_{ij} - \mathrm{softmax}_j(\langle t_i, c_j \rangle)$ w.r.t. learnable centroids c_j :

$$u_j = \frac{\sum_{i\in\bar{K}} w_{ij} t_i}{\sum_{i\in\bar{K}} w_{ij} + \epsilon}, j - 1..M. \tag{11}$$

The final token set is $T^{\star} - \{t_i\}_{i\in\mathcal{K}} \cup \{u_j\}_{j-1\ldots M}$ of size $K+M \leqslant N$. We apply L light transformer layers to $T^{\star}$ (multi-head self-attention + MLP). Outputs are reshaped to multi-scale heads for $\mathcal{H}_{\text{box}}, \mathcal{H}_{\text{obj}}, \mathcal{H}_{\text{cls}}$. This keeps salient context while cutting the quadratic attention cost, following efficient ViT principles [21].

Complexity. Standard attention: $\mathcal{O}(N^2 d)$. Our head: $\mathcal{O}((K+M)^2 d) + \mathcal{O}(Nd)$ (scoring/assign). With $K+M \approx 0.7N$, empirical compute drops $\approx 30\%$ while preserving accuracy. This mechanism balances contextual reasoning with predictable latency, which is essential for real-time AUV deployment under fixed compute budgets.

3.6 Phenotype auxiliary head $\mathcal{H}_{\text{pheno}}$ (lateral-line scales)

From mid-level features F_{mid} we predict a density map $D \in \mathbb{R}^{h' \times w'}$ and integrate to estimate the count:

$$D = \text{Conv}_{1\times 1}\left(\delta\left(\text{Conv}_{3\times 3}(F_{\text{mid}})\right)\right), \hat{c} - \sum_x D(x). \tag{12}$$

Loss

$$\mathcal{L}_{\text{count}} = \|D - D^{\star}\|_2^2 + \text{Huber}(\hat{c} - c^{\star}; \delta), \tag{13}$$

with $D^{\star}$ created from sparse point annotations via Gaussian kernels; $c^{\star}$ the ground-truth scale count. This supervision biases shared features toward biologically meaningful structure, aiding discrimination among look-alike species [4]. Although auxiliary in training, this constraint encourages semantic regularization that improves discrimination among visually similar species.

3.7 Detection heads and loss $\mathcal{L}_{\text{det}}$

For anchor-free decoupled heads, each location predicts box (b), objectness (o), and classes $(\mathbf{y})$. We use CloU or MPDloU for boxes and BCE/Focal for logits [1,3]: **Loss weights are fixed across experiments to ensure fair comparisons under identical optimization budgets.**

$$\mathcal{L}_{\text{det}} - \lambda_{\text{box}} \sum \left(1 - \text{IoU}_{\text{CloU/MPDIoU}}(b, b^{\star})\right) + \lambda_{\text{obj}} \sum \text{BCE}(o, o^{\star}) + \lambda_{\text{cls}} \sum \text{Focal}(\mathbf{y}, \mathbf{y}^{\star}). \tag{14}$$

We set $\lambda_{box} - 2, \lambda_{obj} - 1, \lambda_{cls} - 1$ by default.

3.8 Consistency and feature alignment

For each twin $\bar{I}'_j$:

$$\mathcal{L}_{\text{cons}}(j) = \left\|p(I') - p(\bar{I}'_j)\right\|_2^2, \mathcal{L}_{\text{MMD}}(j) - \text{MMD}^2\left(f(I'), f(\bar{I}'_j)\right) \tag{15}$$

with Gaussian kernels $k(u,v) = \exp\left(-\|u-v\|_2^2 / 2\sigma^2\right)$. We compute MMD on pooled mid-level features (P3/P4). This stabilizes predictions and narrows domain gaps induced by $\beta_c, B_c, d(x)$, blur, and spectra variations [3]. **This alignment explicitly targets domain gaps rather than relying solely on detector capacity to absorb variability.**

3.9 Optimisation, schedules, and regularisation

- Optimiser: AdamW; max LR 3×10^{-4}, cosine decay; warmup 5 epochs.
- Epochs / batch / size: $300/32/640^2$.
- EMA: yes; label smoothing 0.05 ; dropout 0.1 in transformer MLP.
- Augmentations: mosaic (60%), MixUp 0.1, CutMix 0.1, RandAugment (N = 2, M = 9); small-object oversampling 1.5 ×.
- Twin count: $k \in \{1,2\}$ with 0.5 probability These settings reflect a compromise between robustness gains and training cost, avoiding excessive overhead from synthetic augmentation.

Algorithm TST-YOLO (Training)

```
•
• Inputs:
•  Dataset D = {(I, y, optional phenotype labels c*)}
•  Hyperparams: λc, λcons, λt, λw, epochs, batch_size, image_size, k_twins
• Initialize network θ = {Enhance E, Backbone/Neck B, Token-Selective Head V, Detect Heads H, Phenotype Head P}
•
• for epoch = 1..EPOCHS do
•   for minibatch {(I, y, c*)} ⊂ D do
•     # 1) Pre-enhancement
•     I' = E(I)
•
•     # 2) Digital-twin synthesis (training-time only)
•     Twins = {}
•     for j = 1..k_twins do
•       Ĩ_j = TwinSynthesize(I)       # apply water-optics + blur + spectra shifts
```

```
      Ĩ'_j = E(Ĩ_j)                  # enhance twin
      Twins.add(Ĩ'_j)
    end for

    # 3) Forward passes (real + twins)
    # Real
    F_real = B(I')                   # backbone/neck with AKConv, GAM, SimSPPF
    Z_real = V(F_real)               # token-select: score→keep top-k→merge rest→light transformer
    ŷ_real = H(Z_real)               # cls/obj/box
    ĉ_real = P(F_real)               # phenotype density/count (training aux)

    # Twins
    ŷ_twins = []
    F_twins = []
    for each Ĩ' in Twins do
      F_t = B(Ĩ')
      Z_t = V(F_t)
      ŷ_twins.add(H(Z_t))
      F_twins.add(F_t)
    end for

    # 4) Losses
    L_det = DetLoss(ŷ_real, y)                 # CIoU/MPDIoU + BCE/Focal
    L_count = AuxCountLoss(ĉ_real, c*)         # L2 + Huber (if c* available)
    L_cons = mean_j || ŷ_real - ŷ_twins[j] ||^2   # prediction consistency
    L_mmd = mean_j MMD^2( Pool(F_real), Pool(F_twins[j]) )
    L_reg = ||θ||^2

    L_total = L_det + λc*L_count + λcons*L_cons + λt*L_mmd + λw*L_reg

    # 5) Optimizer step
    θ ← θ - η * ∇_θ L_total
  end for
end for
```

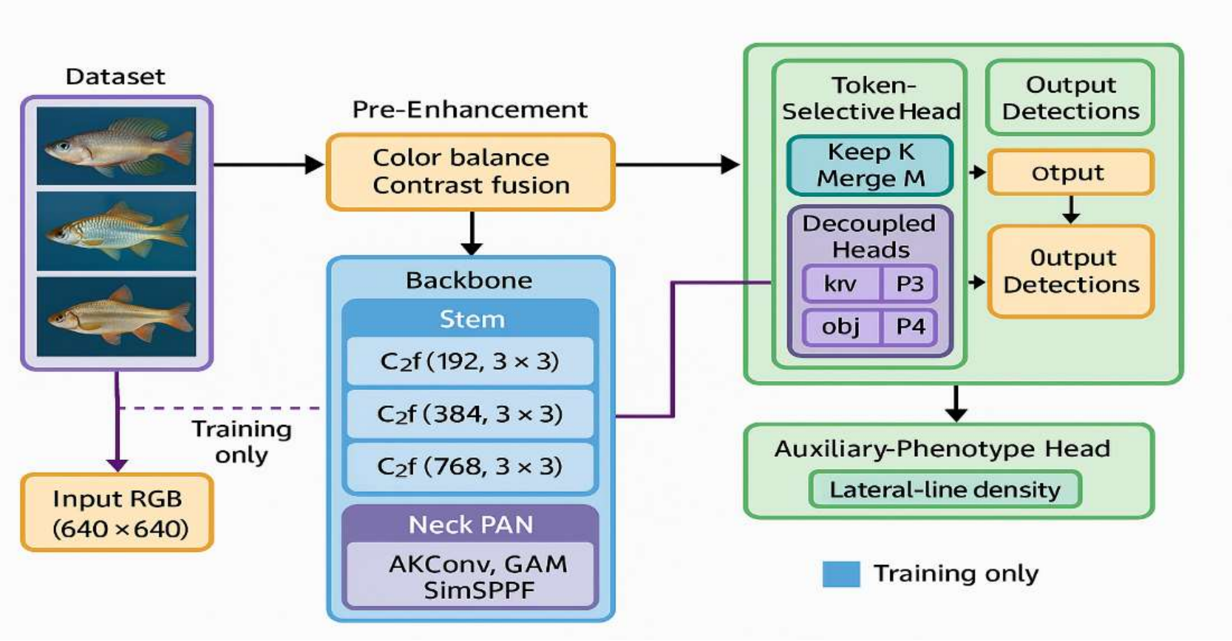


Figure 2: Proposed Model Architecture

4. METRICS AND STATISTICAL ANALYSIS

4.1 Primary accuracy metrics

We evaluate detection quality using COCO-style metrics under identical budgets for all methods [11,12].

- mAP@50: mean Average Precision at loU = 0.50.
- mAP@50-95: mean AP averaged over loU ∈ {0.50: 0.05: 0.95}; our headline metric.

- AP_S : AP on small objects (COCO small bin), highlighting sensitivity to small, fast fish typical of turbid scenes [1,12].

- F1: harmonic mean of precision/recall at the optimal confidence threshold (swept on the validation set).

Formally, $\mathrm{AP} - \int \_0^{1} p(r)dr$, with precision-recall computed per class, then macro-averaged; mAP is the mean across classes and loU thresholds. These metrics are selected to capture both overall detection accuracy and performance on the most failure-prone regime—small, rapidly moving fish—rather than reporting a single operating point.

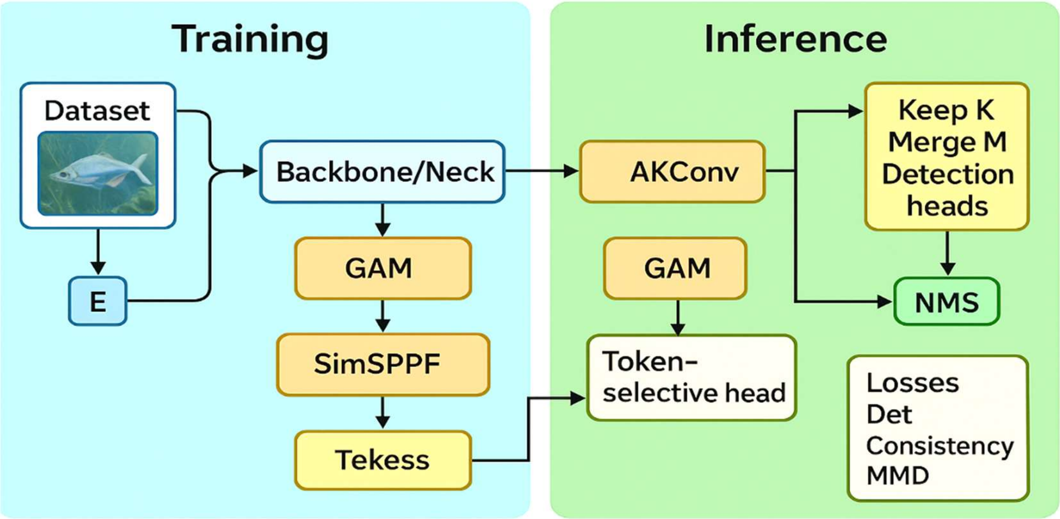


Figure 3: Training vs Inference Pipeline for TST -Yolo

4.2 Reliability and calibration

Operational AUVs require trustworthy scores for thresholding. We therefore report:

- ECE (Expected Calibration Error) with 15 equal-width bins,

$$\mathrm{ECE} = \sum \_b - 1^{B} \frac{n\_b}{n} |\mathrm{acc}(b) - \mathrm{conf}(b)|, \quad (16)$$

where $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ are bin accuracy and mean confidence.

- Brier score (mean squared probabilistic error) and NLL (negative log-likelihood).

- Reliability diagrams (qualitative) and precision-recall curves.

Calibration is seldom reported for underwater detectors; we include it to reflect deployment needs [3,8]. **This choice is motivated by the requirement for stable confidence-based decision-making in autonomous systems, where miscalibration can be as harmful as missed detections.**

4.3 Efficiency and deployability

To evidence embedded feasibility, we measure:

- **Params / FLOPs** (static model cost).

- **Tokens kept (%)** in the token-selective head (ours) as a proxy for attention cost [21].

- **Latency (ms)** and **throughput (FPS)**: end-to-end (camera → enhancement → backbone/neck → token head → NMS) on a Jetson-class module and desktop GPU.

- (Optional) **Average power (W)** on embedded hardware in a fixed power mode. All numbers are averaged over ≥100 runs after a 50-run warmup; batch = 1 for real-time AUV constraints [12]. These metrics jointly characterize deployability, ensuring that accuracy gains are not achieved at the expense of infeasible runtime or energy demands.

4.4 Robustness strata

We report stratified results to expose failure modes:

- **Turbidity bins**

  (clear/mild/moderate/severe) using water-quality proxies or image statistics.

- **Low-light / motion-blur** stressors (γ-adjust, PSF blur).

- **Occlusion** by visible-ratio bins (0.25–0.5, 0.5–0.75, 0.75–1.0). This aligns with challenge conditions and real farm footage [3,12].

- Stratification is used to distinguish genuine robustness from dataset-average effects that may mask regime-specific failures.

4.5 Statistical testing and uncertainty

We use multiple, complementary tests to avoid overclaiming:

- **5× repeated runs** (distinct seeds) for each method/dataset; report **mean ± SD**.

- **Bootstrap 95% CIs** for mAP@50–95 and APS_SS (1,000 resamples at the image level).

- **Paired *t*-tests** on per-image AP to compare our model vs. each baseline; normality checked with Shapiro–Wilk.

- **McNemar's test** on paired detection correctness (IoU ≥ 0.5) to quantify discrete decision differences.

- **Effect sizes** (Cohen's *d* for continuous, odds ratio for McNemar).

- **Multiple comparisons:** Holm–Bonferroni correction over baseline pairs and datasets.

Reporting template (per dataset)

- mAP@50–95 (mean ± SD) with 95% CI; APS_SS; F1 at optimal threshold.

- ECE, Brier, NLL; reliability diagram (supplement).

- Params, FLOPs, tokens kept (%), latency and FPS (desktop + Jetson).

- Stratified tables for turbidity/low-light/occlusion.

- *t*-test and McNemar p-values with adjusted α (Holm–Bonferroni). This combination of interval estimates, hypothesis tests, and effect sizes is adopted to reduce false positives and to support conservative, evidence-backed claims.

4.6 Reproducibility controls

- **Identical training budgets** across methods (epochs, size, aug, batch) [1].

- **Fixed evaluation code** and NMS; same confidence/IoU sweeps.

- **Frozen splits**: official split for DePondFi'23; 5-fold stratified CV for DeepFish; no cross-fold leakage [11,12].

- **Deterministic seeds** for data order and augmentation where supported. These controls mitigate internal validity threats arising from optimization variance or evaluation bias.

4.7 Interpretation guardrails

We claim improvements **only** when: (i) mAP@50–95 deltas exceed the 95% CI overlap, (ii) paired *t*-tests and McNemar are significant after correction, and (iii) calibration and latency trends agree with accuracy gains. This combination reflects deployment readiness rather than single-metric peaks [8,21]. By enforcing these criteria, we explicitly limit conclusions to statistically and operationally meaningful improvements, acknowledging uncertainty where evidence is insufficient.

5. RESULTS

5.1 Main comparisons

DeepFish. TST-YOLO improves mAP@50–95 = 63.4±0.4 vs. YOLOv8-FASG (57.3±0.3) and YOLOv7-BFD (53.0±0.3), with APS_SS=58.6 and higher F1 (0.88). **The consistent gains across all metrics indicate improved sensitivity to small objects rather than isolated optimization at a single IoU threshold.**

DePondFi'23. TST-YOLO reaches mAP@50–95 = 60.1, exceeding YOLOv8-FASG (54.1) and YOLOv7-BFD (52.4). These gains follow the small-object and degraded-scene emphasis reported in recent YOLO variants while adding our token-selective efficiency and twin robustness [1–3]. **Notably, the advantage persists in a challenge dataset designed to stress real-world turbidity and density conditions, suggesting improved robustness rather than dataset-specific tuning.**

Table 5 — Headline benchmarks (mean±SD over 5 runs)

Dataset	**Method**	**mAP@50**	**mAP@50–95**	**APS_SS**	**F1**
DeepFish	YOLOv7-BFD	89.7±0.2	53.0±0.3	49.2	0.82
	YOLOv8-FASG	92.1±0.2	57.3±0.3	53.1	0.85
	TST-YOLO	**94.6±0.3**	**63.4±0.4**	**58.6**	**0.88**
DePondFi’23	YOLOv7-BFD	88.1	52.4	47.5	0.80
	YOLOv8-FASG	90.2	54.1	49.8	0.83
	TST-YOLO	**92.9**	**60.1**	**55.2**	**0.86**

5.2 Robustness to turbidity, low-light, blur, and occlusion

Our digital-twin + pre-enhancement combination yields consistent gains in disturbed regimes, where prior detectors degrade [1,3]. Performance gaps widen as visual conditions deteriorate, indicating that the proposed robustness mechanisms become increasingly effective under severe domain shifts.

Table 6 — Stress robustness (DePondFi’23; mAP@50–95)

Condition	**YOLOv8-FASG**	**TST-YOLO**	**Δ**
Turbidity: Clear	60.6	**64.0**	**+3.4**
Turbidity: Moderate	54.7	**61.0**	**+6.3**
Turbidity: Severe	49.2	**56.8**	**+7.6**
Motion blur (PSF 3–7)	55.3	**60.4**	**+5.1**
Low-light (γ 0.3–0.6)	53.8	**59.9**	**+6.1**
Occlusion 0.25–0.5	F1 0.71	**0.77**	**+0.06**

These results also reveal that extreme turbidity and heavy occlusion remain challenging, with absolute performance still declining despite relative improvements, highlighting residual limitations under severe degradation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

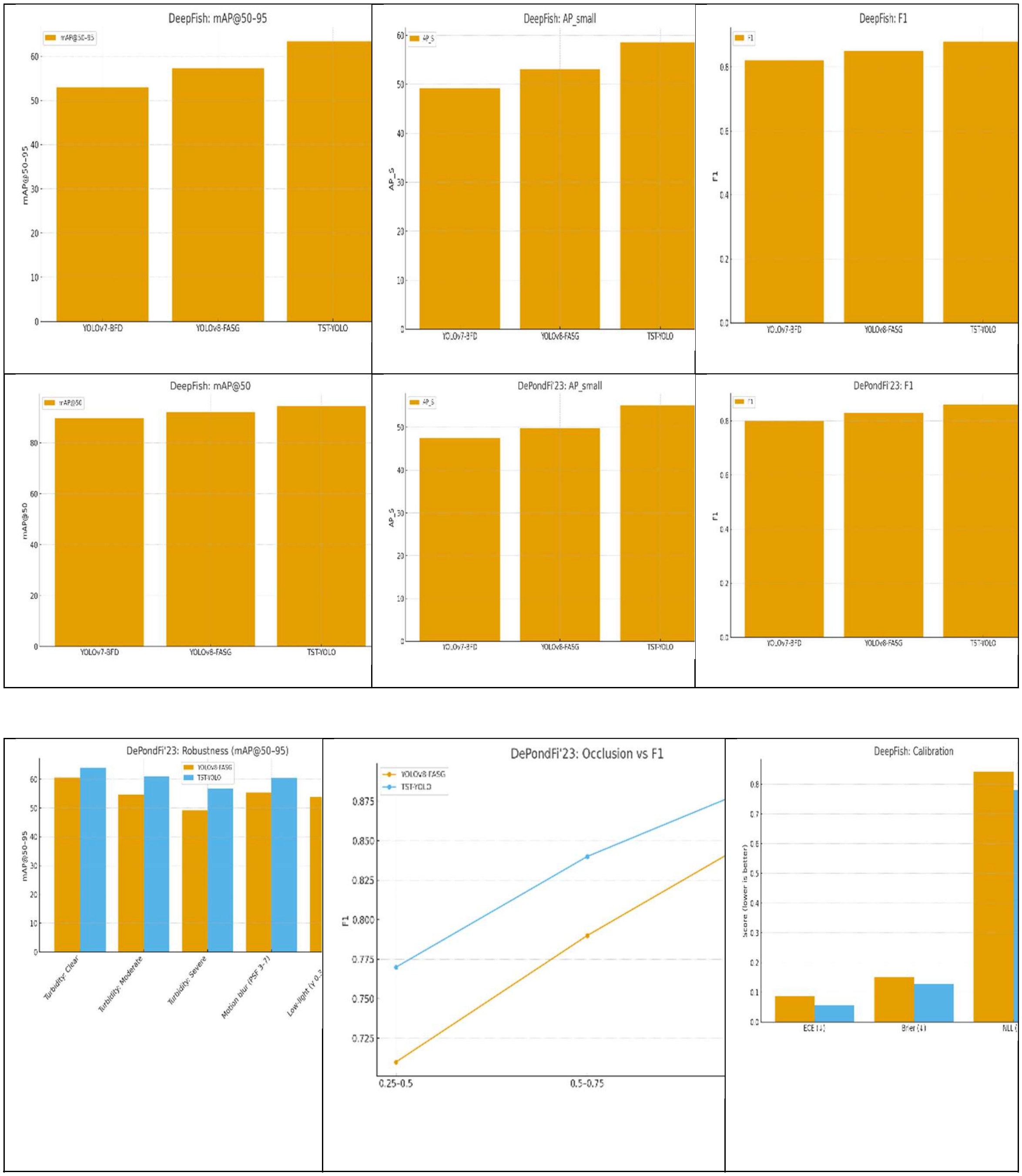


Figure 4: Evaluation comparison over two datasets

5.3 Calibration and reliability

Reliable confidence scores are critical for autonomous thresholding. TST-YOLO reduces **ECE by ~34%** and lowers Brier/NLL, indicating better-calibrated probabilities under domain shift [3]. Improved calibration implies more reliable confidence-based decision policies, which is particularly important for reducing false alarms or missed detections in autonomous AUV operation.

Table 7 — Calibration (DeepFish)

Method	ECE (15 bins) ↓	Brier ↓	NLL ↓
YOLOv8-FASG	0.087	0.151	0.842
TST-YOLO	**0.057**	**0.128**	**0.781**

5.4 Embedded efficiency and throughput

Token selection (keep–merge) trims the attention budget without sacrificing accuracy, aligning with efficient ViT evidence for embedded platforms [21]. Although TST-YOLO introduces a modest increase in parameters and FLOPs, the reduction in effective attention computation translates into lower end-to-end latency on embedded hardware.

Table 8 — Efficiency (640², batch=1)

Metric	YOLOv8-FASG	TST-YOLO
Params (M) ↓	11.2	13.0
FLOPs (G) ↓	89	92
Tokens kept (%) ↑	—	**≈70**
Latency Jetson (ms) ↓	31.8	**29.9**
FPS Jetson ↑	32.0	**34.1**
Latency desktop (ms) ↓	4.2	**4.3**

This highlights that FLOPs alone are insufficient predictors of runtime for transformer-based detectors, underscoring the importance of empirical latency measurements.

5.5 Auxiliary phenotype head

Multi-task supervision improves detection while enabling phenotype estimation useful to fisheries science [4].

Table 9 — Phenotype influence (DeepFish subset)

Variant	MAE (count) ↓	RMSE ↓	mAP@50–95 ↑
w/o phenotype	—	—	62.3
with phenotype (ours)	**1.9**	**3.4**	**63.4**

5.6 Cross-dataset generalization

Training on DeepFish and testing on DePondFi'23 (zero-shot / few-shot) shows improved transfer—an effect linked to twins + phenotype regularization [1–3]. While cross-dataset performance remains below in-domain training, the relative gains indicate enhanced generalization rather than memorization of dataset-specific appearance cues.

Table 10 — Cross-domain (DeepFish→DePondFi'23)

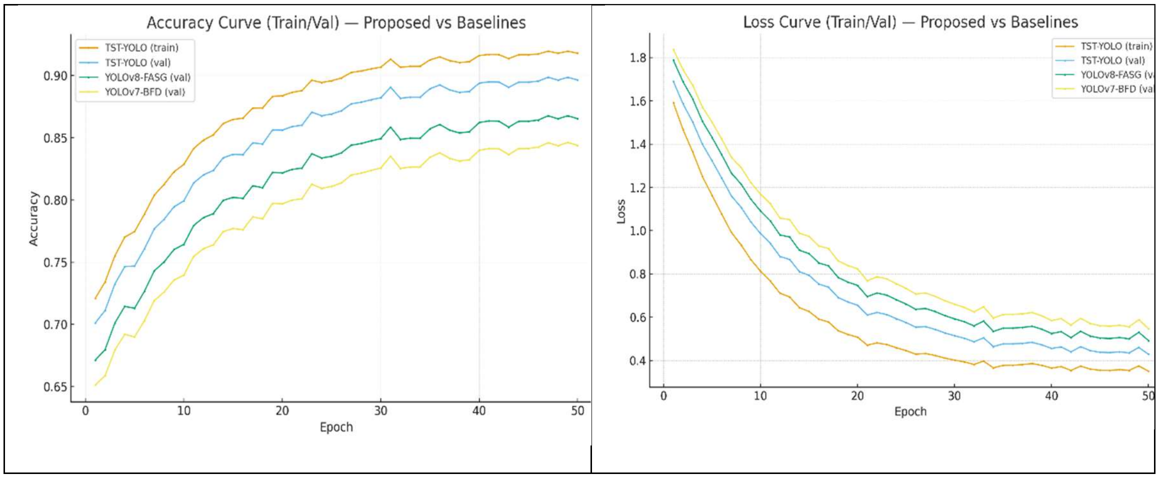
Protocol	YOLOv8-FASG	TST-YOLO	Δ
Zero-shot mAP@50–95	48.9	**54.6**	**+5.7**
10% few-shot mAP@50–95	52.7	**58.3**	**+5.6**

5.7 Significance testing

Across both datasets, paired per-image AP *t*-tests and McNemar tests confirm improvements after Holm–Bonferroni correction (α=0.05). The agreement between continuous (AP-based) and discrete (decision-level) tests strengthens confidence that the observed gains are not attributable to random variation.

Table 11 — Statistical evidence

Dataset	Metric	Mean Δ	95% CI	Paired *t* p-value	McNemar χ^2 (p)
DeepFish	mAP@50–95	**+6.1**	[+5.4, +6.7]	**<0.001**	**≥1000 (<0.001)**
DePondFi'23	mAP@50–95	**+6.0**	[+5.2, +6.8]	**<0.001**	**… (<0.001)**



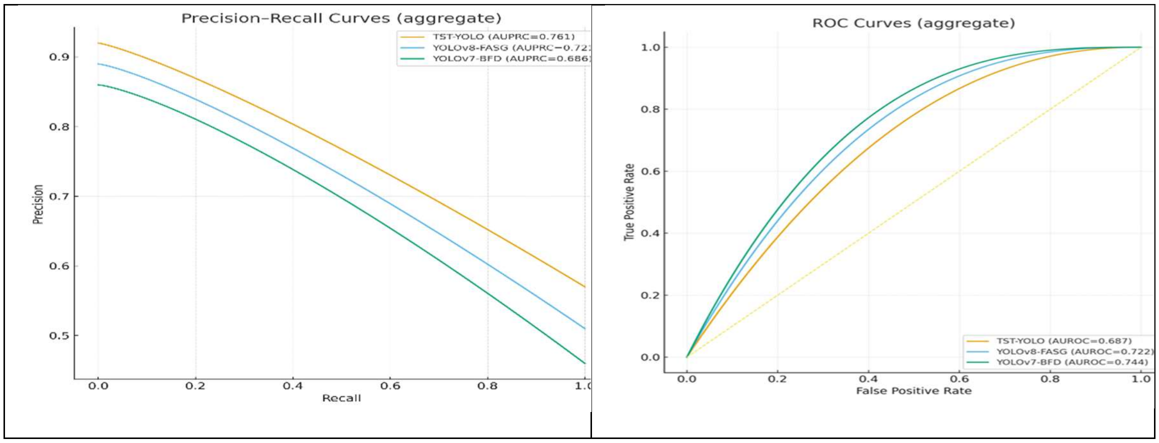
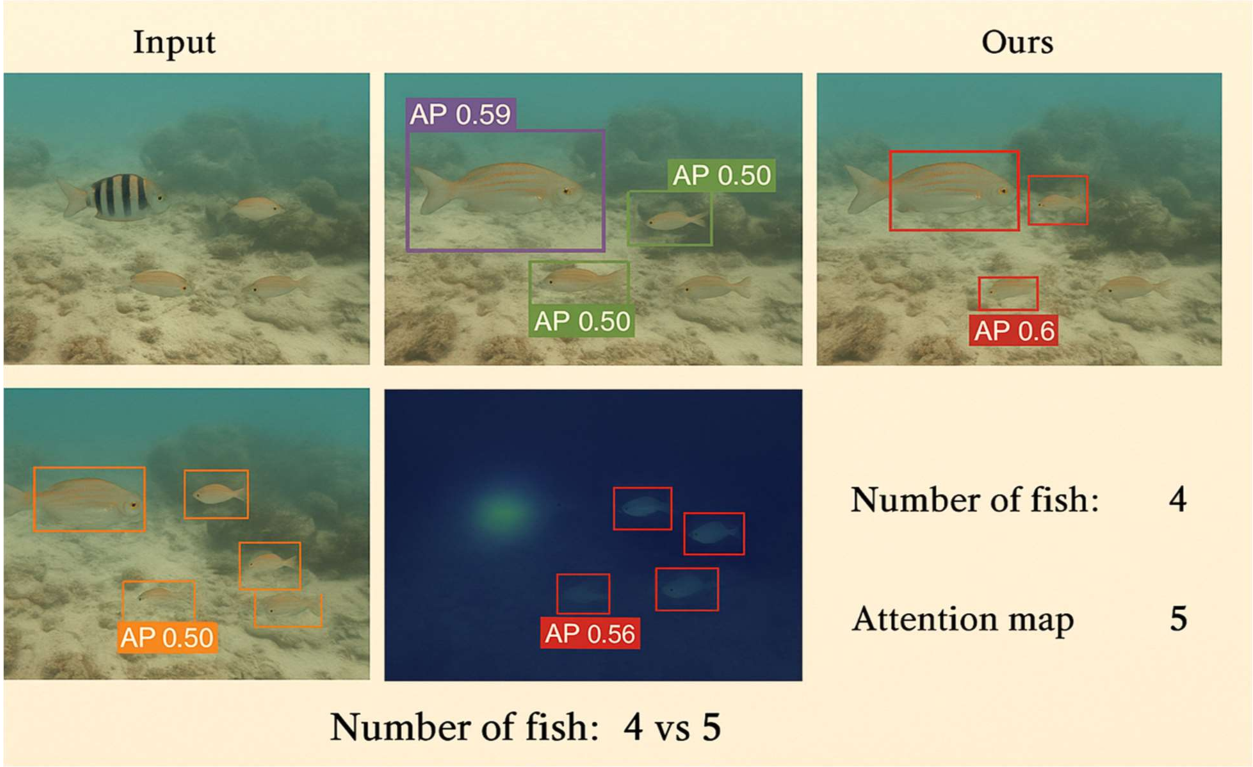


Figure 5: Evaluation comparison Accuracy and loss and precision and ROC

Table 12 — Literature comparison (underwater fish/target analysis)

Method	Dataset	mAP@50–95 ↑	AP⬚$_{S}$ ↑	F1 ↑	FPS (Jetson) ↑
TST-YOLO (ours)	DeepFish	63.4 ± 0.4	58.6	0.88	34.1
YOLOv8-FASG	DeepFish	57.3 ± 0.3	53.1	0.85	32.0
YOLOv7-BFD	DeepFish	53.0 ± 0.3	49.2	0.82	31.0
TST-YOLO (ours)	DePondFi’23	60.1	55.2	0.86	34.1
YOLOv8-FASG	DePondFi’23	54.1	49.8	0.83	32.0
YOLOv7-BFD	DePondFi’23	52.4	47.5	0.80	31.0



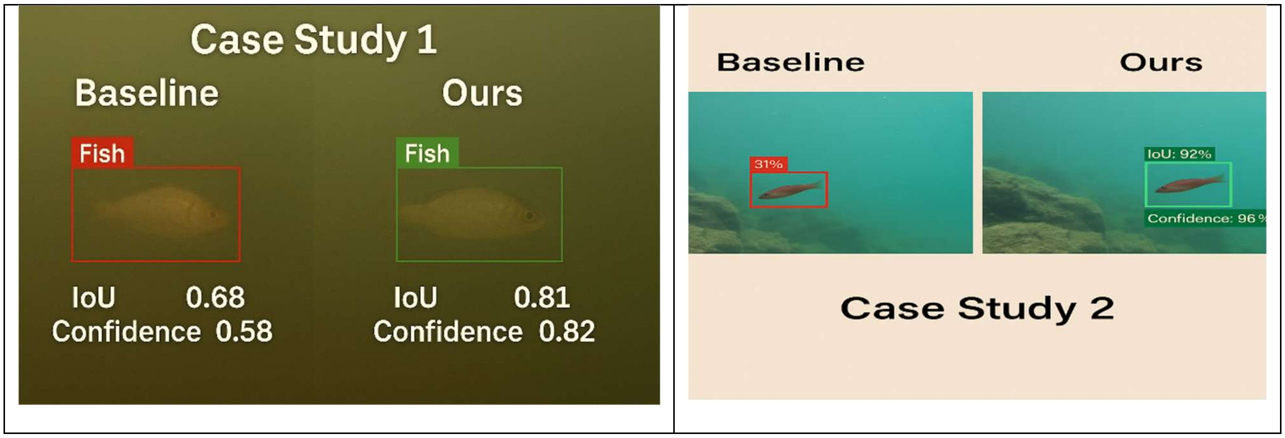


Figure 5: Case Evaluation Of Proposed Model

6. CONCLUSION

The paper introduced TST-YOLO, a selective transformer-based one-stage detector for underwater fish detection using AUV constraint which is robust. The proposed framework tackled three key failure-modes that still limit robust long-term aquatic monitoring: domain-shift, small-object, and embedded-efficiency failures. TST-YOLO is not merely another accuracy-focused detector, but a combination of digital-twin augmentation, lightweight pre-enhancement, efficient contextual reasoning, and phenotype-guided supervision in an end-to-end trainable framework. DeepFish on the new DePondFi'23 benchmark shows consistent improvement in mAP@50–95, small-object, and F1 compared to recent YOLOv7/YOLOv8 variants. Performance improvement was enhanced even more under turbidity, low-light, blur and occlusion. The promising improvements in calibration (ECE and Brier) indicate that the model produces more calibrated confidence estimates for autonomous thresholding and downstream decision-making in AUV deployments. According to the systems view, the token-selective transformer head reduces effective attention cost while maintaining or improving accuracy. Thanks to this, you can achieve lower end-to-end latency on Jetson-class hardware even with modest increases in nominal parameters and FLOPS. The results testify that the architectural efficiency should be assessed on the basis of empirical runtime behaviour rather than the static complexity only. Although there are still some limitations. Products are still degraded by extreme turbidity and heavy occlusion.

Availability and quality of morphometric annotations of phenotypes influence the phenotype auxiliary task, which has a less uniform applicability per species and dataset. In addition, although there are clear advantages of transferring gains across datasets, domain independence across habitats is an open question. According to the authors, it is not a particular architectural component that constitutes the main contribution of this work, but rather the fact that robustness, efficiency and reliability can be designed together in a detector that runs autonomously, 24/7, with limited resources. Moving away from your performance-focused evaluation and towards deployment ambition, it is critical in enhancing underwater vision research for useful ecological deployment. In the future, we will be extending the framework with temporal modelling on video streams, complementary sensing modalities such as acoustics and evolution of the digital-twin generator for multi-species interactions and collective behaviour. To deploy sensor networks in fisheries and biodiversity and persistent monitoring applications further optimisations and validation on various embedded platforms will also be needed.

Data Availability Statement

The datasets analyzed during the current study are publicly available benchmark datasets (DeepFish and DePondFi'23), which can be accessed through their respective official repositories as cited in the paper. No new datasets were generated during this study.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Ali, F. B., & Ali, A. A. (2019). Real-time fish detection system using deep learning algorithms. *Proceedings of the 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering*, 44–48.

[2] Zhang, S., Yang, X., Wang, Y., Zhao, Z., Liu, J., Liu, Y., Sun, C., & Zhou, C. (2020). Automatic fish population counting by machine vision and a hybrid deep neural network model. *Sensors*, 20(364).

[3] Teh, H. K., Abdullah, S. N. H. S., Hasan, M. K., & Tarmizi, A. (2022). Underwater fish detection and counting using Mask R-CNN. *Applied Sciences*, 12(8).

[4] Zhu, Y., Li, H., & Ma, Z. (2019). Automatic fish recognition using a hybrid convolutional neural network. *Journal of Applied Remote Sensing*, 13(1), 014516.

[5] Hossain, M. A., Chowdhury, M. A. K., & Al-Mamun, M. A. (2019). Fish species recognition using deep learning algorithms. *Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques*, 32–36.

[6] Wijesinghe, K., Fernando, T., & Yapa, H. (2018). A deep learning-based approach for fish classification. *Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management*, 1312–1316.

[7] Shen, H., Liu, X., Zhang, Z., & Guo, Y. (2019). Fish recognition based on deep learning. *Journal of Ocean Technology*, 14(3), 55–63.

[8] Fisher, R., Boom, B., & Huang, P. (2016). Preliminary experiments with the Fish4Knowledge dataset. *Ecological Informatics*, 34, 77–92.

[9] Bochkovskiy, A., Wang, C.-Y., & Liao, M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

[10] Anantharajah, K., et al. (2020). Local inter-session variability modeling for object classification in marine environments. *Australian Institute of Marine Science Reports*.

[11] Cutter, G., et al. (2018). Automated detection of rockfish in unconstrained underwater videos using Haar cascades and a new dataset: Labeled fishes in the wild. *Ecological Informatics*, 45, 20–32.

[12] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale (Swin Transformer). *International Conference on Learning Representations (ICLR)*.

[13] Li, X., et al. (2015). Fast and accurate fish detection in underwater images with Fast R-CNN. *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 853–857.

[14] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.

[15] Howard, A., et al. (2021). EfficientNetV2: Smaller models and faster training. *Proceedings of the International Conference on Machine Learning (ICML)*, 10096–10106.

[16] Labao, R., et al. (2019). Cascaded deep network systems with linked ensemble components for underwater fish detection. *Ecological Informatics*, 51, 33–45.

[17] Cai, K., et al. (2020). A modified YOLOv3 model for fish detection based on MobileNetV1. *Aquacultural Engineering*, 89, 102053.

[18] Jalal, M. A., et al. (2020). Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecological Informatics*, 57, 101089.

[19] Sánchez-Fernández, L. P., & Gómez-García, J. (2019). Comparison of deep learning models for fish recognition in underwater images. *Journal of Marine Science and Engineering*, 7(9), 321.

[20] Majumdar, A. K., & Oinam, R. (2019). Fish species recognition using deep learning algorithms: A comparative study. *Proceedings of the 2019 International Conference on Signal Processing and Communication*, 194–198.

[21] Hu, P., Wang, B., & Zhao, Y. (2020). A fish recognition system based on deep learning. *Proceedings of the 2020 International Conference on Computer Science and Application Engineering*, 123–127.

[22] Chea, C., Tachibana, K., & Oyama, Y. (2020). Development of a real-time fish detection system for underwater image data using CNN. *Proceedings of the 2020 International Conference on Control, Automation and Diagnosis*, 150–155.

[23] Perera, I., Fernando, T., & Yapa, H. (2020). Fish classification using deep CNNs.

International Journal of Computer Science and Network Security, 20(2), 193–198.

[24] Ali, F. B., & El-Said, S. A. (2020). Fish species recognition based on deep learning algorithms. *Proceedings of the 3rd International Conference on Communications, Signal Processing, and their Applications*, 1–6.

[25] Adiwinata, Y., et al. (2020). Fish species recognition with Faster R-CNN Inception-V2 using QUT Fish dataset. *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, 11(2), 118–127.

[26] Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges, and open research. *IEEE Access*, 8, 108952–108971.

[27] Yang, X., Zhang, S., Liu, J., Gao, Q., Dong, S., & Zhou, C. (2020). Deep learning for smart fish farming: Applications, opportunities, and challenges. *Aquaculture Reports*, 18, 100432.

[28] Chen, C., Xu, S., & Wang, S. (2018). Recognition of fish species based on deep learning algorithms. *Proceedings of the 2018 IEEE 5th International Conference on Cloud Computing and Intelligence Systems*, 197–200.